UBC OVARIAN CANCER SUBTYPE CLASSIFICATION AND OUTLIER DETECTION

DATA255 Deep Learning
Prof. Simon Shim
Group 5: Hui Yun, Manyu Zhang, Tingshu Lu, Jason Djuandy

TABLE OF CONTENTS

1. INTRODUCTION

PROBLEM AND MOTIVATION

According to the American Cancer Society:

- Ovarian cancer ranks **5th** in cancer deaths among women
- Risk of getting ovarian cancer during her lifetime is about **1 in 78**
- Lifetime chance of dying from ovarian cancer is about **1 in 108**

Problems with Ovarian Cancer:

- Heterogeneity
  - Multiple Subtypes
  - Often overlapping features between subtypes
- Different Subtype, Different Treatment
- Traditional Histopathological Diagnosis - Diagnosis based on Tissue Sample
  - Interobserver Variability between pathologists (Subjective)
  - Time Consuming (Manual)
  - Shortage in Pathology Experts with expertise in gynecologic malignancies

SOLUTION

Utilize the Growing Field of Artificial Intelligence and Machine Learning

- Convolutional Neural Networks are capable of performing Image Classification
- Use said Image Classification and apply it to Histopathology Images
  - Train on subtype labeled data
  - Perform multiclass-classification on different ovarian cancer subtypes
  - Potentially even able to detect rare outlier subtypes
- **Potential Applications:**
  - Automate Diagnosis - Not as time consuming
    - Fast Diagnosis → Treat Immediately → Better Patient Outcome
  - Solve the issue of interobserver variability - 'One' Observer
  - Allows Ovarian Cancer Diagnosis to be accessible everywhere
  - Unearth Rare Outlier Subtypes

SOLUTION

Market Research:

- Most startups are funded only recently using AI to detect Cancer. (Example: CureMetrix). Relatively 'New' Field.

University of British Columbia Kaggle Competition:

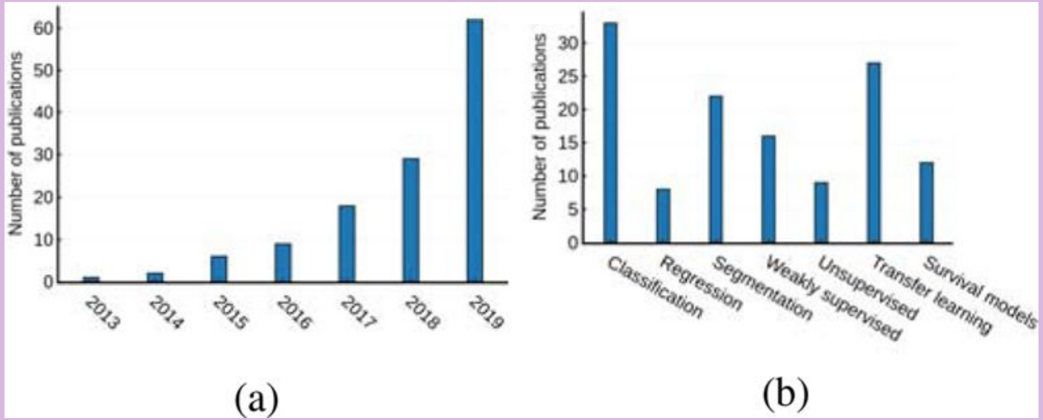
- **UBC Ovarian Cancer Subtype Classification and Outlier Detection (UBC-OCEAN)**
    - Dataset Provided of Tissue Images
    - Perform Multiclass Classification based on different Subtypes
    - Contains Multiple Subtypes
    - Perform Outlier Detection*

2. LITERATURE REVIEW

Author	Problem	Models Used	Results
Mostavi et al. (2020)	Cancer Classification	CNN (1D-CNN, 2D-Vanilla-CNN, 2D-Hybrid-CNN)	**Varies** 93.9-95% accuracy for 34 classes; 88% for 5 subtypes
Parvin et al. (2020)	Breast Cancer Classification	CNN (LeNet-5, AlexNet, VGG-16, ResNet-50, Inception-v1)	**Inception-v1** High accuracy and AUC across magnification factors
Chaturvedi et al. (2020)	Skin Cancer Classification	Pre-trained CNNs (Xception, InceptionV3, etc.) and Ensembles	**ResNeXt101** ResNeXt101 (93.20%), Best Ensemble (92.83%)
Wang et al. (2018)	Prostate Cancer Detection	Fully Convolutional Networks (FCN)	**FCN (Cascaded)** 100% detection rate at low false-positive rate
Nguyen et al. (2019)	Breast Cancer Classification	CNN	**CNN** 73.68% validation accuracy for multi-class classification

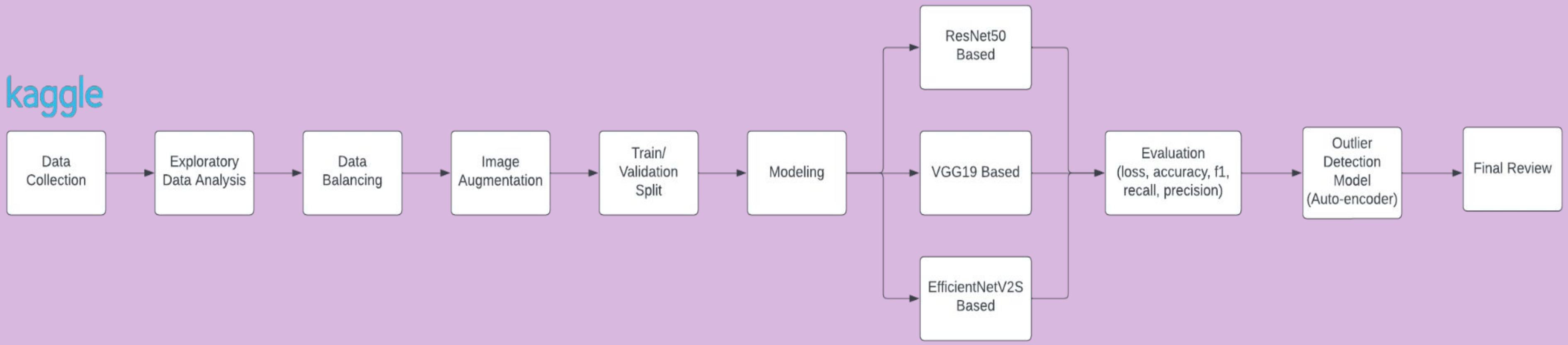
RELATED WORK

- Machine Learning in the Medical Field
  - Growing Research Field
  - Large Potential
  - Capable Solution
    - Classification
    - Segmentation, etc
- Srinidhi et al. (2021) Deep neural network models for computational histopathology: A survey



