Model Comparison Documentation

1. Introduction

This documentation compares the performance of three deep learning models:

1. **ResNet18** (implemented from scratch)
2. **DenseNet** (imported from libraries)
3. **Xception** (imported from libraries)

The evaluation focuses on the following metrics:

- ROC-AUC (Receiver Operating Characteristic - Area Under Curve)
- Accuracy
- Loss curves (training and validation)
- Confusion matrix

The comparison helps assess the effectiveness and generalization ability of these models for a given classification task.

2. Models Overview

2.1 ResNet18 (Implemented from Scratch)

ResNet18 is a variant of the ResNet (Residual Network) architecture. It utilizes residual blocks with skip connections to address the vanishing gradient problem in deep networks. For this study, ResNet18 was implemented from scratch, including the custom definition of layers and residual blocks.

2.2 DenseNet (Imported)

DenseNet (Densely Connected Convolutional Network) connects each layer to every other layer in a feed-forward manner. The architecture enhances feature propagation and reduces redundancy.

- **Source**: Predefined architecture from deep learning libraries like TensorFlow/Keras or PyTorch.

2.3 Xception (Imported)

Xception (Extreme Inception) is a deep learning architecture that replaces standard Inception modules with depthwise separable convolutions, improving computational efficiency.

- **Source**: Imported from frameworks such as Keras or PyTorch

3. Evaluation Metrics

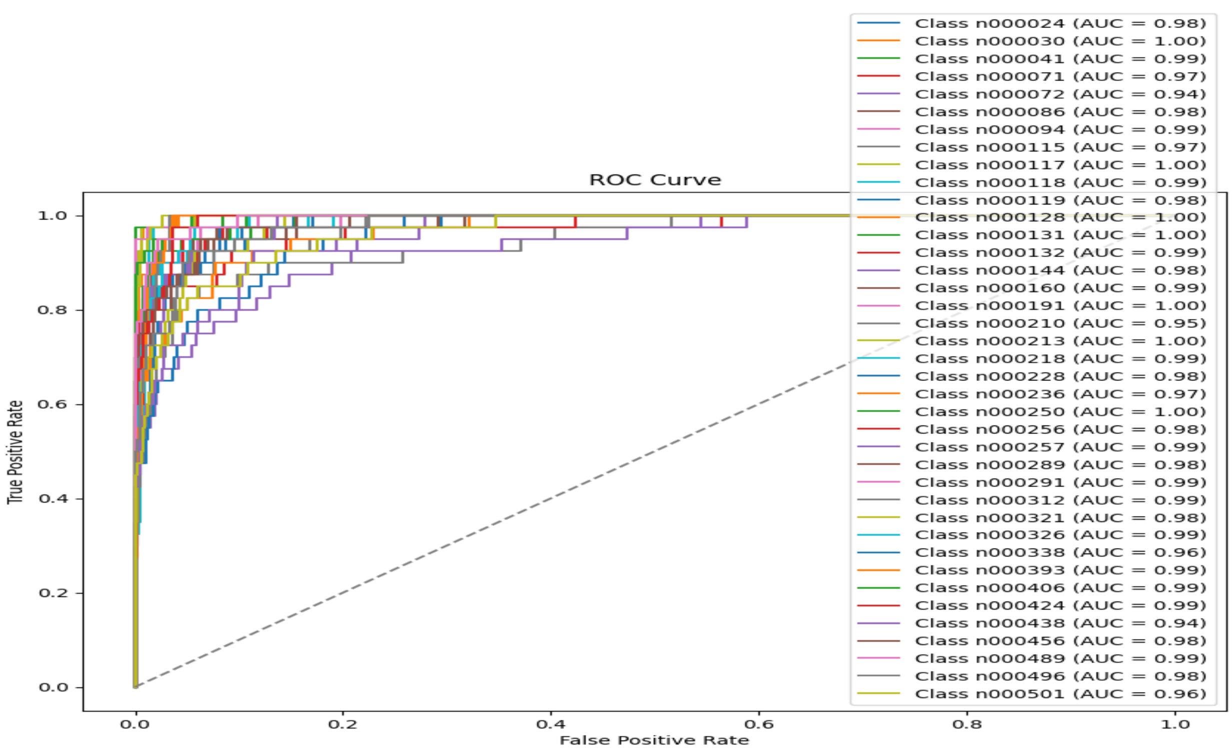
The following evaluation metrics were used to compare the models:

- **ROC-AUC**: Measures the area under the ROC curve, which evaluates model performance across all classification thresholds.
- **Accuracy**: The ratio of correct predictions to total predictions.
- **Loss Curves**: Visual representation of the training and validation loss over epochs.
- **Confusion Matrix**: Summarizes true positives, false positives, true negatives, and false negatives.

