ResNet:-

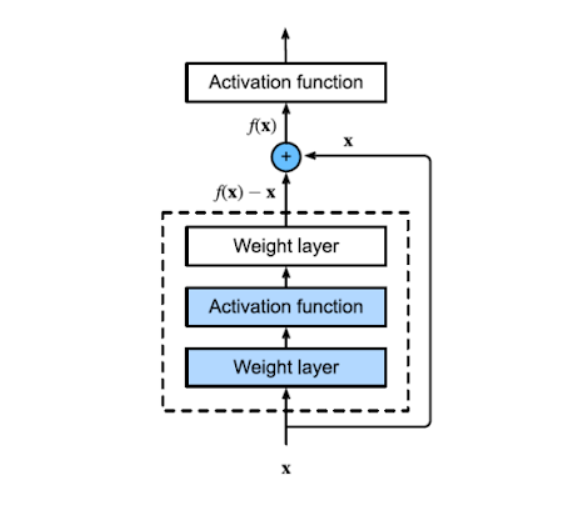
ResNet (Residual Network), introduced in 2015 by Kaiming He et al., is a deep neural network architecture designed to address the issue of vanishing gradients in very deep networks. Its key innovation is the use of residual connections, allowing the network to skip layers and focus on learning the differences (residuals) between inputs and outputs, making it easier to train deep models.

ResNet is widely used in computer vision tasks like image classification and object detection, achieving state-of-the-art performance on benchmark datasets. Variants of ResNet include ResNet-50, ResNet-101, and ResNet-152, differing in depth and the number of layers.

Architecture:

ResNet architecture uses **residual connections** to skip layers and pass data directly between them, enabling efficient training of deep networks. Its core building block, the **residual block**, consists of 2-3 convolutional layers combined with residual connections, allowing the network to focus on learning the differences between input and output.

The overall structure includes multiple residual blocks, a global average pooling layer, and a fully connected layer for classification. Variants may include additional components like batch normalization or dropout layers to enhance performance and reduce overfitting.



* F(x) is the desired underlying mapping that we are trying to learn in order to serve as the input for the activation function above.
* The box with the dotted line must learn the residual mapping f(x) - x.
* A residual connection is the solid line that carries the layer input x to the addition operator.
* Two 3x3 convolutional layers with the same number of output channels make up the n Block. A batch normalizing layer and a [ReLU activation function](https://deeplearningofpython.blogspot.com/2023/04/Activation%20Functions-Neural%20Network.html" \t "_blank) are then applied on top of it.

**Working:**

ResNet uses **residual connections** to train very deep neural networks by allowing information to skip layers, preventing gradients from vanishing during backpropagation. The network learns the **residual function** (differences between inputs and outputs) rather than the complete mapping, focusing on the challenging aspects of the task while residual connections handle simpler parts.

Through backpropagation, gradients are computed to adjust weights and minimize the loss function, enhancing performance. ResNet is highly effective for computer vision tasks like image classification and object recognition, becoming a widely used architecture that advances deep learning research and applications.

Applications:

For computer vision applications including image classification, object recognition, and semantic segmentation, the ResNet architecture has been extensively employed. Here are a few examples of ResNet's particular uses:

* **Image Classification:** ResNet performed at the cutting edge on the ImageNet Large Scale Visual Recognition Challenge (ILSVRC), among other benchmark datasets for image classification.
* **Object detection** models like F**aster R-**[**CNN**](https://deeplearningofpython.blogspot.com/2023/04/CNN-Architecture-example-Deeplearning.html)**, Mask R-CNN, and YOLOv3** have all employed ResNet as their underlying architecture. On the COCO object detection dataset, these models performed at the cutting edge.
* ResNet has been used for **semantic segmentation** tasks, in which a class label is intended to be assigned to each pixel in an image. Modern performance has been attained by ResNet-based models on datasets like Cityscapes and PASCAL VOC.
* ResNet has been employed as a pre-trained **feature extractor** for transfer learning, in which a model is initially trained on a huge dataset and then [fine-tuned](https://deeplearningofpython.blogspot.com/2023/04/Hyperparametertuning-Deeplearning-example.html) on a smaller dataset for a particular task. On a variety of **computer vision** tasks, state-of-the-art performance has been achieved using this method.