3. METHODOLOGY

METHODOLOGY WORKFLOW DIAGRAM



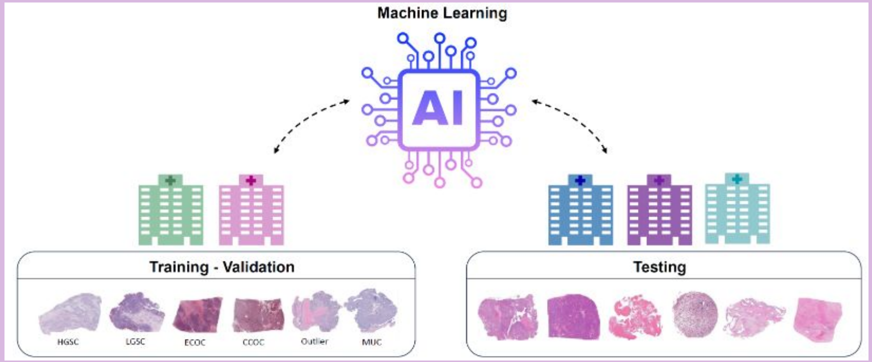
4. DATA PROCESSING

4.1 DATA EXPLANATION

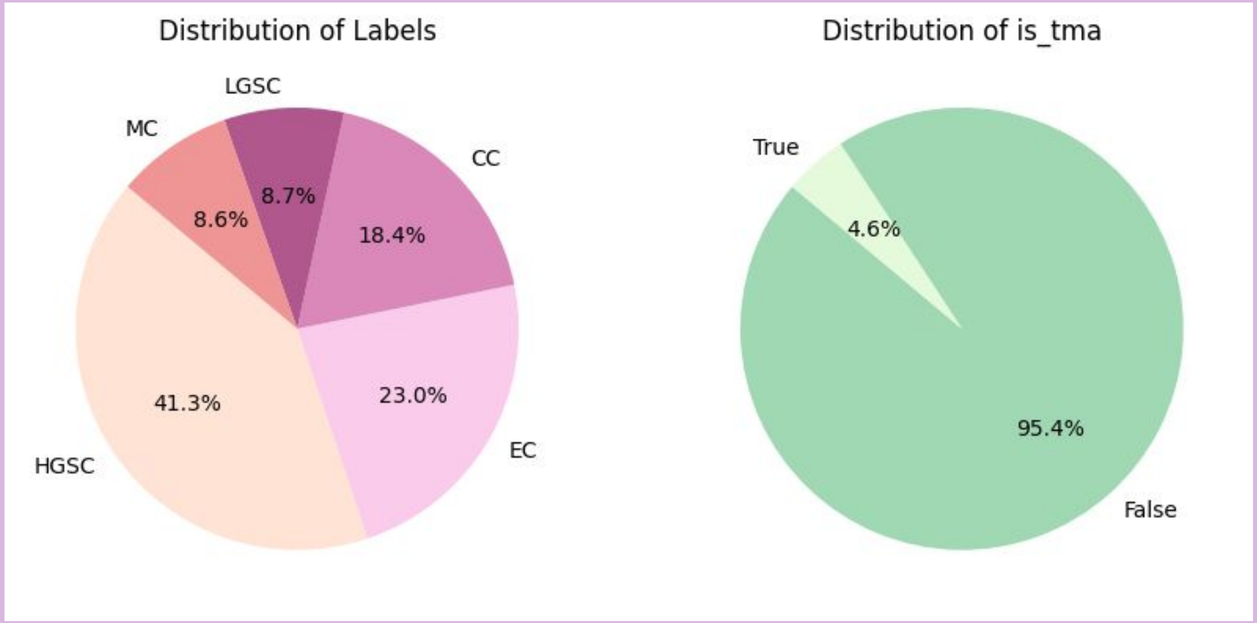
- **Dataset Provided by Kaggle Competition:**
  UBC Ovarian Cancer Subtype Classification and Outlier Detection (UBC-OCEAN)
- **Contains Subtype Images:**
  - High-grade serous carcinoma (HGSC)
  - Clear-cell ovarian carcinoma (CC)
  - Endometrioid (EC)
  - Low-grade serous (LGSC)
  - Mucinous carcinoma (MC).
  - Other* - Outlier Detection
- **Tissue Sample images**
  - <u>Whole Slide Image (WSI)</u>
    - 20x Magnification
    - Very Large File Size
  - <u>Tissue Microarray (TMA)</u>
    - 40x Magnification
    - Roughly 4000 x 4000 pixels

4.1 DATA EXPLANATION

- **Provided Dataset Folder ~ 790 GB**
  - Test_images - 1 image
    - 1 WSI - Original Size
  - Train_images - 538 image
    - 513 WSI - Original Size
    - 25 TMA - Original Size
  - **Test_thumbnails - 1 image**
    - 1 WSI - Smaller Size
  - **Train_thumbnails - 513 images**
    - 513 WSI - Smaller Size
- **Utilized Folders**
  - Test_thumbnails
  - Train_thumbnails
  - Train_images
    - 25 TMA Only



4.2 DATA VISUALIZATION (EDA)



4.2 DATA VISUALIZATION (EDA)

HGSC Summary Table

	count	mean	std	min	25%	50%	75%	max
image_width	222.000000	48637.690000	19250.380000	2964.000000	33713.000000	46182.000000	65055.750000	105763.000000
image_height	222.000000	28939.700000	9605.100000	2964.000000	21426.750000	28148.000000	36520.000000	49215.000000

LGSC Summary Table

	count	mean	std	min	25%	50%	75%	max
image_width	47.000000	43519.260000	20967.170000	2964.000000	28327.000000	47191.000000	61055.000000	79527.000000
image_height	47.000000	24774.680000	1954.180000	2964.000000	18419.000000	22106.000000	32227.000000	50155.000000

EC Summary Table

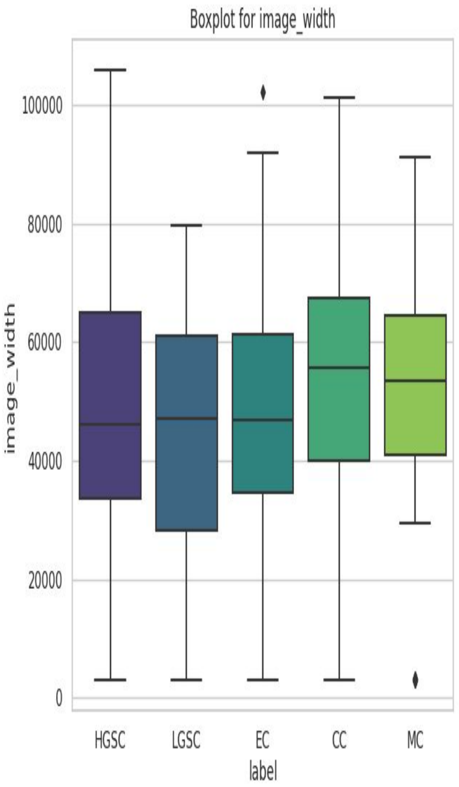
	count	mean	std	min	25%	50%	75%	max
image_width	124.000000	47486.190000	19315.680000	2964.000000	34610.000000	46843.500000	61233.000000	102100.000000
image_height	124.000000	29935.230000	10485.050000	2964.000000	22937.000000	30521.500000	38077.000000	48293.000000

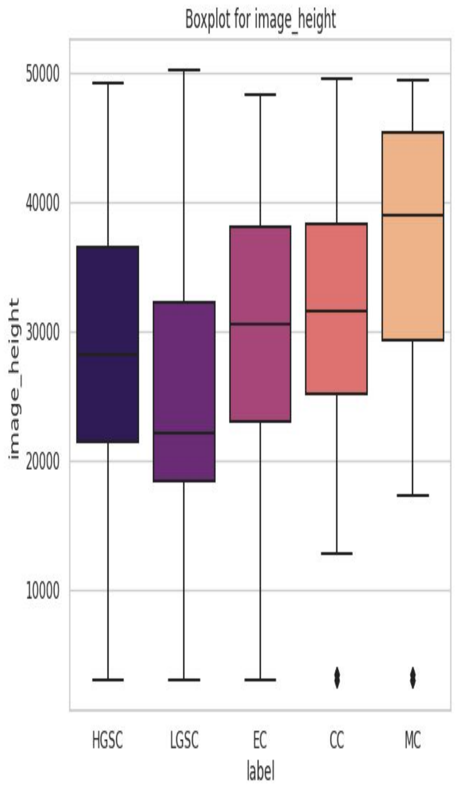
CC Summary Table

	count	mean	std	min	25%	50%	75%	max
image_width	99.000000	52992.680000	21069.160000	2964.000000	39887.500000	55551.000000	67358.500000	101254.000000
image_height	99.000000	31205.060000	0439.520000	2964.000000	25160.000000	31569.000000	38313.500000	49543.000000

