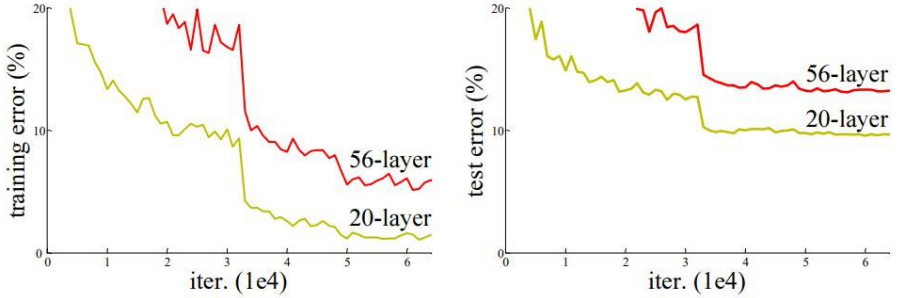
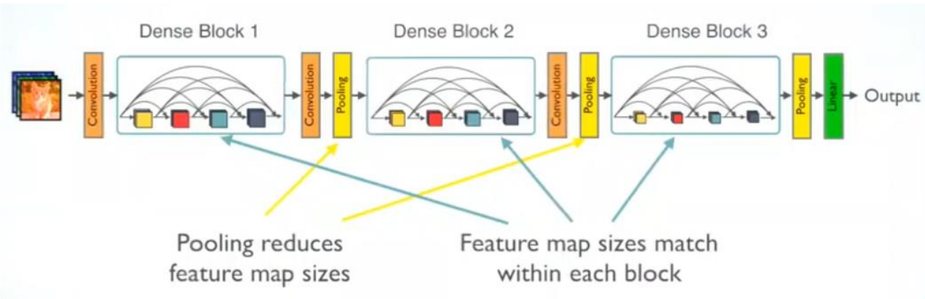
Name :- Ramishka Madhushan

Assignment No :- 01

MODEL	Pro	Cons	High level structure of the algorithm
RESNET (Residual Neural Network)	• It introduced the concept of residual learning, which helps alleviate the vanishing gradient problem in deep neural networks. • Achieves state-of-the-art accuracy on various computer vision tasks. (Object detection, Image classification) • Allows training of much deeper networks without degradation in performance.	• It may be prone to overfitting when dealing with small datasets. • It required more computational resources and memory due to its increased depth compared to other models.	• It utilizes skip connections, or "identity shortcuts," to allow the flow of information from earlier layers directly to later layers. • It consists of residual blocks containing multiple convolutional layers followed by batch normalization and ReLU activation. • The input is passed through multiple residual blocks, and the output is finally fed into fully connected layers for classification.  *Figure 01 :- Comparison of 20-layer vs 56-layer architecture*

DensNet (Densely Connected Convolutional Network)	• It promotes feature reuse and strengthens feature propagation by connecting each layer to every other layer in a dense manner. • It reduces the number of parameters compared to other architectures, improving parameter efficiency. • It achieves competitive performance with fewer layers, making it computationally efficient.	• DenseNet may suffer from increased memory requirements due to the dense connections, especially in deep networks. • DenseNet can be more susceptible to overfitting than other models, mainly when dealing with small datasets.	• DenseNet is composed of dense blocks, where each block contains multiple convolutional layers. • The input of each layer is the concatenation of the feature maps from all preceding layers. • Transition layers, consisting of convolutional and pooling layers, are used to downsample feature maps and control the growth of dimensions. • The output of the last dense block is fed into global average pooling, followed by fully connected layers for classification.  *Figure 02 :- DensNet Architecture*

<table>
<tr><th>MODEL</th><th>Pro</th><th>Cons</th><th>High level structure of the algorithm</th></tr>
<tr>
<td>VGG (Visual Geometry Group)</td>
<td>

VGG has a simple and uniform architecture, making it easy to understand and implement.
It has achieved outstanding performance on various image recognition tasks and serves as a strong baseline for CNN models.
VGG's design principles have influenced subsequent architectures and research in deep learning.

</td>
<td>

VGG has many parameters, which leads to increased memory requirements and slower training and inference times.
It may struggle with overfitting when dealing with small datasets due to its high capacity.

</td>
<td>

VGG consists of multiple convolutional layers stacked on top of each other, with smaller 3x3 filters used extensively.
Max pooling layers are employed to down sample feature maps.
The network ends with fully connected layers for classification.
VGG is characterized by its depth, with variations like VGG16 and VGG19 having 16 and 19 layers, respectively.