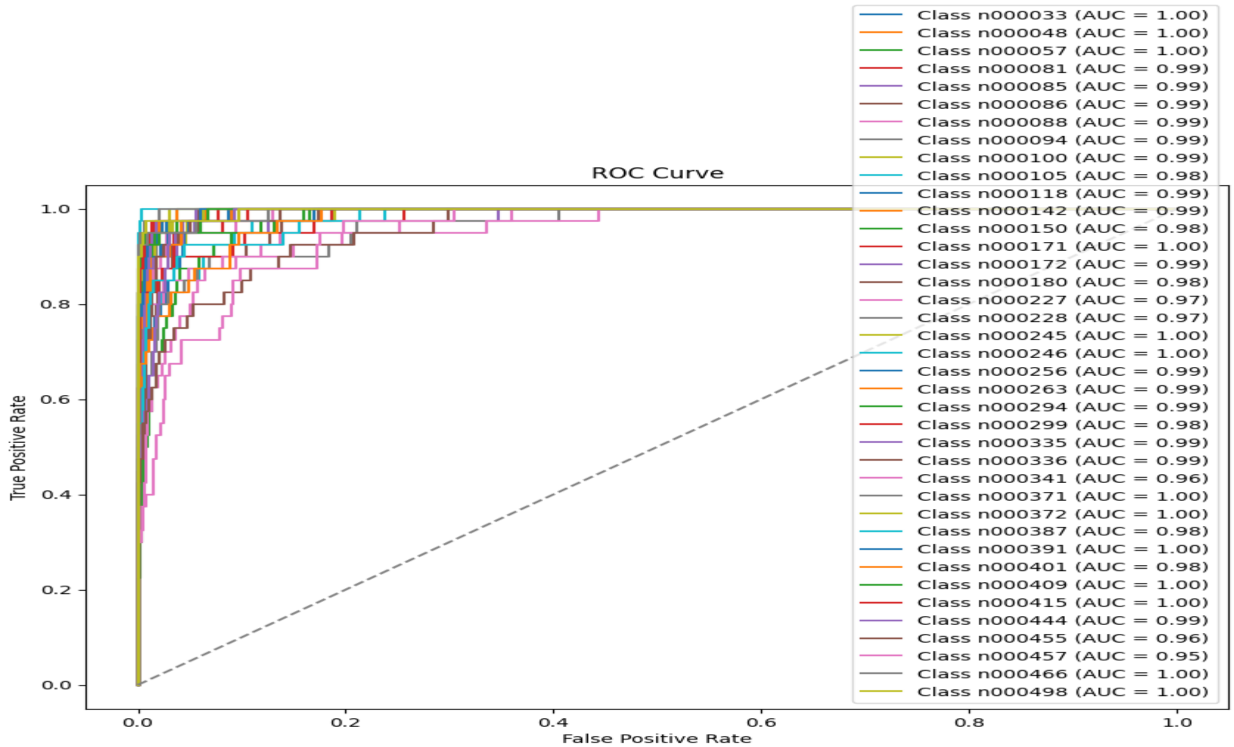
4. Results

4.1 ROC (AUC)

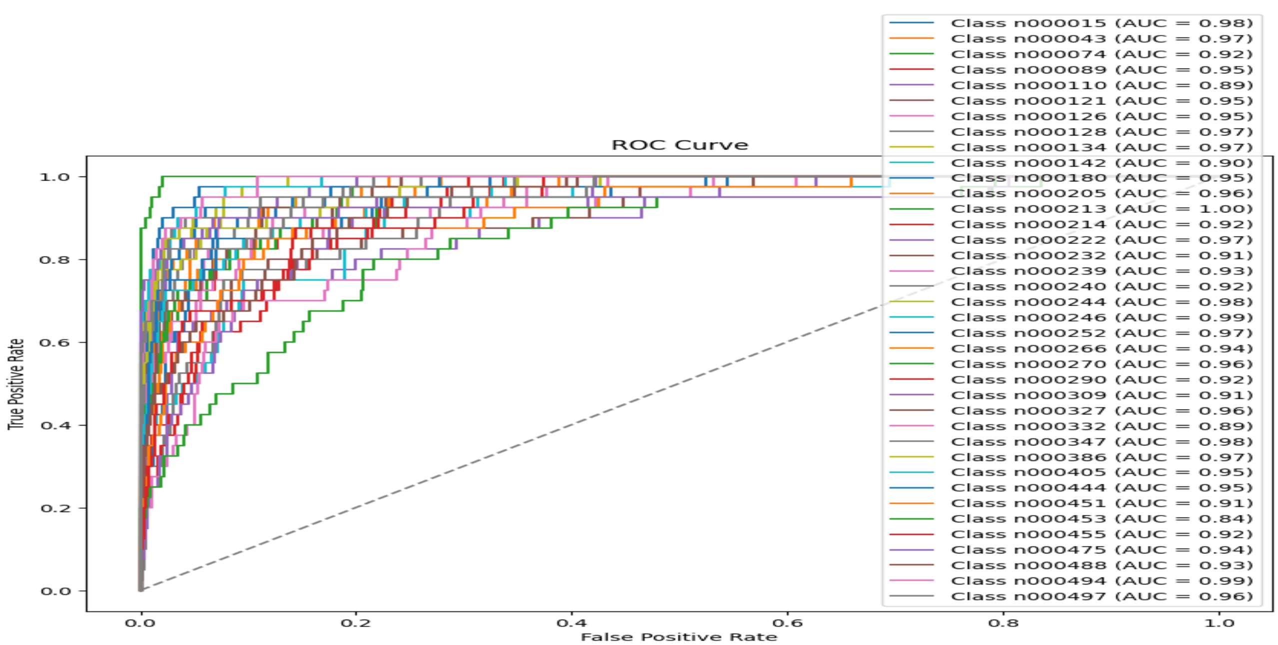
4.1.1 ResNet18



4.1.2 DenseNet

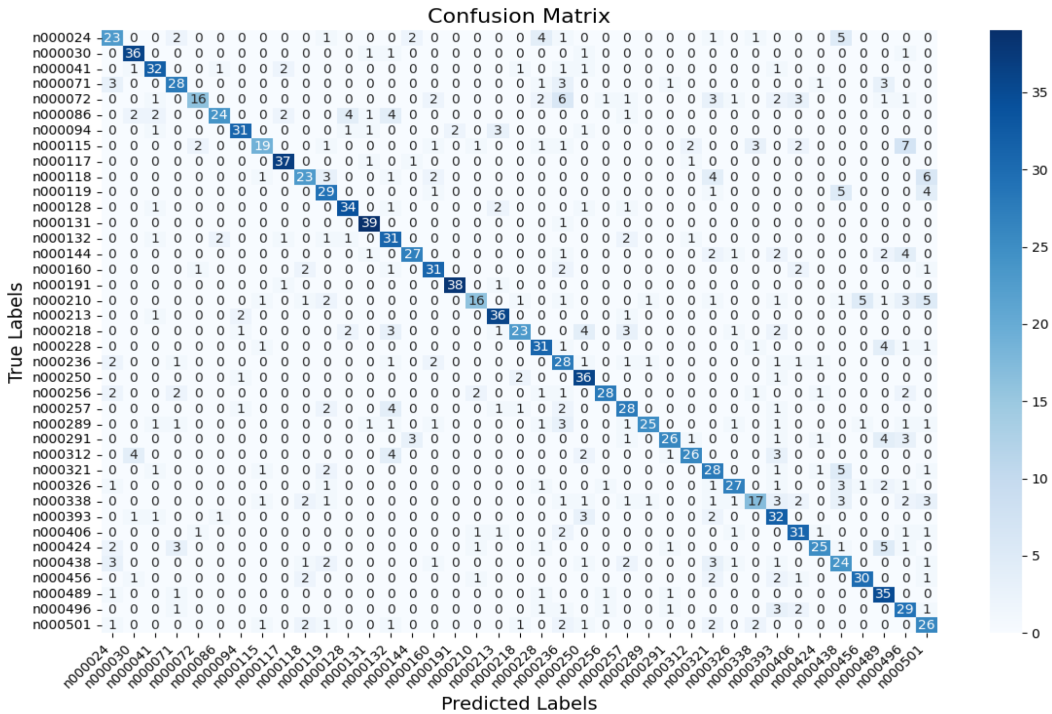


4.1.3 Xception

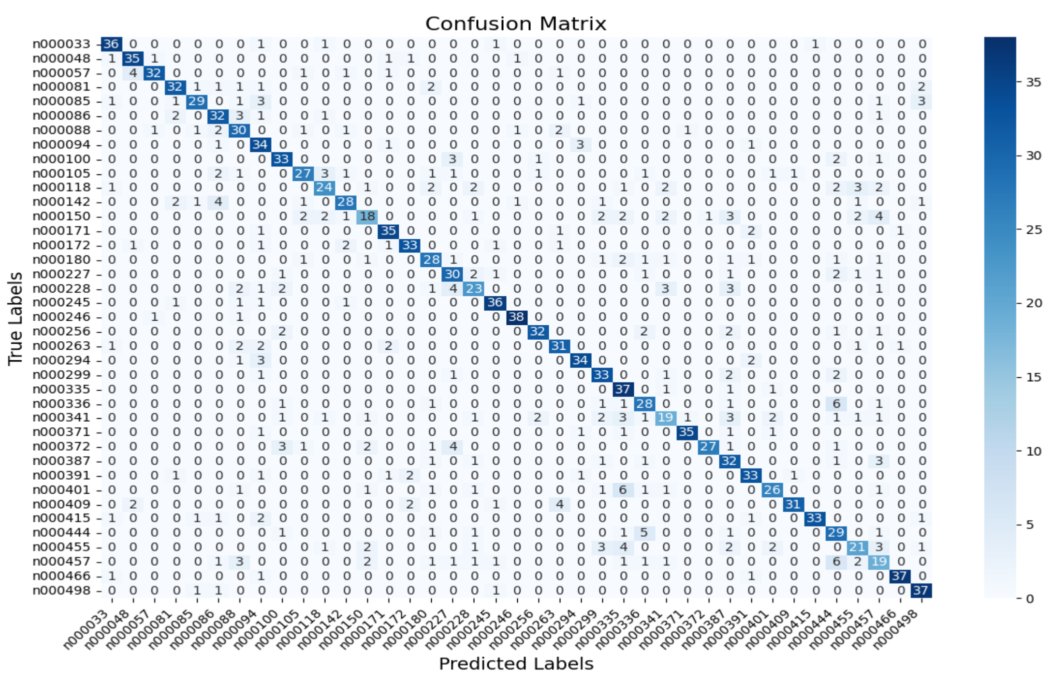


4.2 Confusion Matrix

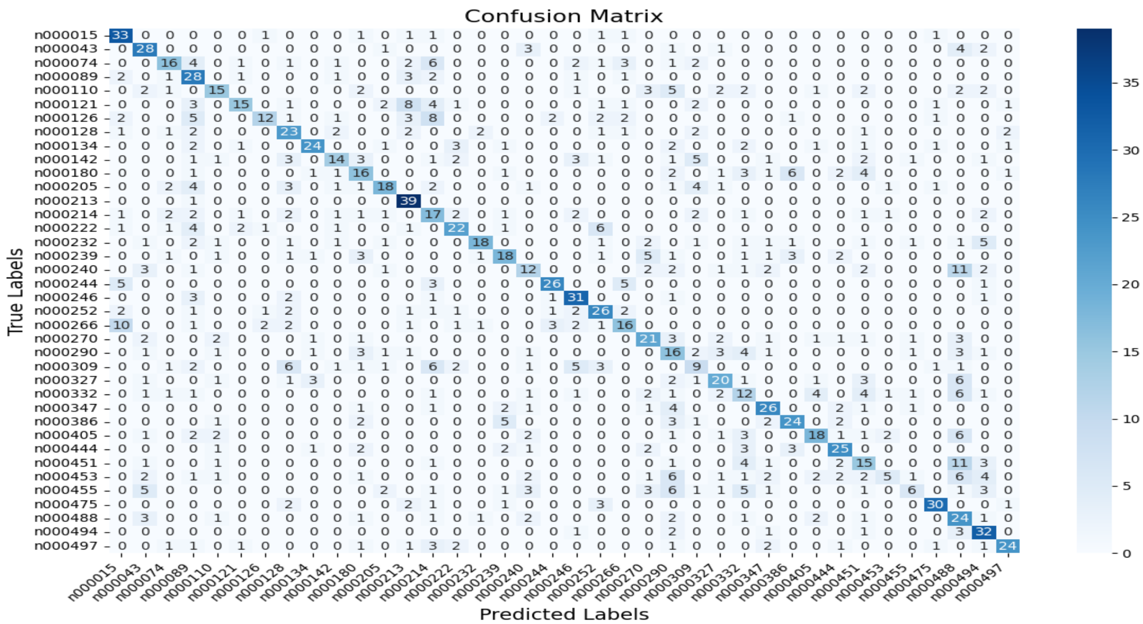
4.2.1 ResNet18



4.2.2 DenseNet

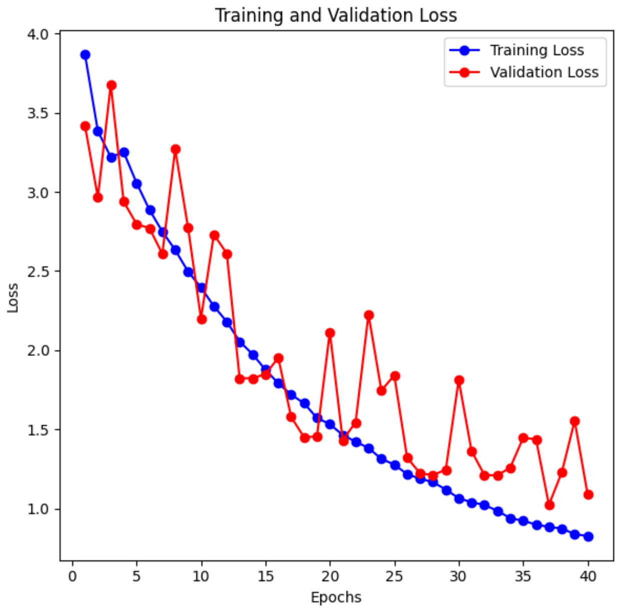


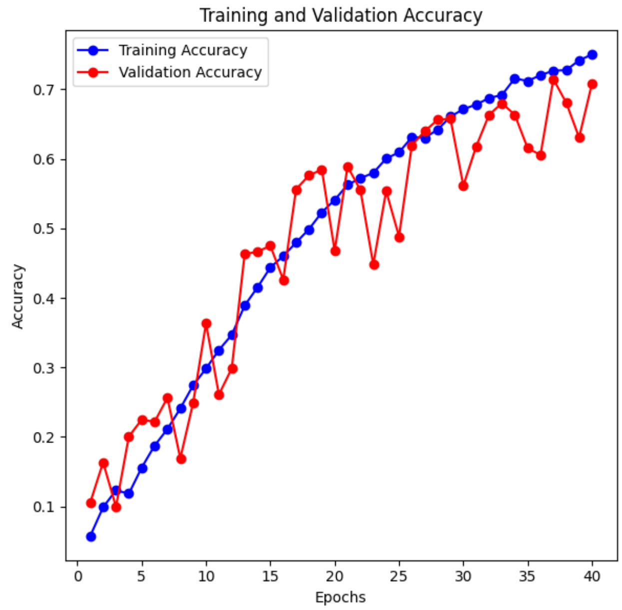