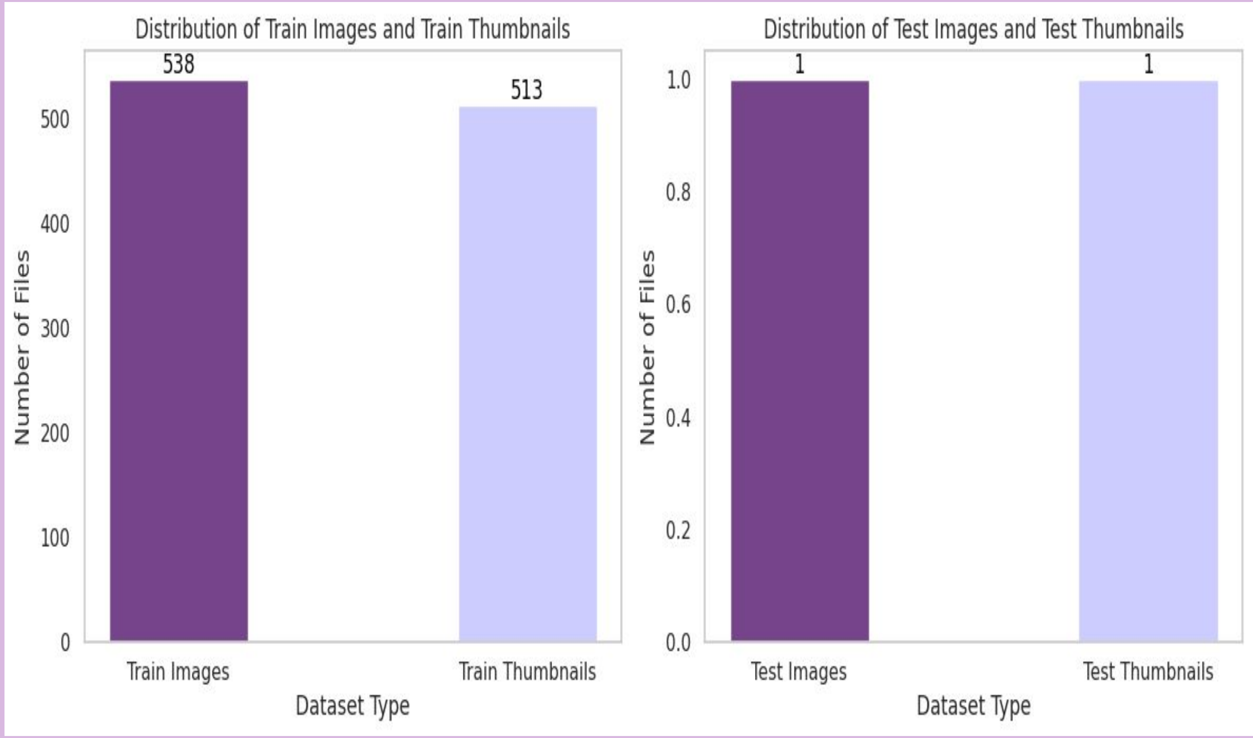
MC Summary Table

	count	mean	std	min	25%	50%	75%	max
image_width	46.000000	50193.350000	21504.080000	2964.000000	40890.000000	53381.500000	64451.000000	91031.000000
image_height	46.000000	34872.980000	3589.990000	2964.000000	29308.500000	38963.500000	45346.750000	49395.000000

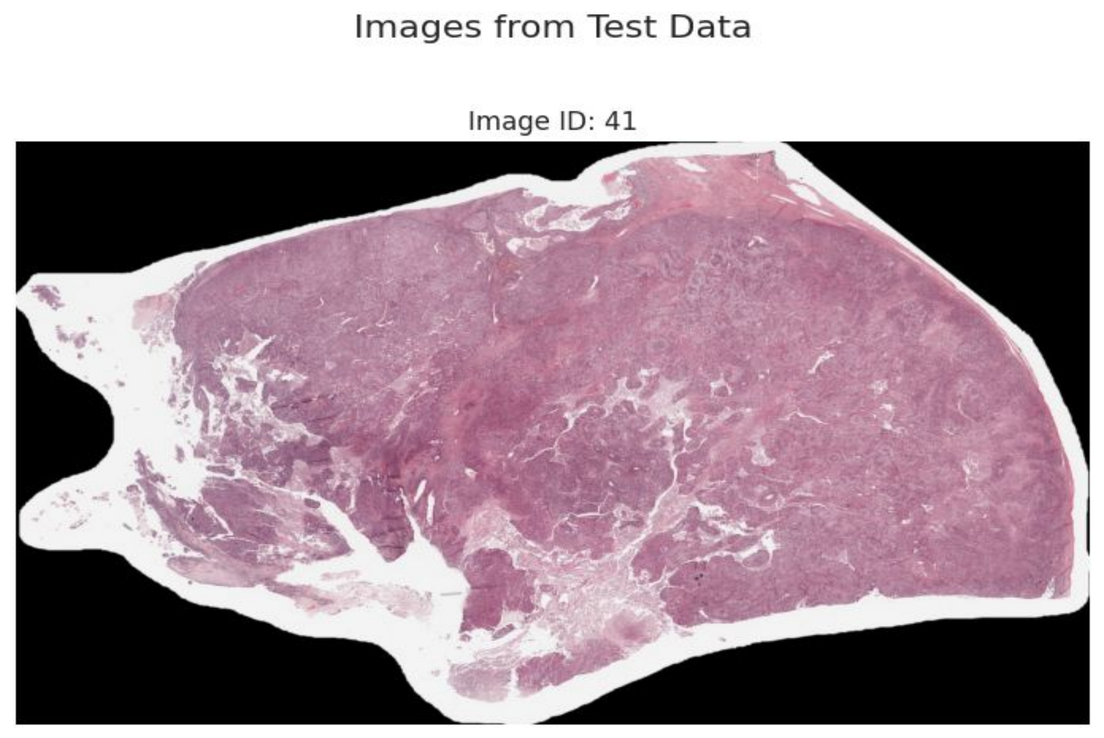




4.2 DATA VISUALIZATION (EDA)



4.2 DATA VISUALIZATION (EDA)



4.3 DATA CLEANING AND VALIDATION

1. Data Cleaning

- Check duplicate image to remove - None

2. Data Validation

1. Thumbnail folder does not contain all the train images but the missing images are all TMA images that are in small size. We used thumbnail image + missing images that are TMA from train folder.
2. Adding full_path
3. Data Imbalance

```
label
HGSC    222
EC      124
CC       99
LGSC     47
MC       46
Name: count, dtype: int64
```

Solution:

The imbalance class is presented that would skew the prediction to the class that has the most weight. We increased number to image to the same level to balance out the class

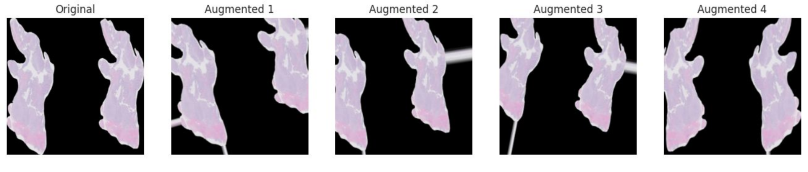
4.4 DATA TRANSFORMATION

Image (x)

```
[[  0   0   0]
 [  0   0   0]
 [  0   0   0]
 ...
 [229 228 233]
 [229 227 232]
 [229 227 232]]
```

Resize image and convert images(x) to numpy array

Data Augmentation



rotate, flip, zoom and shift

Label (y)

```
[2 2 2 ... 4 4 4]
```

Convert label(y) to numpy array

```
[[0. 0. 1. 0. 0.]
 [0. 0. 1. 0. 0.]
 [0. 0. 1. 0. 0.]
 ...
 [0. 0. 0. 0. 1.]
 [0. 0. 0. 0. 1.]
 [0. 0. 0. 0. 1.]]
```

Perform one-hot encoding on Target Value(y value)

5. PROBLEMS & MODELING

5.1 PROBLEM FORMATION & PROPOSED SOLUTION

1. ***Subtype Classification:***
   Subtype classification is a multi-class (5-classes) classification problem.
   Difficulties:
   1) Number of images is less
   2) Class distribution is imbalanced
   3) Image size is too large, high resolution

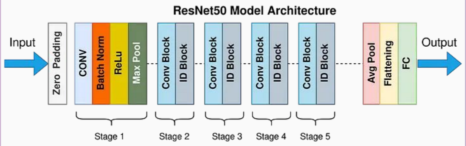
   Solution:
   Train multi-class classification neural network based on balanced and processed images.

2. ***Outlier Detection:***
   Outlier detection is predicting if a given image is one of the subtypes or not.
   Difficulties:
   1) No "outlier" image sample given
   2) No clear model selection guidance

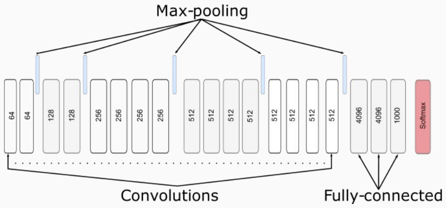
   Solution:
   Train autoencoder to learn non-outliers' pattern and predict outliers, no need of "outlier" image.