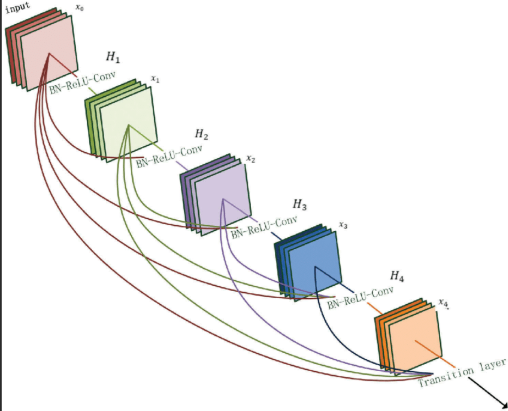
Overall, the ResNet design has proven to be quite successful for a variety of computer vision tasks, making it one of the most popular architectures in deep learning research and applications.

DenseNet

DenseNet (Dense Convolutional Network), introduced in 2017 by Huang et al., connects each layer to every other layer in a feedforward manner, unlike traditional architectures that link only adjacent layers. These **dense connections** reduce parameters, improve information flow, and help mitigate the vanishing gradient problem, enabling efficient training and learning of complex representations.

DenseNet consists of **dense blocks** with multiple layers connected densely and **transition layers** that reduce the number of channels and spatial dimensions before passing data to the next block. It has achieved state-of-the-art results in image classification benchmarks like ImageNet and has been effectively applied to tasks like object detection and segmentation.

Architecture :



The DenseNet architecture consists of transition layers that decrease the spatial dimensions of the feature maps and the number of channels, as well as various dense blocks that are made up of numerous densely connected layers

* An image-representing 3D tensor serves as the network's input.
* **The first layer of convolution:**This layer applies a convolution operation on the input image to extract features.
* The network consists of a number of dense blocks, each of which is composed of numerous densely connected layers. A dense block of layers uses the feature maps created by all of the layers before it as input.
* Between each pair of dense blocks, the network has a transition layer that, by combining convolution and pooling processes, shrinks the spatial dimensions of the feature maps and the number of channels.
* The feature maps are then sent through a global [average pooling layer](https://deeplearningofpython.blogspot.com/2023/04/CNNlayers-implementation-python.html), which creates a fixed-length feature vector for the entire image, after the last dense block.
* The final layer is a fully linked layer with [softmax activation](https://deeplearningofpython.blogspot.com/2023/04/Activation%20Functions-Neural%20Network.html" \t "_blank) that generates a probability distribution across all conceivable classes as the output layer.
* Each dense block's layers are connected to one another in a feedforward fashion, meaning that each layer's output serves as an input to all levels after it. The network can learn more efficient and compact representations thanks to its dense connection, which also makes it easier to reuse features, which can improve generalization.
* DenseNet is an extremely flexible architecture that can be applied to a variety of computer vision problems and has been demonstrated to deliver state-of-the-art results on a number of benchmarks.

Working:

* A 3D tensor that represents an image serves as the network's input.
* **Convolutional layers:** To extract features, the input image is passed through several convolutional layers. Through the use of convolution, pooling, and activation functions, these features are learned.
* Multiple dense blocks are present in the network, each of which is composed of several densely connected layers. Each layer in a dense block creates a new set of feature maps as an output while utilizing the feature maps created by all earlier layers in the block as input. The network can learn more effective representations and promote feature reuse thanks to its dense connectivity.
* Each pair of dense blocks in the network has a transition layer in between them, which lowers the number of channels and the spatial dimensions of the feature maps. The network's computing cost is decreased, and overfitting is avoided, by performing this via a combination of convolution and pooling processes.
* **Global average pooling layer:**The feature maps are then subjected to a global average pooling layer after the last dense block, which results in a fixed-length feature vector for the entire image. As a result, it is easier to aggregate features throughout the entire image and the spatial dimensions of the feature maps are reduced to a size that is manageable.
* **Layer that generates the output:** The last layer is a fully linked layer with softmax activation, which creates a probability distribution across the possible classes.
* The network modifies the [convolutional layer](https://deeplearningofpython.blogspot.com/2023/04/CNNlayers-implementation-python.html) and fully connected layer weights during training to reduce the discrepancy between the predicted class probabilities and the actual class labels. [Gradient descent optimization](https://deeplearningofpython.blogspot.com/2023/04/Optimizationalgorithm-types-Deeplearning.html) and [backpropagation](https://deeplearningofpython.blogspot.com/2023/04/Stochastic%20Gradient%20Descent-Deep%20Learning.html) are used for this.
* In general, DenseNet is a strong and adaptable architecture that has been demonstrated to provide state-of-the-art performance on a variety of computer vision applications.