4.2.3 Xception



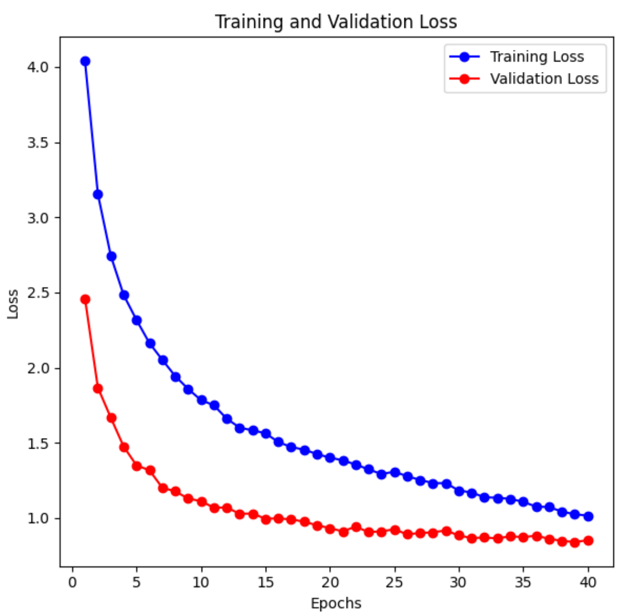
4.3 Accuracy and loss curves

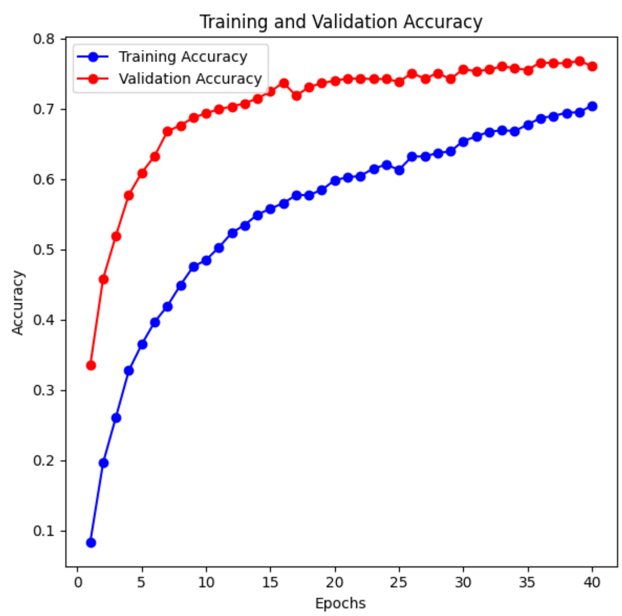
4.3.1 ResNet18



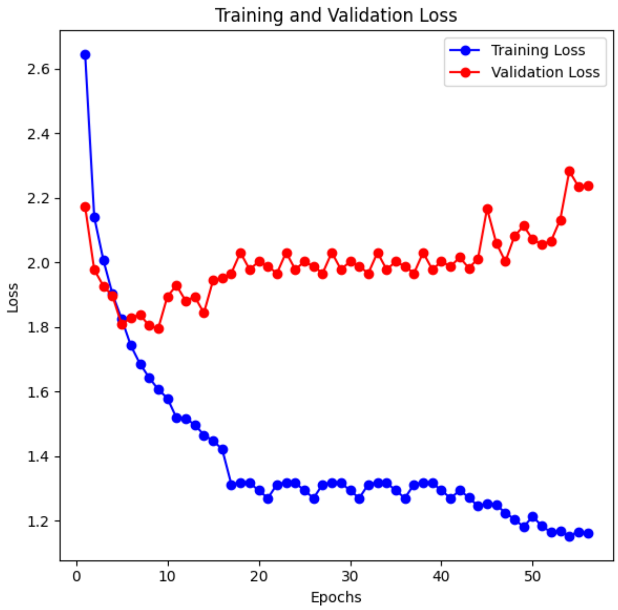


4.3.2 DenseNet



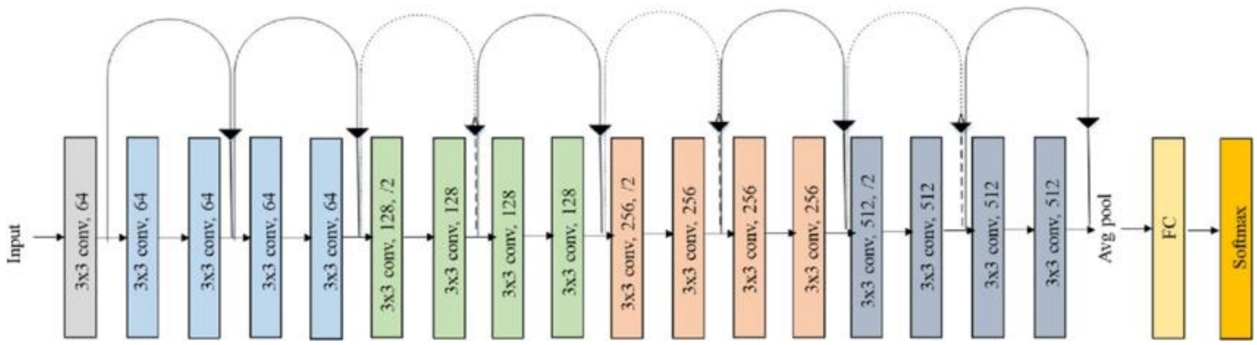


4.3.3 Xception

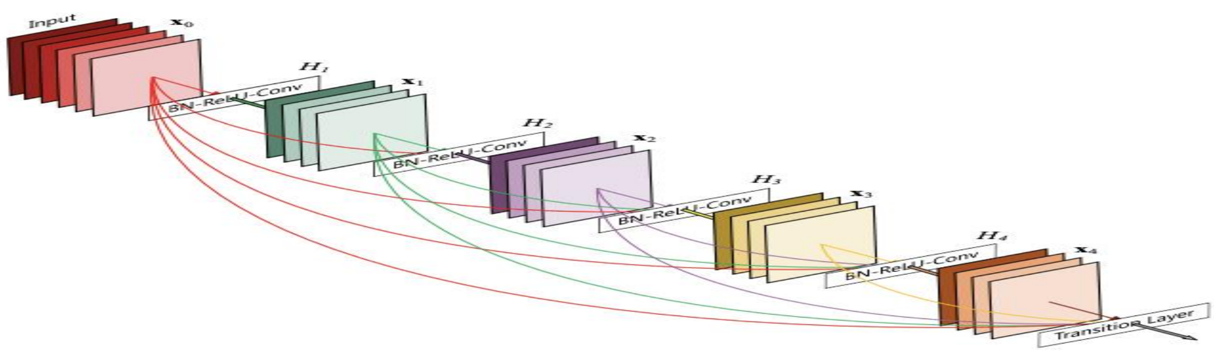


4.4 Architectures

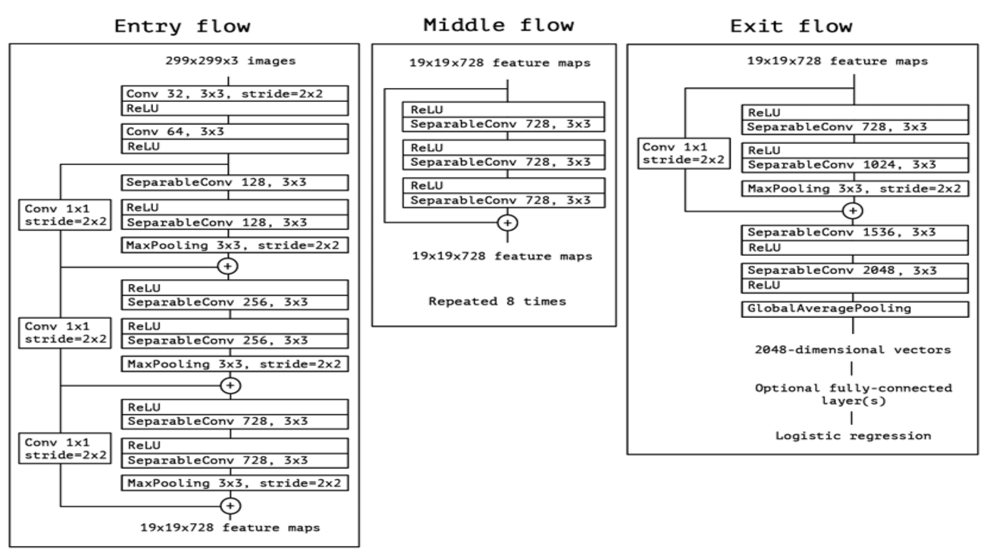
4.4.1 ResNet18



4.4.2 DenseNet