5.2 MULTICLASS CLASSIFICATION

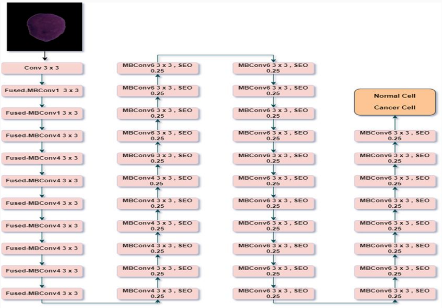
ResNet50



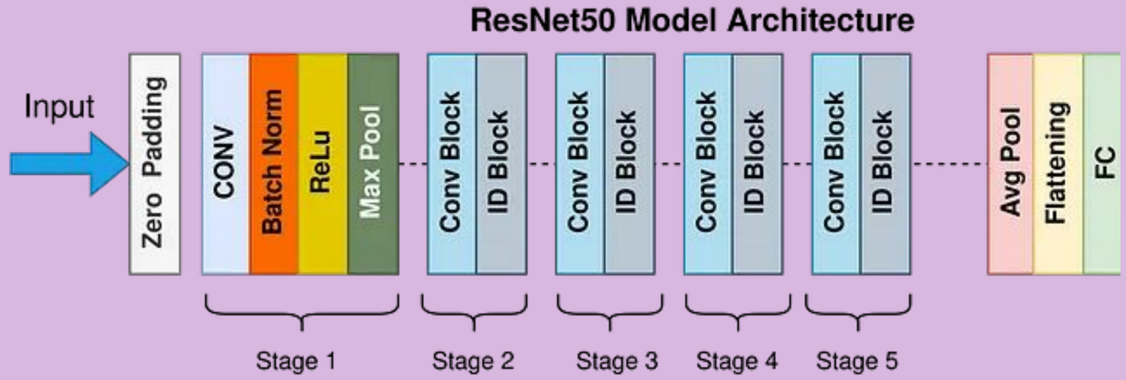
VGG19



EfficientNetV2S



5.2.1 ResNet50 – Model Justification



Input Layer

```
Model: "resnet50"
__________________________________________________________________________________________________
 Layer (type)                   Output Shape         Param #     Connected to
==================================================================================================
 input_1 (InputLayer)           [(None, 256, 256, 3  0           []
                                )]
```

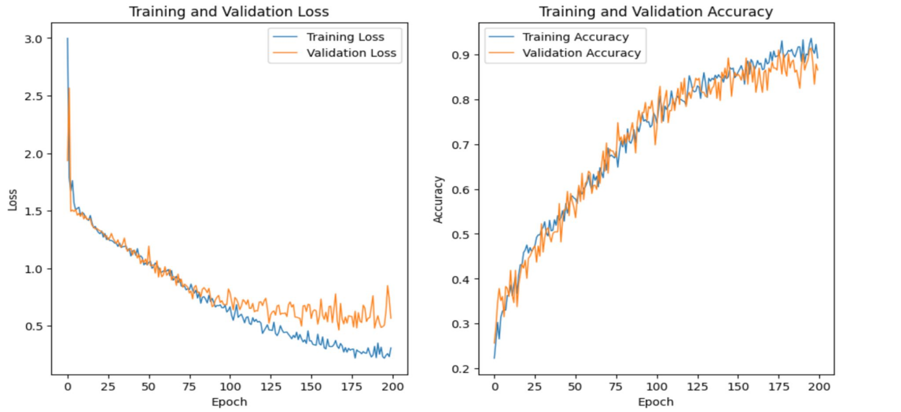
Size -> (512, 512, 3)

Output Layer

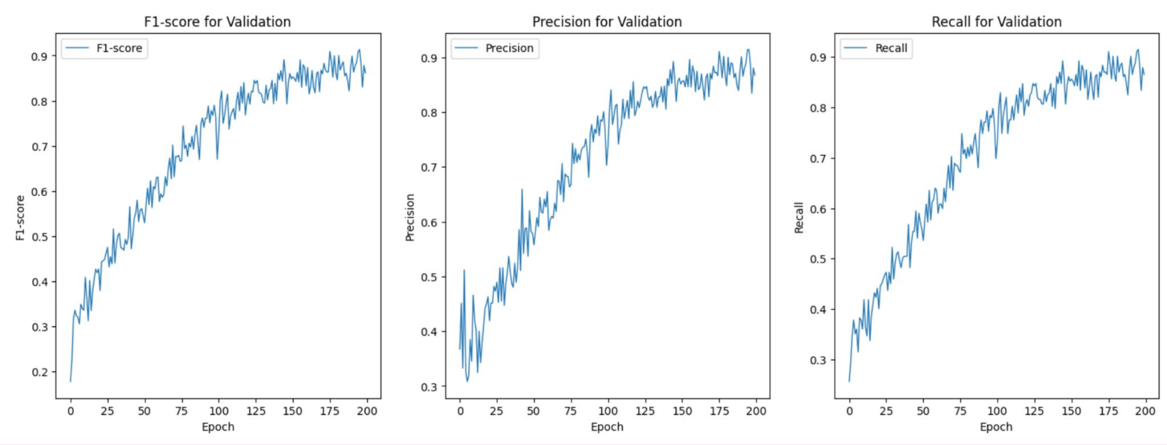
```
dense (Dense)                 (None, 5)            10245       ['flatten[0][0]']

dropout (Dropout)             (None, 5)            0           ['dense[0][0]']
```

Number of Class -> 5

EVALUATION, RESULT AND VISUALIZATION



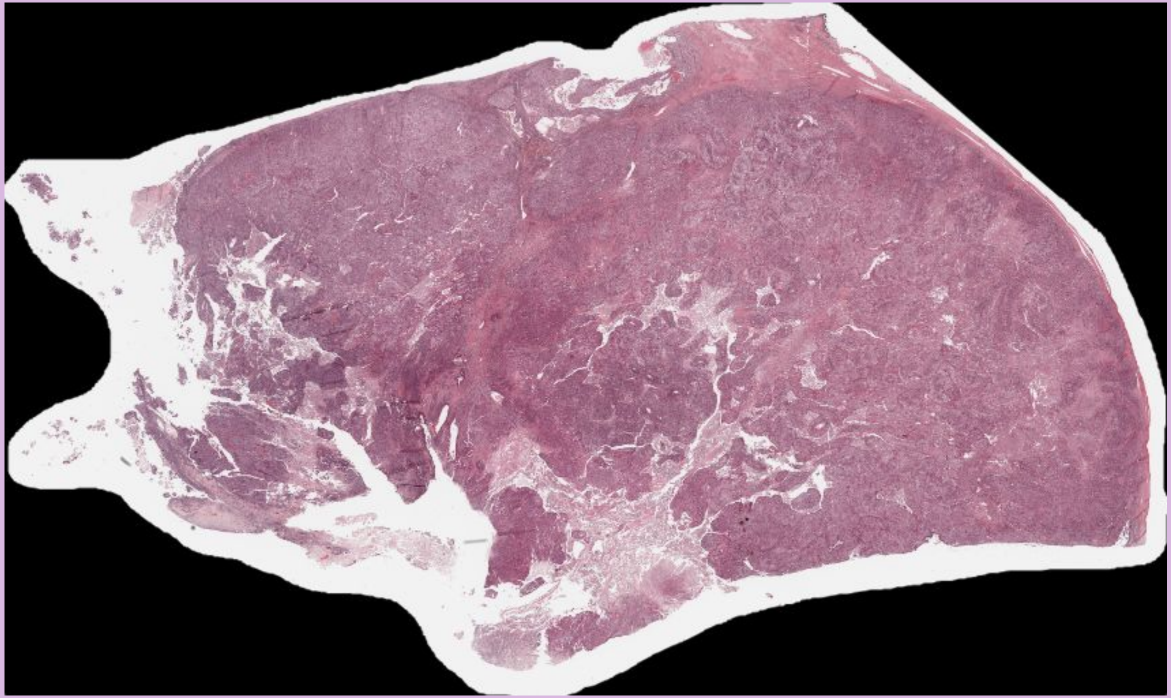
Final Training accuracy: 89%
Final Validation accuracy: 87%



