Real vs Fake Image Identification

Paul, Kate, Sean

Background

Image generation

- With the recent advancements in AI models and web interfaces, the average person is now able to create hyper-realistic images with only a simple prompt

- This will be extremely useful, but always raises questions of security and integrity of images

Faces

- Human faces are one example which can now be easily generated.
- It has now become fairly easy to falsify or create images of people doing things they have never done, or videos people saying things they have never said
- This obviously may have serious implications
    - The legal system is quickly trying to catch up

Problem statement

Our goal

“In an attempt to catch up to this ever changing space, we will attempt to create a model which will be able to identify real vs fake images of human faces”

Data

Real and Fake Face Detection dataset

- This dataset was found on Kaggle and was create by dataset contains expert-generated high-quality photoshopped face images.
    - Computational Intelligence and Photography Lab
      Department of Computer Science, Yonsei University
- The images are composite of different faces, separated by eyes, nose, mouth, or whole face.

Data Visualization

Real Images:



Fake Images:

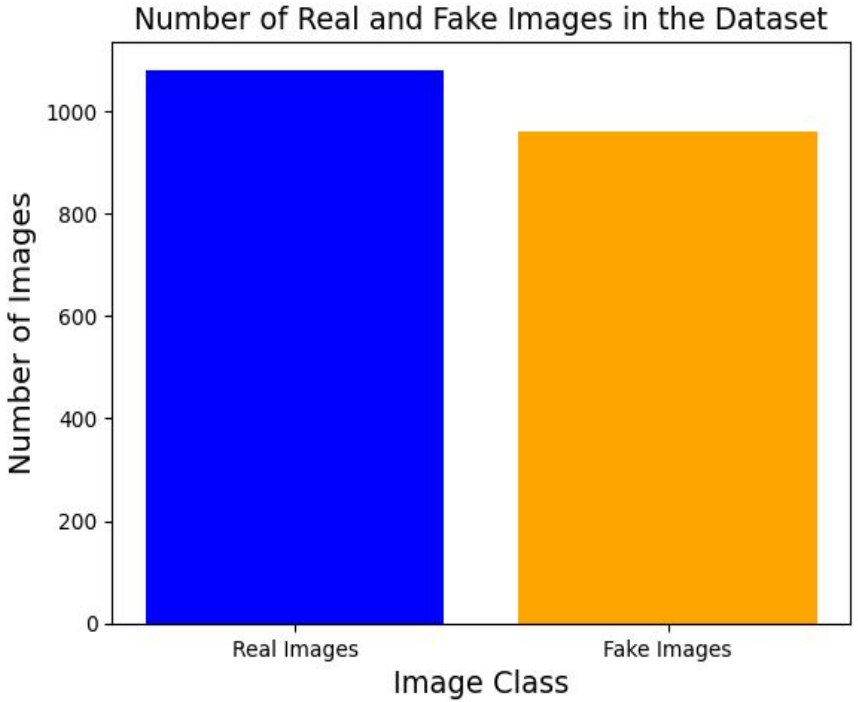


Dataset Distribution

Real images: 1,081 (53%)