4.4.3 Xception



4.5 Classification report

4.5.1 ResNet

... Classification Report:

	precision	recall	f1-score	support
n000024	0.61	0.57	0.59	40
n000030	0.80	0.90	0.85	40
n000041	0.76	0.80	0.78	40
n000071	0.72	0.70	0.71	40
n000072	0.80	0.40	0.53	40
n000086	0.86	0.60	0.71	40
n000094	0.86	0.78	0.82	40
n000115	0.76	0.47	0.58	40
n000117	0.86	0.93	0.89	40
n000118	0.70	0.57	0.63	40
n000119	0.63	0.72	0.67	40
n000128	0.81	0.85	0.83	40
n000131	0.87	0.97	0.92	40
n000132	0.58	0.78	0.67	40
n000144	0.82	0.68	0.74	40
n000160	0.76	0.78	0.77	40
n000191	0.95	0.95	0.95	40
n000210	0.73	0.40	0.52	40
n000213	0.80	0.90	0.85	40
n000218	0.79	0.57	0.67	40
n000228	0.69	0.78	0.73	40
n000236	0.48	0.70	0.57	40
...				
accuracy			0.71	1560
macro avg	0.73	0.71	0.71	1560
weighted avg	0.73	0.71	0.71	1560

4.5.2 Dense

Classification Report:

	precision	recall	f1-score	support
n000033	0.8571	0.9000	0.8780	40
n000048	0.8333	0.8750	0.8537	40
n000057	0.9143	0.8000	0.8533	40
n000081	0.8205	0.8000	0.8101	40
n000085	0.8529	0.7250	0.7838	40
n000086	0.7111	0.8000	0.7529	40
n000088	0.6383	0.7500	0.6897	40
n000094	0.6182	0.8500	0.7158	40
n000100	0.7500	0.8250	0.7857	40
n000105	0.7941	0.6750	0.7297	40
n000118	0.7273	0.6000	0.6575	40
n000142	0.8000	0.7000	0.7467	40
n000150	0.6429	0.4500	0.5294	40
n000171	0.8333	0.8750	0.8537	40
n000172	0.8684	0.8250	0.8462	40
n000180	0.7000	0.7000	0.7000	40
n000227	0.6667	0.7500	0.7059	40
n000228	0.6765	0.5750	0.6216	40
n000245	0.8571	0.9000	0.8780	40
n000246	0.9268	0.9500	0.9383	40
n000256	0.8889	0.8000	0.8421	40
n000263	0.7750	0.7750	0.7750	40
...				
accuracy			0.7609	1560
macro avg	0.7709	0.7609	0.7606	1560
weighted avg	0.7709	0.7609	0.7606	1560