Validation Classification Report:

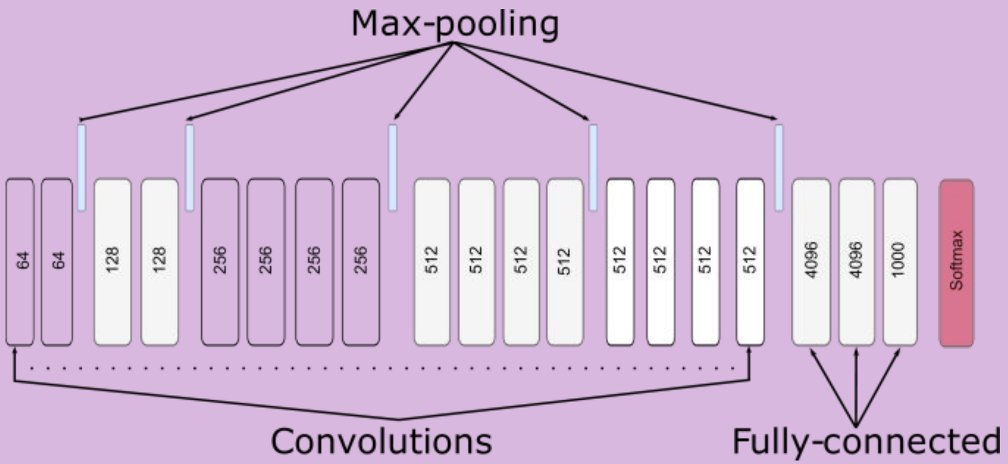
	precision	recall	f1-score	support
0	0.90	0.80	0.84	44
1	0.76	0.87	0.81	45
2	0.88	0.68	0.77	44
3	0.90	1.00	0.95	44
4	0.90	0.98	0.94	45
accuracy			0.86	222
macro avg	0.87	0.86	0.86	222
weighted avg	0.87	0.86	0.86	222

5.2.1 RESNET50 TEST PREDICTION



image_id	label
41	EC

5.2.2 VGG19 – Model Justification

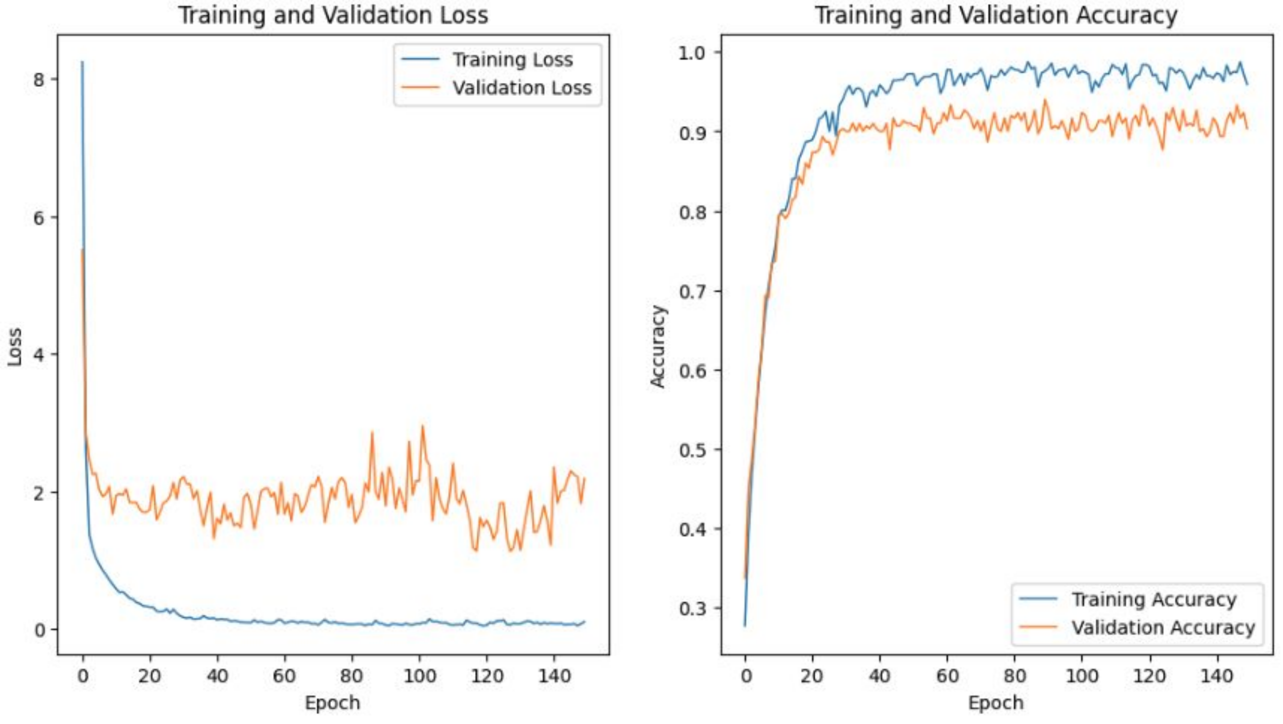



```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 vgg19 (Functional)          (None, 18, 18, 512)       20024384

 global_average_pooling2d (  (None, 512)               0
 GlobalAveragePooling2D)

 dense (Dense)               (None, 64)                32832

 dropout (Dropout)           (None, 64)                0

 dense_1 (Dense)             (None, 5)                 325

=================================================================
Total params: 20057541 (76.51 MB)
Trainable params: 33157 (129.52 KB)
Non-trainable params: 20024384 (76.39 MB)
_________________________________________________________________
```

Image Size: (600, 600, 3)
Images each class: 300
Images in total: 1500
Training : Validation = 1200 : 300