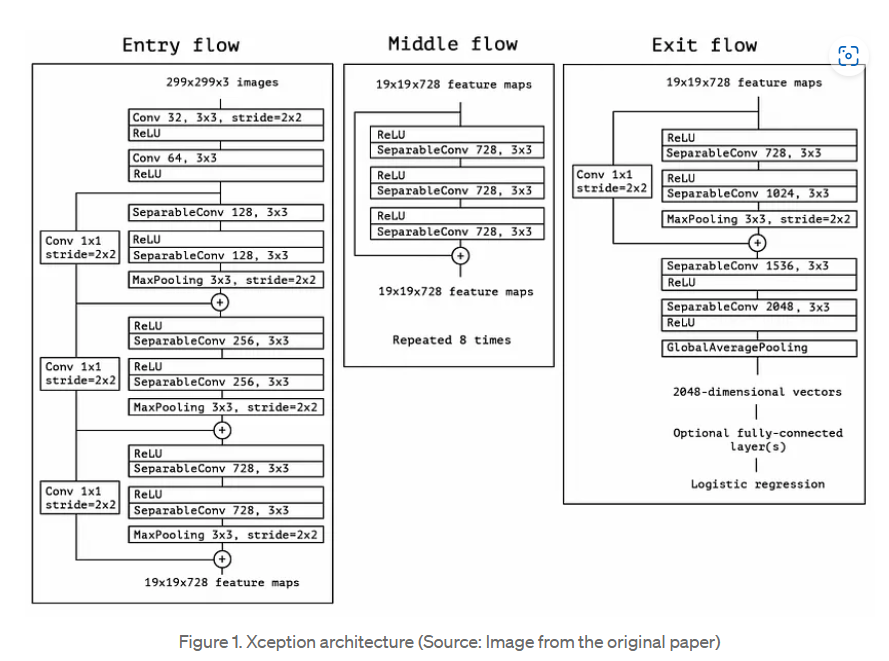
Applications:

DenseNet has been utilized for a variety of computer vision tasks and has proven to perform at the cutting edge on a number of benchmarks. The following are some of the uses for DenseNet:

* DenseNet has been used for image classification tasks, including the categorization of images into several groups. On a number of benchmarks, including the **ImageNet Large Scale Visual Recognition Challenge (ILSVRC)** dataset, DenseNet has demonstrated state-of-the-art performance.
* DenseNet has been used for jobs involving object detection, where the objective is to find and identify items in an image. DenseNet-based object detectors have attained cutting-edge results on a number of benchmarks, including the COCO dataset.
* DenseNet has been used for **semantic segmentation** tasks, where the objective is to give each pixel in an image a class label. Modern performance has been attained by DenseNet-based semantic segmentation models on a number of benchmarks, including the Cityscapes dataset.
* **Image captioning:** Tasks requiring the creation of a natural language description of an image have been carried out using DenseNet. Modern performance has been attained by DenseNet-based image captioning models on a number of benchmarks, including the MSCOCO dataset.
* **Medical image analysis:** Tasks involving medical image analysis, such as locating and segmenting tumors in medical pictures, have also been handled using DenseNet. DenseNet-based models have attained cutting-edge performance on a number of benchmarks, including the ISIC 2017 challenge for the categorization of melanoma.

Xception

Convolutional Neural Networks (CNN) have come a long way, from the LeNet-style, AlexNet, VGG models, which used simple stacks of convolutional layers for feature extraction and max-pooling layers for spatial sub-sampling, stacked one after the other, to Inception and ResNet networks which use skip connections and multiple convolutional and max-pooling blocks in each layer. Since its introduction, one of the best networks in computer vision has been the Inception network. The Inception model uses a stack of modules, each module containing a bunch of feature extractors, which allow them to learn richer representations with fewer parameters.



the Xception module has 3 main parts. The Entry flow, the Middle flow (which is repeated 8 times), and the Exit flow.

The entry flow has two blocks of convolutional layer followed by a ReLU activation. The diagram also mentions in detail the number of filters, the filter size (kernel size), and the strides.

There are also various Separable convolutional layers. There are also Max Pooling layers. When the strides are different than one, the strides are also mentioned. There are also Skip connections, where we use ‘ADD’ to merge the two tensors. It also shows the shape of the input tensor in each flow. For example, we begin with an image size of 299x299x3, and after the entry flow, we get an image size of 19x19x728.

Similarly, for the Middle flow and the Exit flow, this diagram clearly explains the image size, the various layers, the number of filters, the shape of filters, the type of pooling, the number of repetitions, and the option of adding a fully connected layer in the end.