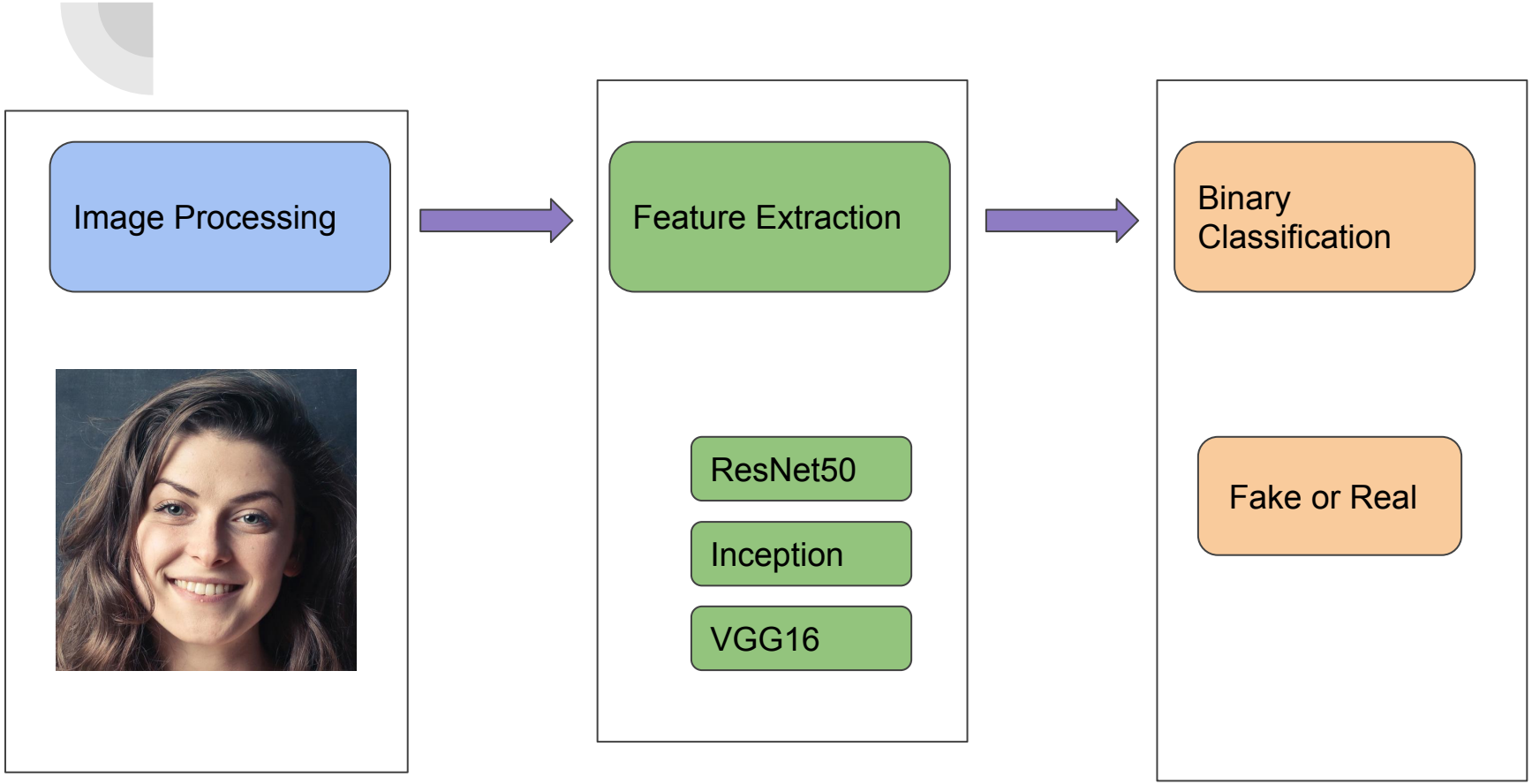
Fake images: 960 (47%)

A relatively balanced dataset



Framework

Workflow Diagram of Fake Detection Algorithm



Data Preprocessing

Image Resizing

For VGG16 & Resnet- 50 , resize all the image data into 224 x 224

For GoogLeNet (Inception v3), resize all the images into 299 x 299

Pre-processing and splitting

Pixel normalization : To scale the pixel values into a range of [0, 1]:

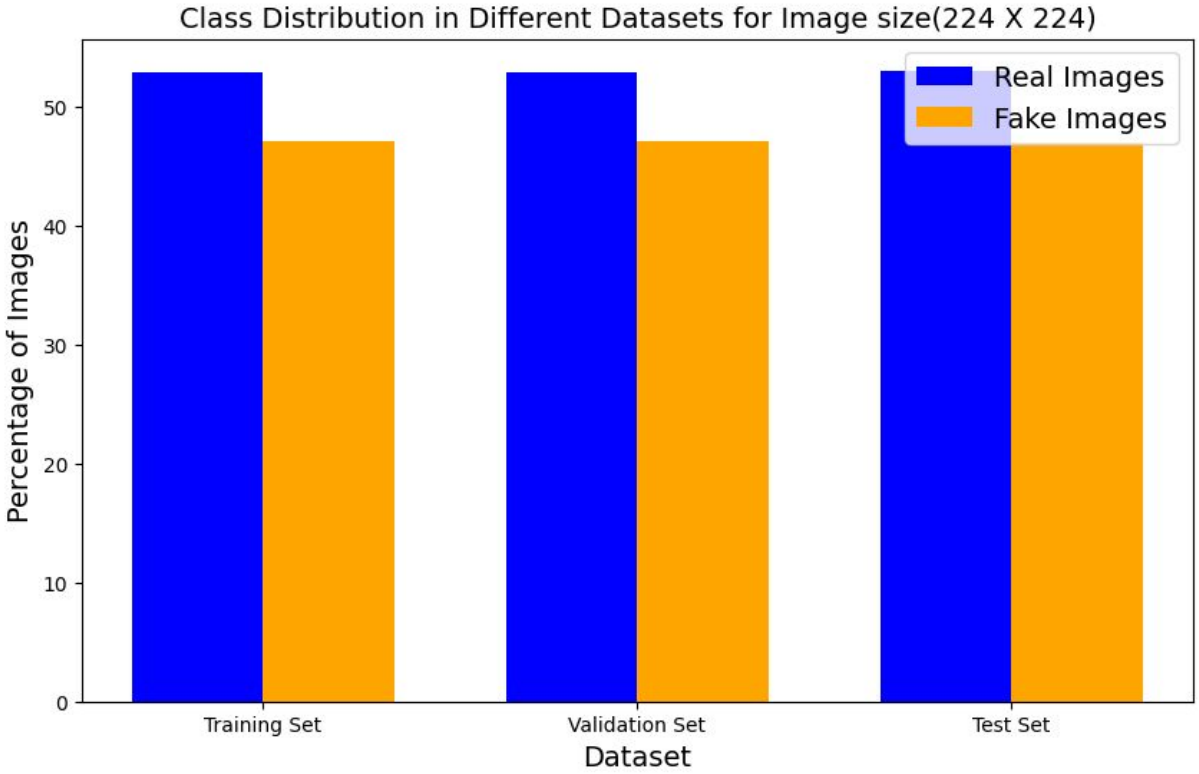
Regardless of the image size = (224, 224) or (299 x 299), we divide the images by 255 in the case of 8-bit images

Labels:
Positive Class (1) → Fake images
Negative Class (0) → Real images

Data splitting: Even though our dataset is considered as relatively balanced, we still want to apply stratification on the labels when splitting the images and labels into the training/validation and test sets, this can ensure that the proportions of different classes in the original image dataset are preserved in each of these 3 subsets.

→ To avoid potential biases in the data distribution across different subsets.

Class Distribution after Data Splitting



Model Selection

ResNet50

ResNet50 is a deep convolutional neural network with 50 layers, featuring skip connections and residual blocks.

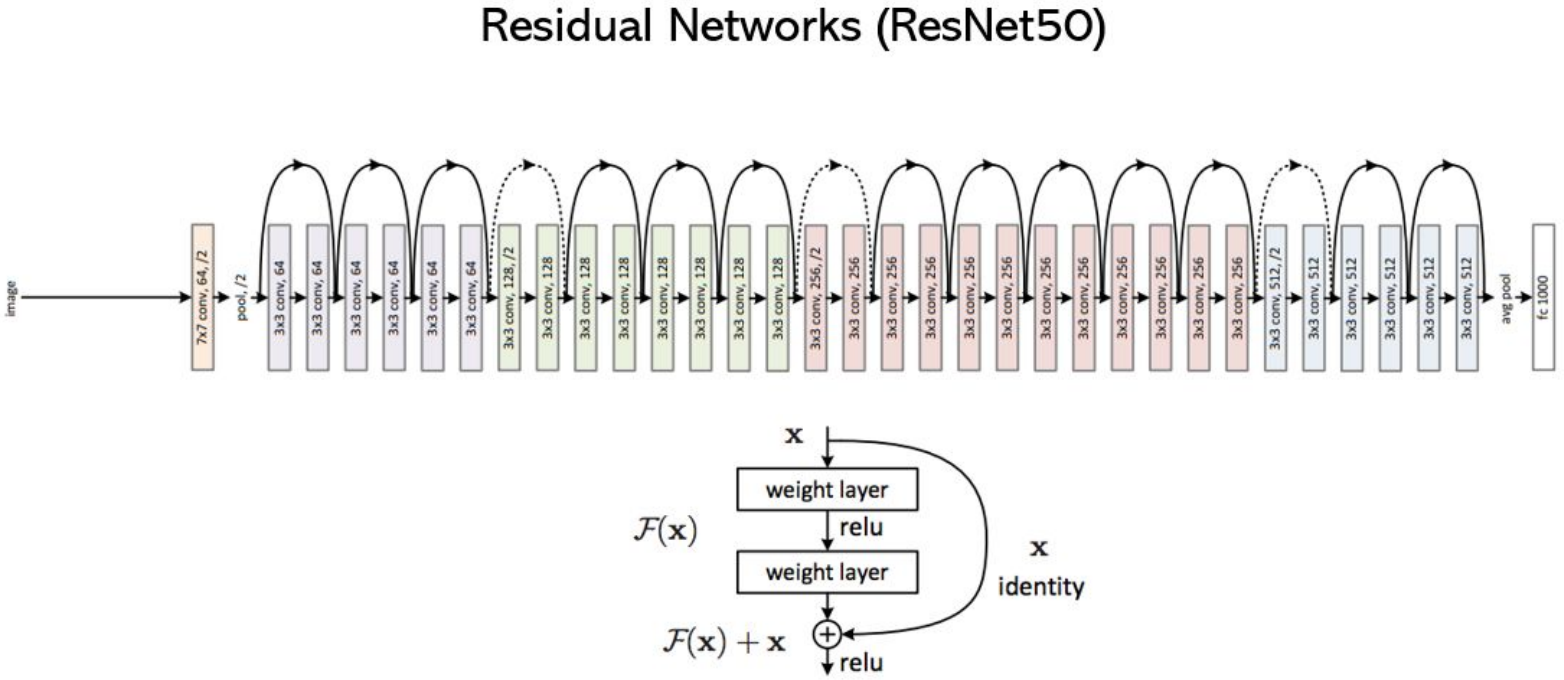
Advantages:

- Its depth (50 layers) allows it to capture complex features and patterns in the data.
- Uses residual(skip) connections, which can help deal with the vanishing gradient problem.

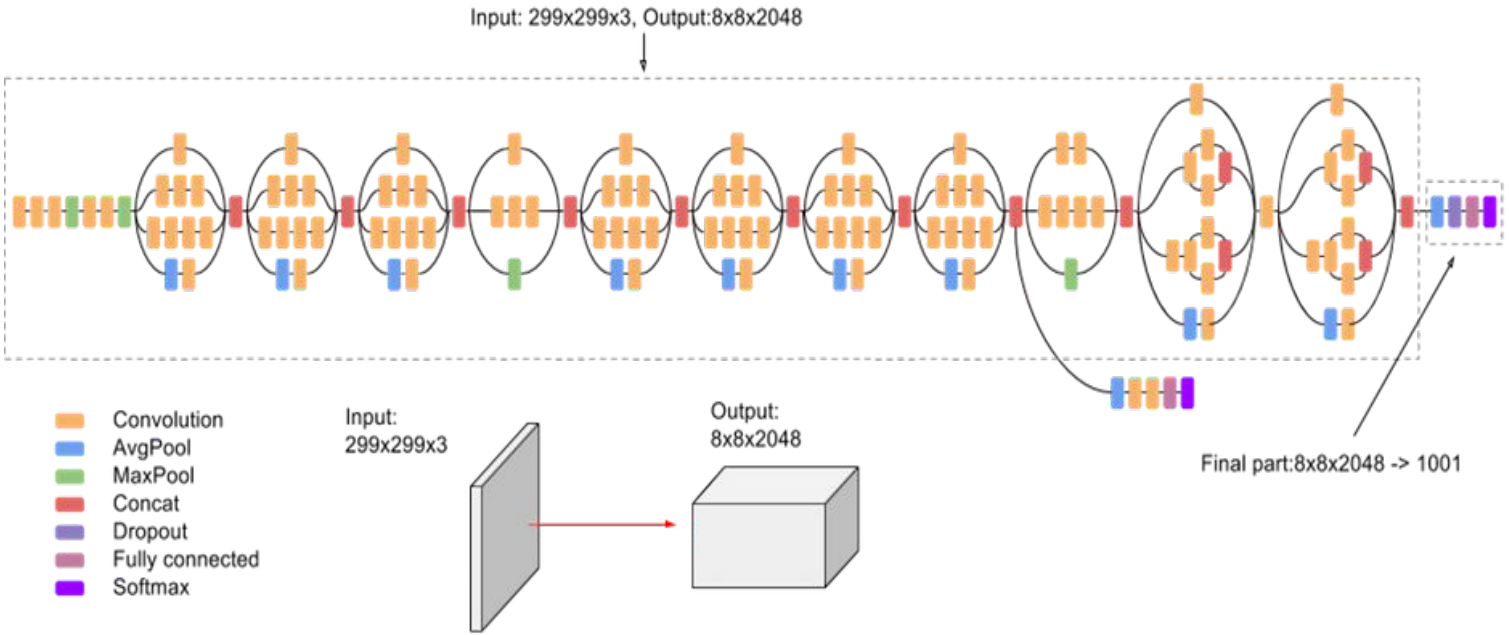
Disadvantages:

- The ResNet50's depth make it computationally expensive and memory consuming.
- It has a tendency towards overfitting

ResNet 50


Residual Networks (ResNet50)
image
7x7 conv, 64, /2
pool, /2
3x3 conv, 64
3x3 conv, 128, /2
3x3 conv, 128
3x3 conv, 256, /2
3x3 conv, 256
3x3 conv, 512, /2
3x3 conv, 512
avg pool
fc 1000
x
weight layer
relu
weight layer
$\mathcal{F}(\mathbf{x})$
$\mathcal{F}(\mathbf{x}) + \mathbf{x}$
relu
x
identity

Inception v3 (GoogLeNet)



Inception v3 (GoogLeNet)

It has already trained on more than millions of images from the “ImageNet” database.

Total Number of layers: 48 (including the fully connected networks)
[individual operational layers: 311 layers]

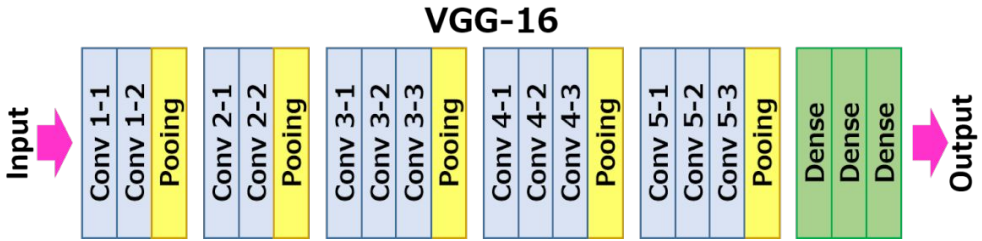
The concept of Inception Module: the term "Mixed" refers to the Inception modules

In Inception v3, there are a total of 11 "Mixed" layers (Inception modules)

Instead of using a single filter size, the module uses filters of different sizes (1x1, 3x3, and 5x5) within the same layer. By using multiple filter sizes, this can capture features at various scales effectively, leading to higher feature extraction efficiency.

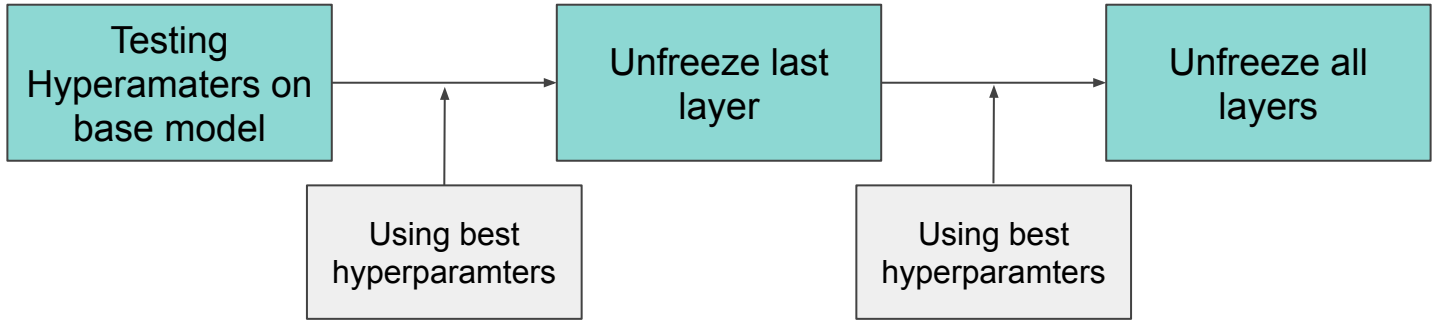
It requires less memory and computational power during training.