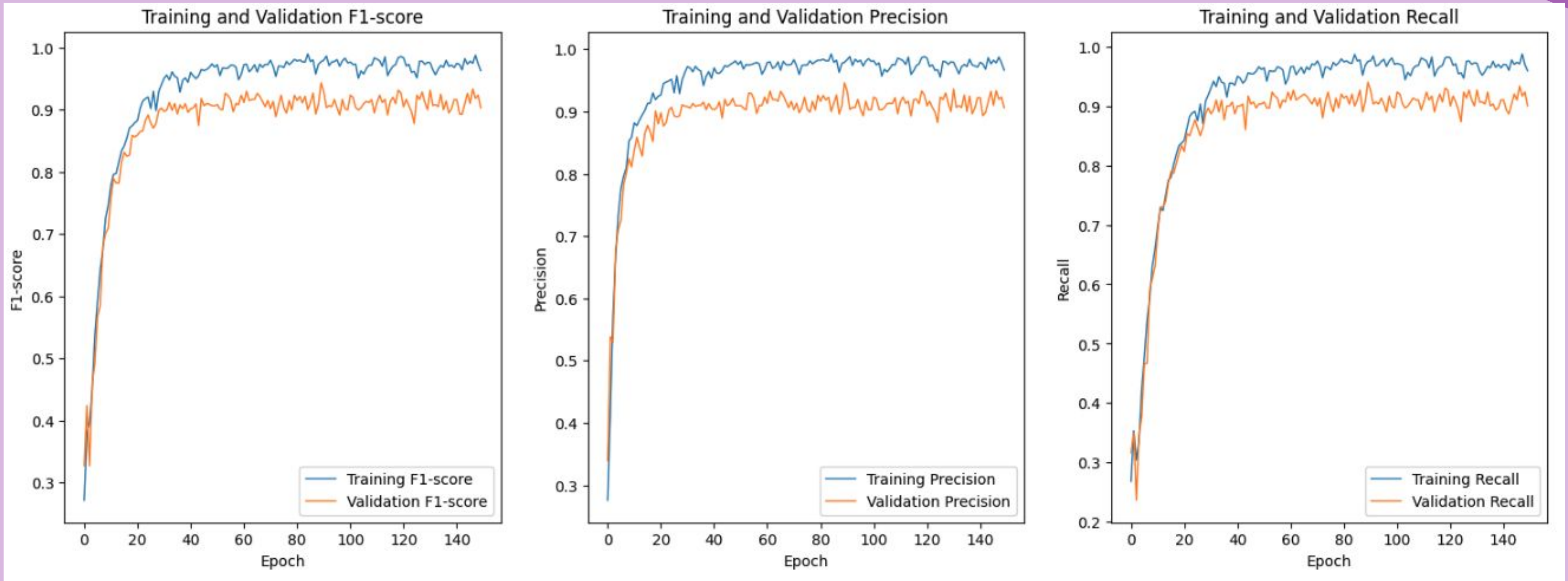
5.2.2 VGG19 RESULTS

150 Epochs



Final Training Accuracy: 99%
Final Validation Accuracy: 90%

5.2.2 VGG19 RESULTS



Final Training F1 Score: 99%
Final Validation F1 Score: 90%

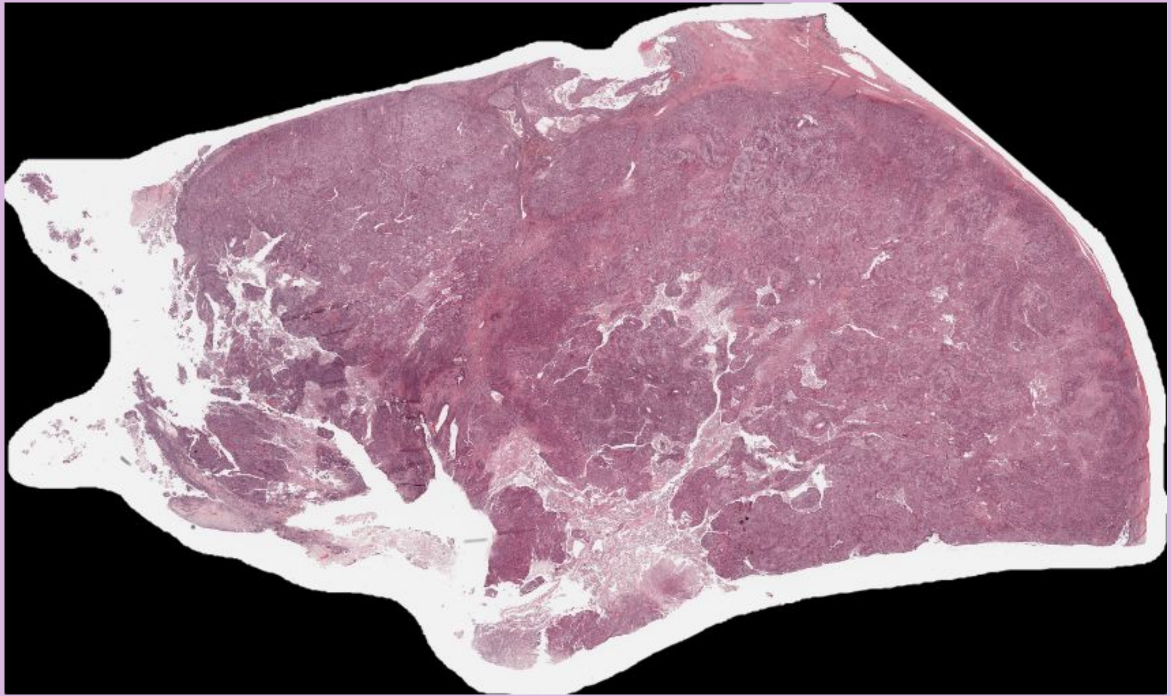
5.2.2 VGG19 RESULTS

Validation Classification Report:

	precision	recall	f1-score	support
0	0.93	0.82	0.87	66
1	0.76	0.92	0.83	49
2	0.84	0.88	0.86	56
3	0.98	0.97	0.98	65
4	0.98	0.94	0.96	64
accuracy			0.90	300
macro avg	0.90	0.90	0.90	300
weighted avg	0.91	0.90	0.90	300

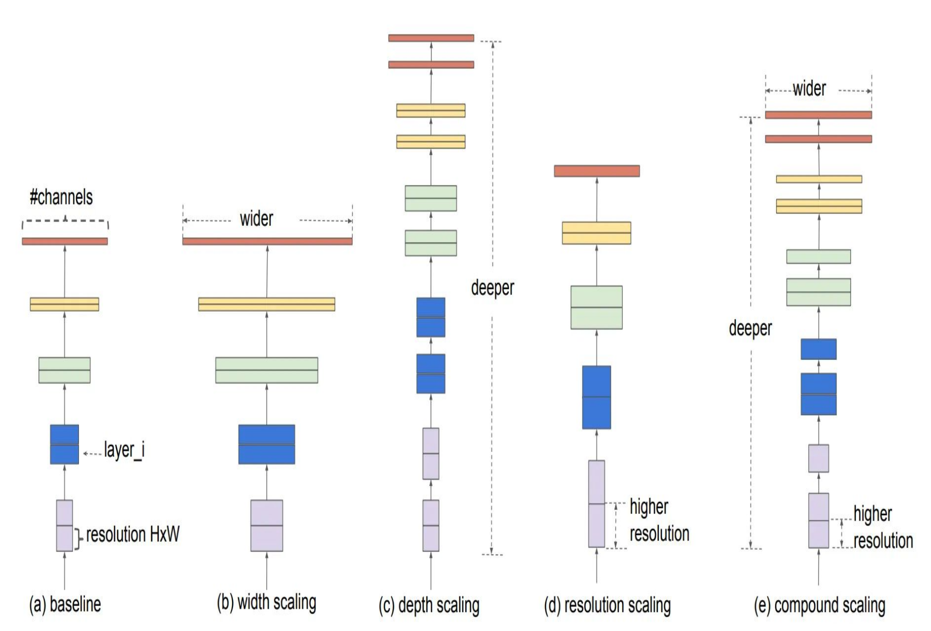
F1 Score of all classes all over 0.83
Weighted F1 score reached 0.9