Also, all Convolutional and Separable Convolutional layers are followed by batch normalization.

References:

<https://deeplearningofpython.blogspot.com/2023/04/DenseNet-Architecture-Keras-Applications.html>

[ResNet Architecture with Keras](https://deeplearningofpython.blogspot.com/2023/04/ResNet-Architecture-Keras.html)

[Xception: Implementing from scratch using Tensorflow | by Arjun Sarkar | Towards Data Science](https://medium.com/towards-data-science/xception-from-scratch-using-tensorflow-even-better-than-inception-940fb231ced9)

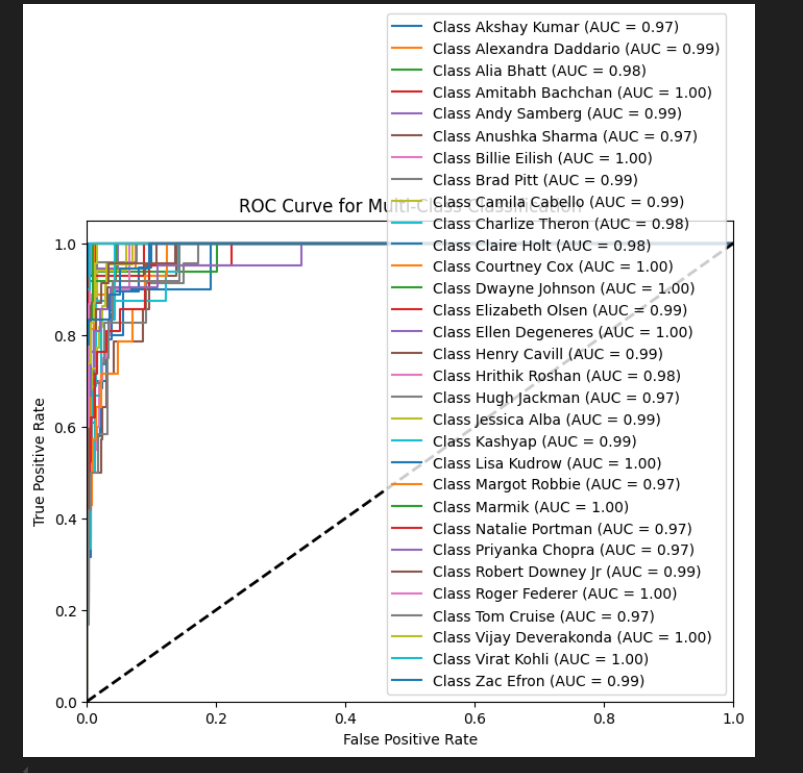
[[1610.02357] Xception: Deep Learning with Depthwise Separable Convolutions](https://arxiv.org/abs/1610.02357)

1-Face Recognition Dataset:

1-Resnet Evaluation:

A screenshot of a computer screen

Description automatically generated



2-DenseNet:

A screenshot of a computer screen

Description automatically generated

A screenshot of a computer

Description automatically generated

3-Xception:

A graph on a computer screen

Description automatically generated

A graph of loss and loss

Description automatically generated

A graph with different colored lines

Description automatically generated

-The Evaluation of models for this Data is : **ResNet** demonstrated superior performance compared to the other models, achieving the highest validation accuracy (~79.14%) while maintaining a low validation loss. This makes ResNet the most suitable choice for this dataset. However, in scenarios where computational resources are constrained or training time is a critical factor, **Xception** offers a slightly less accurate yet robust alternative.

**DenseNet**, on the other hand, appears to be under-optimized. It may require further fine-tuning, such as adjustments to the learning rate, improvements in data preprocessing, or modifications to its architectural parameters.

2-pins-face-Recognition:

1-Resnet Evaluation:

A graph with blue and orange lines

Description automatically generated

A graph with a line

Description automatically generated

A graph with lines and lines

Description automatically generated with medium confidence

2-Densenet:

A screenshot of a graph

Description automatically generated

A graph with numbers and lines

Description automatically generated

3-Xception:

A graph showing a line

Description automatically generated

A graph of a person with a blue line

Description automatically generated with medium confidence

A screen shot of a graph

Description automatically generated

Evaluation of models for this data:

Based on the results, **DenseNet** is the best-performing model for this dataset, achieving higher validation accuracy and lower loss within fewer epochs. Its efficient training and robust performance make it the optimal choice.

Thank You