VGG16



- VGG16 (Visual Geometry Group 16) is a convolutional neural network (CNN) architecture for computer vision tasks.
- Architecture: 13 Convolutional Layers, 5 Max-Pooling Layers, 3 Fully Connected Layers, and 1 Output Layer.
- Input: Fixed-size RGB image (224x224 pixels).
- Activation: ReLU after each convolutional layer.
- Max-Pooling: 2x2 window with a stride of 2 to reduce spatial dimensions.

VGG16 - Model Framework



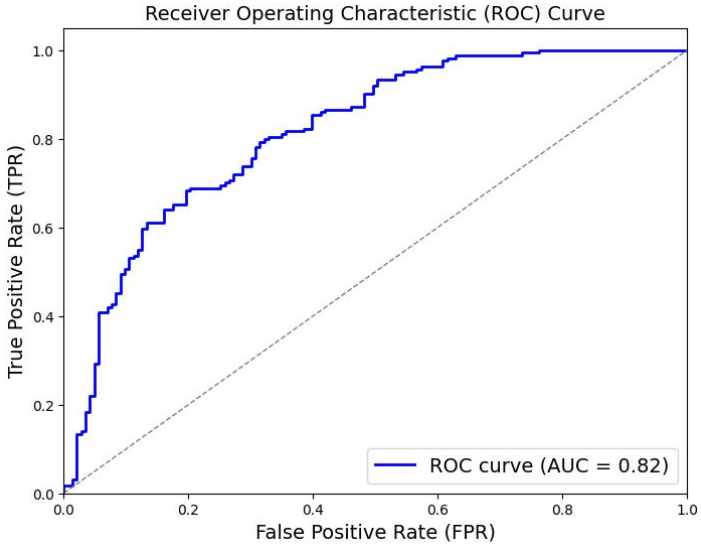
Results

ResNet 50

	All Layers Frozen	Last layer unfrozen	First and last layer unfrozen	All layers unfrozen
Training Accuracy	85	85	85	86
Testing Accuracy	62	64	66	70
AUC (from ROC)	65	71	69	75

Inception v3 (GoogLeNet)

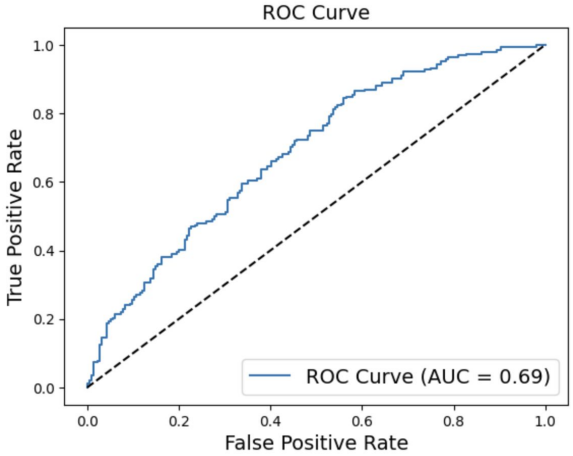
With a relatively balanced dataset and by considering the trade off between “Overfitting” and “Better Probability Prediction” , the best model we choose for Inception is to unfreeze all the layers.



	All layers frozen	Unfreezing till “Mixed 4” layers (using 40% of the total layers)	All layers unfrozen
Training Accuracy	67%	88%	96%
Testing Accuracy	58%	72%	73%
AUC (from ROC)	0.62	0.76	0.82

VGG16

- The model which fared the best in terms of accuracy was the model with all layers frozen



	All layers frozen	Last layer unfrozen	All layers unfrozen
Training Accuracy	79%	87%	78%
Testing Accuracy	63%	62%	60%

Overall Results : Model Analysis

	ReNet-50	Inception v3	VGG16
Training Accuracy	85%	96%	79%
Validation Accuracy	65%	71.2%	72%
Training loss	6.7	5.3	0.53
Validation loss	3.4	6.02	
Testing accuracy (Accuracy Score)	70%	73%	63%
AUC (from ROC)	.75	0.82	0.69

Conclusions

Model Selected

Pick the best model

Essential Considerations:
“Overfitting” - A wide discrepancy between training and testing accuracies
AUC (Area Under the Curve) if we have a balanced dataset

It is essential to strike a balance between overfitting and better probability prediction.

A high AUC indicates that our model is better at distinguishing between the positive and negative classes, and its predicted probabilities are more accurate and reliable.