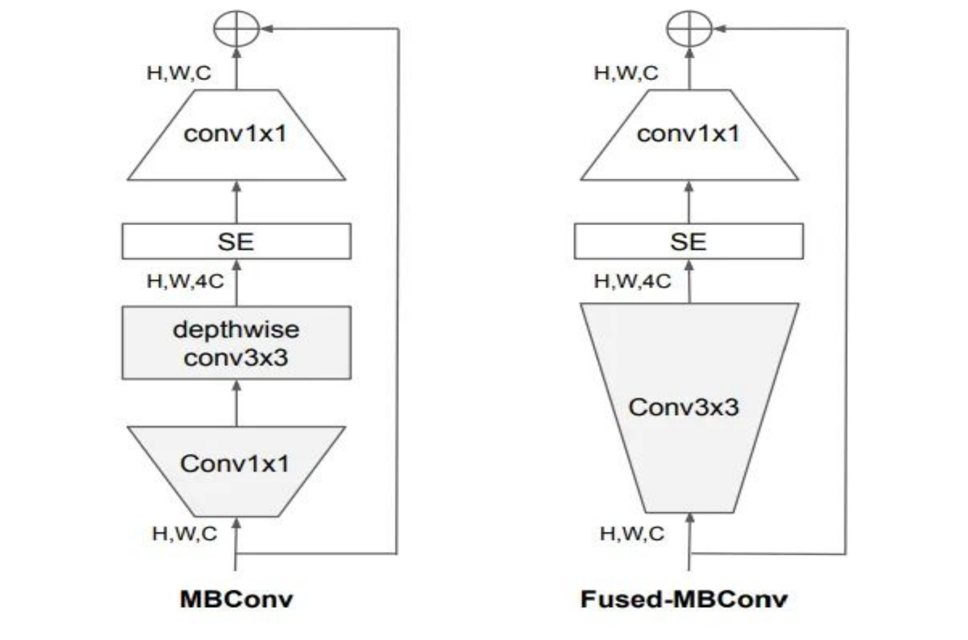
5.2.2 VGG19 RESULTS



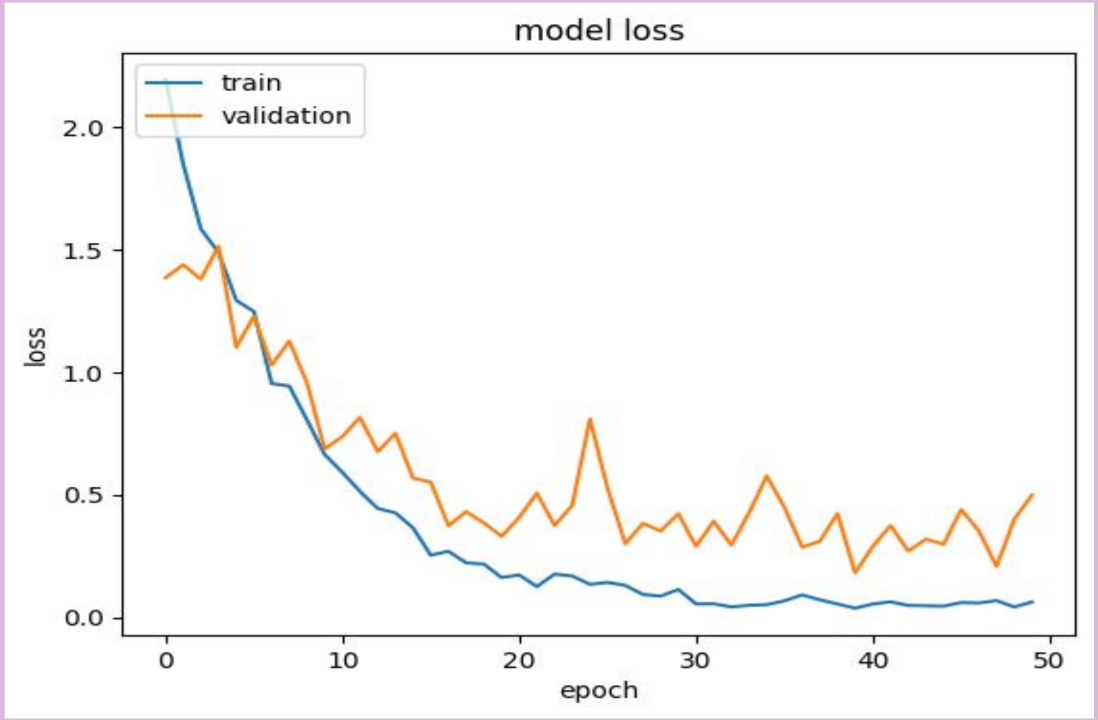
	image_id	label
0	41	LGSC

5.2.3 EfficientNetV2-Model Justification





5.2.3.2 EFFICIENTNETV2 RESULTS

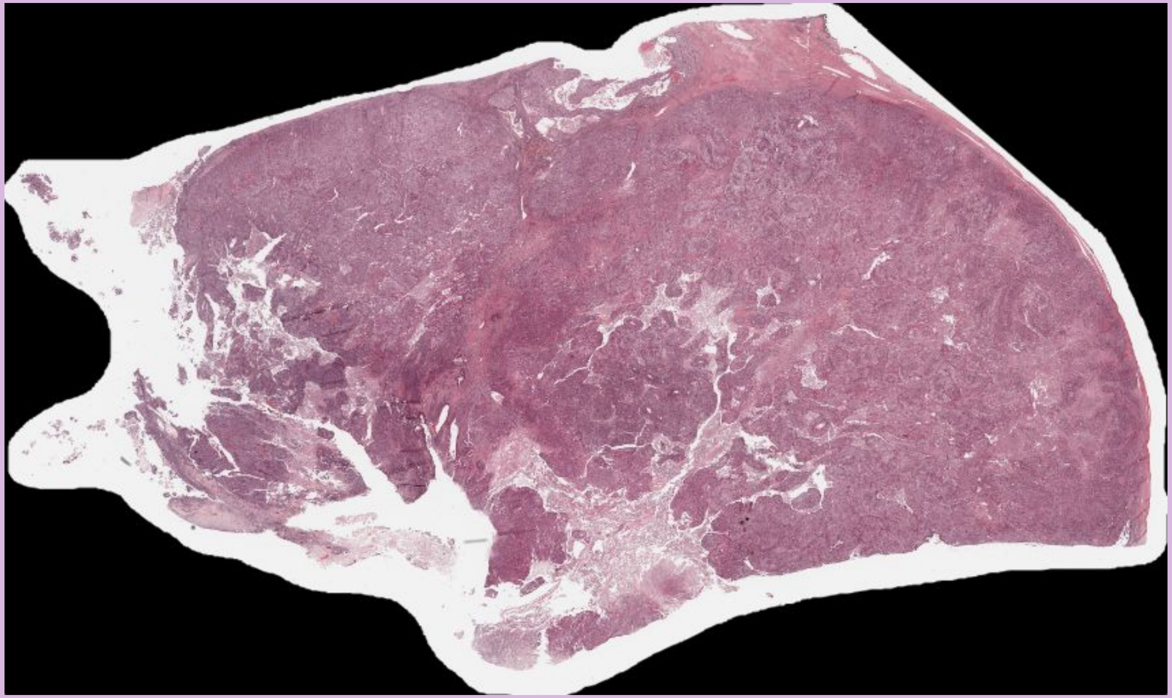


5.2.3.2 EfficientNetV2 Results

Validation Classification Report:

	precision	recall	f1-score	support
0	0.94	0.94	0.94	72
1	0.93	0.88	0.90	16
2	0.97	0.91	0.94	34
3	0.97	1.00	0.98	29
4	0.85	1.00	0.92	11
accuracy			0.94	162
macro avg	0.93	0.95	0.94	162
weighted avg	0.95	0.94	0.94	162

5.2.3.2 EfficientNetV2 Results



	image_id	label
0	41	CC

MODEL RESULT COMPARISON

All models achieved above 85% validation accuracy and balanced metrics across different classes.

ResNet50

Validation Classification Report:

	precision	recall	f1-score	support
0	0.90	0.80	0.84	44
1	0.76	0.87	0.81	45
2	0.88	0.68	0.77	44
3	0.90	1.00	0.95	44
4	0.90	0.98	0.94	45
accuracy			0.86	222
macro avg	0.87	0.86	0.86	222
weighted avg	0.87	0.86	0.86	222

VGG19

Validation Classification Report:

	precision	recall	f1-score	support
0	0.93	0.82	0.87	66
1	0.76	0.92	0.83	49
2	0.84	0.88	0.86	56
3	0.98	0.97	0.98	65
4	0.98	0.94	0.96	64
accuracy			0.90	300
macro avg	0.90	0.90	0.90	300
weighted avg	0.91	0.90	0.90	300

EfficientNetV2S

Validation Classification Report:

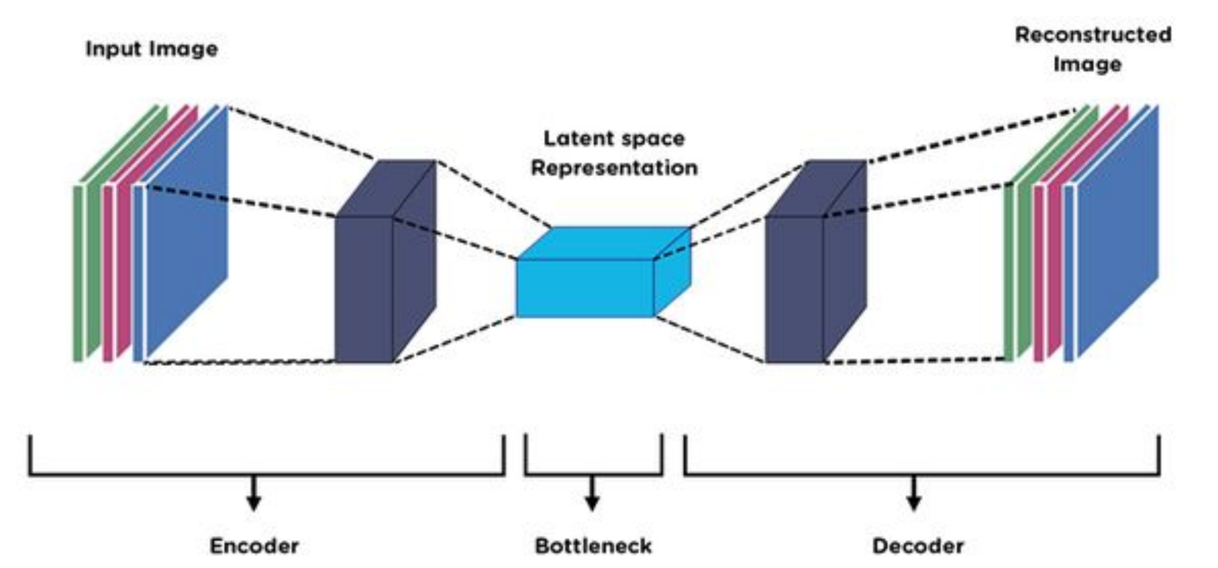
	precision	recall	f1-score	support
0	0.94	0.94	0.94	72
1	0.93	0.88	0.90	16
2	0.97	0.91	0.94	34
3	0.97	1.00	0.98	29
4	0.85	1.00	0.92	11
accuracy			0.94	162
macro avg	0.93	0.95	0.94	162
weighted avg	0.95	0.94	0.94	162