4.5.3 Xception

```
...  Classification Report:
              precision    recall  f1-score   support

     n000015     0.5789    0.8250    0.6804        40
     n000043     0.5490    0.7000    0.6154        40
     n000074     0.5714    0.4000    0.4706        40
     n000089     0.3889    0.7000    0.5000        40
     n000110     0.5172    0.3750    0.4348        40
     n000121     0.6818    0.3750    0.4839        40
     n000126     0.7059    0.3000    0.4211        40
     n000128     0.4510    0.5750    0.5055        40
     n000134     0.7500    0.6000    0.6667        40
     n000142     0.5600    0.3500    0.4308        40
     n000180     0.4000    0.4000    0.4000        40
     n000205     0.6207    0.4500    0.5217        40
     n000213     0.6000    0.9750    0.7429        40
     n000214     0.2787    0.4250    0.3366        40
     n000222     0.6111    0.5500    0.5789        40
     n000232     0.7826    0.4500    0.5714        40
     n000239     0.5455    0.4500    0.4932        40
     n000240     0.4000    0.3000    0.3429        40
     n000244     0.7879    0.6500    0.7123        40
     n000246     0.6078    0.7750    0.6813        40
     n000252     0.5417    0.6500    0.5909        40
     n000266     0.5000    0.4000    0.4444        40
...
    accuracy                         0.5092      1520
   macro avg     0.5380    0.5092    0.5035      1520
weighted avg     0.5380    0.5092    0.5035      1520
```

Comparison of ResNet, Xception, and DenseNet Models

Pros and Cons of Each Architecture

1. **ResNet**
   - **Pros:**
     - **Introduces skip connections to avoid vanishing gradients.**
     - **Performs well on large datasets with deep networks.**
     - **Relatively easy to train compared to other deep architectures.**
   - **Cons:**
     - **High computational cost due to the number of layers.**
     - **Potential overfitting on smaller datasets without proper regularization.**
   - **Best for:**

- Applications needing robust baseline architectures, e.g., medical imaging or traffic accident severity.

2. **DenseNet**

- **Pros:**
  - **Dense connections enhance gradient flow and feature reuse.**
  - **Typically achieves state-of-the-art performance with fewer parameters.**
  - **Reduces the risk of overfitting through feature reuse.**
- **Cons:**
  - **High memory usage due to feature concatenation.**
  - **Longer training times with larger architectures.**
- **Best for:**
  - **Situations requiring efficient use of network parameters, such as medical image classification or detailed texture analysis.**

3. **Xception**

- **Pros:**
  - **Uses depthwise separable convolutions, reducing parameter count.**
  - **Excellent for fine-grained visual tasks due to efficient feature extraction.**
  - **High accuracy and precision for larger datasets with complex features.**
- **Cons:**
  - **Requires more computational resources for training.**
  - **Slower training compared to ResNet.**
- **Best for:**
  - **Fine-grained image classification tasks, e.g., identifying specific bone fractures in X-ray analysis.**

models Comparison

	RESNET	DENSENET	XCEPTION
PROS	• - Prevents vanishing gradient problem using residual connections. • - Efficient with deep layers. • - Good generalization.	• - Efficient parameter use via feature reuse. • - Mitigates vanishing gradients. • - Strong performance on smaller datasets.	• - High performance on large datasets. • - Efficient depthwise separable convolutions reduce computation. • - Good for transfer learning.
CONS	• - Computationally expensive for very deep networks. • - May overfit on small datasets.	• - Computationally expensive due to concatenating feature maps. • - Larger memory footprint.	• - Requires large labeled datasets. • - Tends to underperform on small datasets without fine-tuning.
SUITABLE DATASETS	• - Large image datasets (e.g., ImageNet). • - Moderate-size datasets with clear features.	• - Small to medium image datasets. • - Medical imaging or datasets with complex features.	• - Large-scale datasets (e.g., ImageNet, COCO). • - Datasets with subtle spatial patterns.
PROBLEMS	• - Object detection. • - Image classification. • - Feature extraction	• - Medical imaging (e.g., X-ray analysis). • - Fine-grained classification tasks.	• - Image segmentation. • - Fine-grained image classification. • - Transfer learning tasks.
ACCURACY	• 0.71	• 0.7609	• 0.5092
PRECISION	• 0.73	• 0.7709	• 0.5380
RECALL	• 0.71	• 0.7609	• 0.5092
F1-SCORE	• 0.71	• 0.7606	• 0.5103

5.GitHub repository

Link: [https://github.com/gamal101/DeepLearningModels/tree/main]

6.References

ResNet: [https://arxiv.org/abs/1512.03385]

DenseNet: [https://arxiv.org/abs/1608.06993]

Xception: [https://arxiv.org/abs/1610.02357]