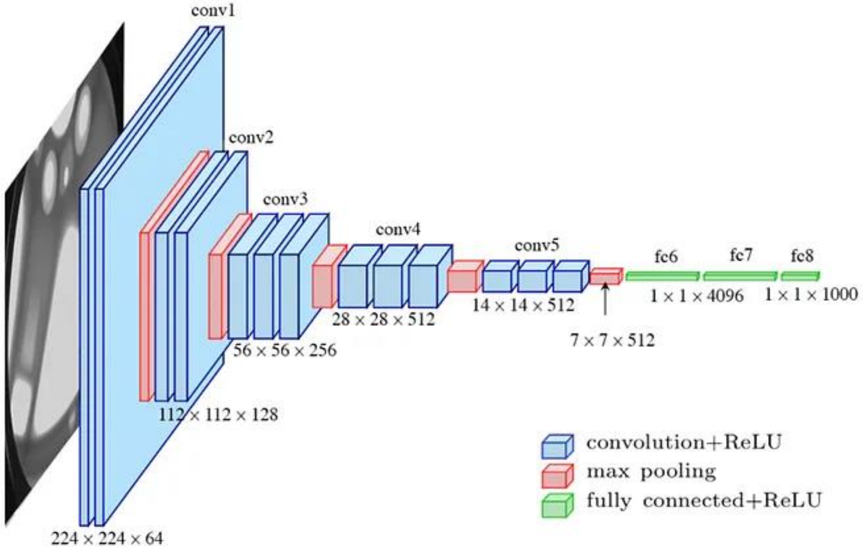


Figure 03 :- VGG-Net Architecture
</td>
</tr>
</table>

<table>
<tr>
<td>Inception (GoogLeNet / Inception-v3)</td>
<td>

Inception incorporates the use of "Inception modules," which employ multiple filter sizes in parallel, capturing different scales of information.
It achieves excellent performance with fewer parameters compared to other architectures.
Inception models are known for their computational efficiency and are often used in resource-constrained environments.

</td>
<td>

Inception networks can be challenging to train due to the complexity of their architecture.
They require careful tuning of hyperparameters to achieve optimal performance.

</td>
<td>

Inception models consist of a chain of Inception modules, where each module has parallel convolutional branches with different filter sizes.
These branches are combined by concatenation of their output feature maps.
Inception networks also incorporate auxiliary classifiers at intermediate layers to provide additional regularization during training.
The final output is obtained through global average pooling and fully connected layers.

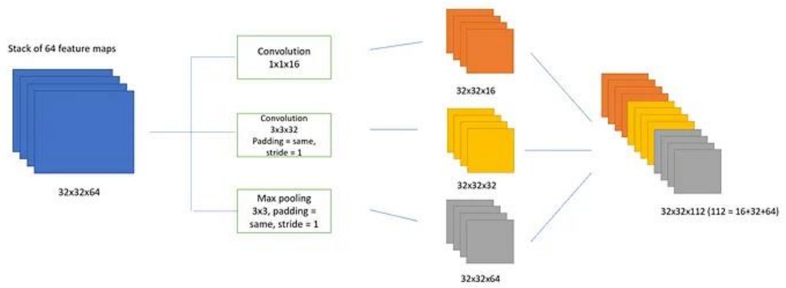
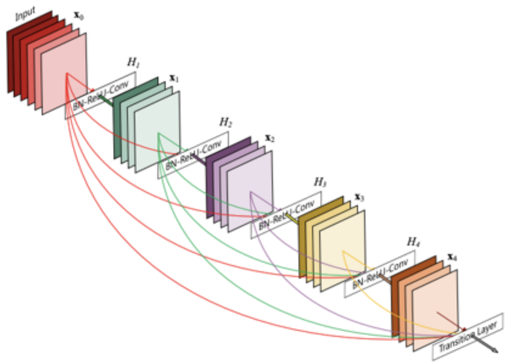
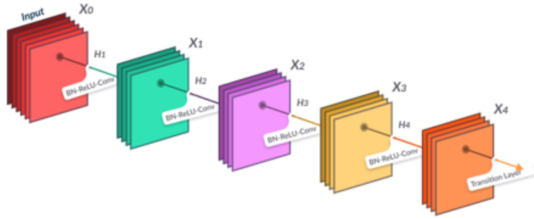


Figure 04 :- VGG-Net Architecture
</td>
</tr>
</table>





DenseNet Structure

ResNet Structure

$$a^{[l]} = g\left([a^{[0]}, a^{[1]}, a^{[2]}, \dots \dots \dots \dots, a^{[l-1]}\right)$$

$$a^{[l]} = g\left(z^{[l+1]} + a^{[l]}\right)$$

Figure 05 :- DenseNet vs ResNet Structure