5.3 OUTLIER DETECTION - AUTOENCODER

Training Phase: The autoencoder learns to **encode normal patterns** in the input data, capturing the essential features of the majority of the dataset.

Testing (or Inference) Phase: If the image is **normal** (similar to the training data), the autoencoder should be able to reconstruct it with low error.

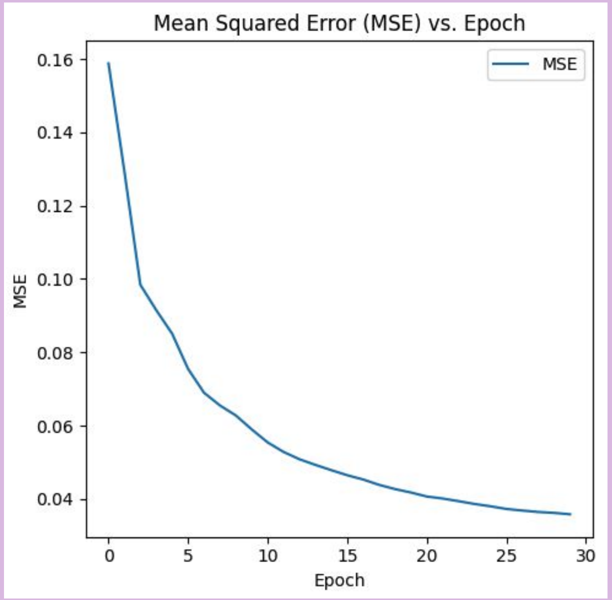
If the image is an **outlier** (different from the training data), the autoencoder may struggle to reconstruct it accurately, resulting in a higher reconstruction error.



5.3 OUTLIER DETECTION - RESULT

- Process and feed given image (all non-outliers) to auto-encoder
- Train for 30 epochs
- Set threshold based on highest MSE (Mean Squared Error), threshold = 0.0992619545…
- Give any image, detect if it is a non-outlier (one of the 5 subclasses) or an outlier

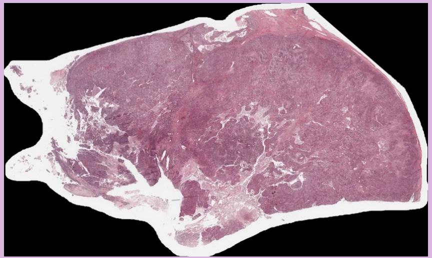
```
encoder = keras.Sequential([
    layers.InputLayer(input_shape=input_shape),
    layers.Conv2D(32, (3, 3), activation='relu', padding='same'),
    layers.MaxPooling2D((2, 2), padding='same'),
    layers.Conv2D(64, (3, 3), activation='relu', padding='same'),
    layers.MaxPooling2D((2, 2), padding='same'),
    layers.Flatten(),
    layers.Dense(32, activation='relu'),
])

decoder = keras.Sequential([
    layers.InputLayer(input_shape=(32,)),
    layers.Dense(np.prod(input_shape), activation='sigmoid'),
    layers.Reshape(input_shape),
])

autoencoder = keras.Sequential([encoder, decoder])
```



5.3 OUTLIER DETECTION - INFERENCE

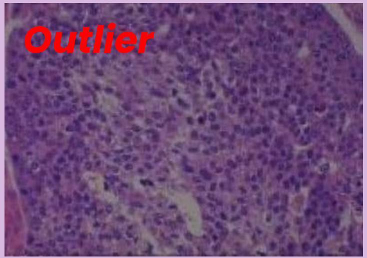
MSE threshold = 0.09926195452676544

1) Official Test Image:



```
Is the single test image an outlier?
False
```

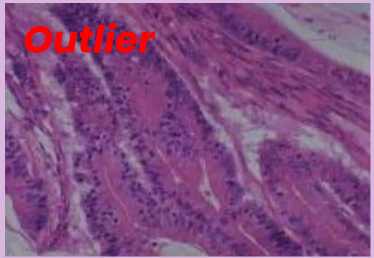
2) Lung Cancer:




```
Is the single test image an outlier? True
```

```
array([0.11325034], dtype=float32)
```

3) Colon Cancer:




```
Is the single test image an outlier? True
```

```
array([0.1096436], dtype=float32)
```

4) Cartoon:


```
Is the single test image an outlier? True
```

```
array([0.12271416], dtype=float32)
```

6. CONCLUSION

FINDINGS & CONCLUSION

1. **All models perform reasonably well with balanced evaluation metrics across all classes.**
2. **Data quality is important in modeling.**

   Based on the experiments of multi-class classification modeling, we compared the results of training images with different resolution, higher resolution training images deliver better accuracy and f1 score. Moreover, it eased overfitting.
3. **Data quantity and balanced class distribution is vital.**

   After image augmentation and balancing, the performance of model improved.
4. **Outlier detection is challenging under the unawareness of outliers, auto-encoder is good for this situation.**

   If trained for more epochs, model can capture more subtle and essential pattern in normal data, which enhance its ability to identify outliers and anomalies.

REFERENCE

Chaturvedi, S. S., Tembhurne, J. V., & Diwan, T. (2020). A multi-class skin cancer classification using deep convolutional Neural Networks. Multimedia Tools and Applications, 79(39–40), 28477–28498. https://doi.org/10.1007/s11042-020-09388-2

Howlander, N. (n.d.). Ovarian cancer statistics: How common is ovarian cancer. Ovarian Cancer Statistics | How Common is Ovarian Cancer | American Cancer Society. https://www.cancer.org/cancer/types/ovarian-cancer/about/key-statistics.html

Srinidhi, C. L., Ciga, O., & Martel, A. L. (2021). Deep neural network models for computational histopathology: A survey. *Medical Image Analysis*, *67*, 101813. https://doi.org/10.1016/j.media.2020.101813

Mostavi, M., Chiu, Y.-C., Huang, Y., & Chen, Y. (2020, April 3). *Convolutional neural network models for cancer type prediction based on gene expression - BMC medical genomics*. BioMed Central. https://bmcmedgenomics.biomedcentral.com/articles/10.1186/s12920-020-0677-2

Nguyen, P. T., Nguyen, T. T., Nguyen, N. C., & Le, T. T. (2019). Multiclass breast cancer classification using Convolutional Neural Network. 2019 International Symposium on Electrical and Electronics Engineering (ISEE). https://doi.org/10.1109/isee2.2019.8920916

Parvin, F., &; Mehedi Hasan, Md. A. (2020). A comparative study of different types of convolutional neural networks for breast cancer histopathological image classification. 2020 IEEE Region 10 Symposium (TENSYMP). https://doi.org/10.1109/tensymp50017.2020.9230787

Wang, Y., Zheng, B., Gao, D., &; Wang, J. (2018). Fully convolutional neural networks for prostate cancer detection using multi-parametric Magnetic Resonance Images: An initial investigation. 2018 24th International Conference on Pattern Recognition (ICPR). https://doi.org/10.1109/icpr.2018.8545754

THANKS
Q & A

DATA255 Deep Learning
Prof. Simon Shim
Group 5: Hui Yun, Manyu Zhang, Tingshu Lu, Jason Djuandy