Training Models

- A large amount of data is necessary to achieve optimal accuracy in a neural network. The data needs to be trained for many hours on a powerful Graphics Processing Unit (GPU) to achieve results. With the advent of transfer learning, significant changes have occurred in deep neural network learning processes. Models pretrained on large datasets like ImageNet enhance transfer learning. Transfer learning works by freezing initial hidden layers of the model and fine-tuning the final layers. The frozen state of a layer indicates that it will not be trained, and consequently, its weights will not change. Due to the relatively small dataset used in this study with limited images per class, transfer learning is most suitable. Pretrained models used in this study will be further explained in the subsection.

1. ResNet101.

- Proposed by He Kaiming, initially with 101 layers. Apart from a 7x7 convolutional layer and an FC layer, it arranges 3x3 convolutional layers in the sequence [3], [4], [23], [3], and performs dimensionality reduction at each residual unit. This network efficiently mitigates gradient vanishing. The overall model structure is illustrated in Figure 1.

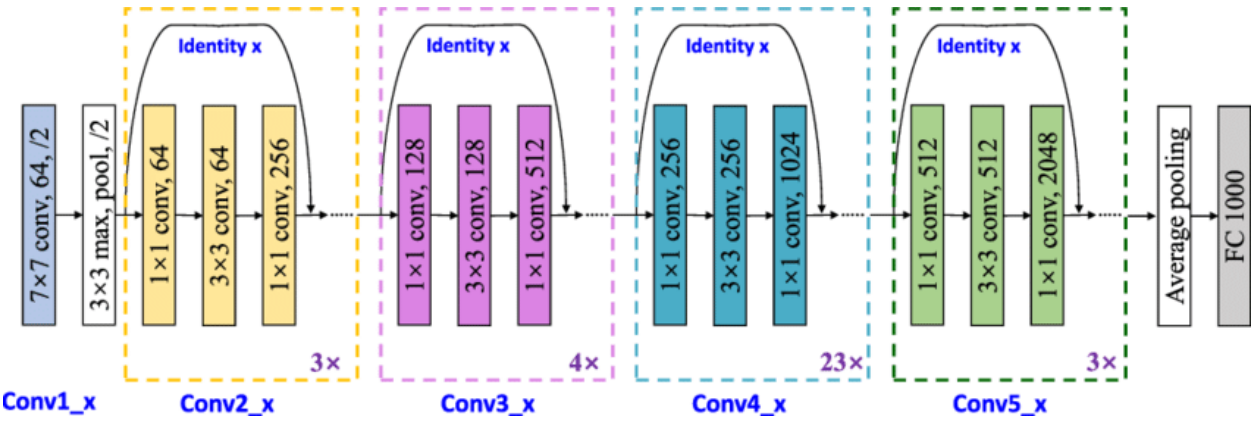

Figure 1. Resnet101 network architecture

2. ResNet50.

- ResNet-50 is a variant of the Residual Network (ResNet) architecture proposed by He Kaiming. It consists of 50 layers and was designed to address the degradation problem encountered in very deep networks by introducing residual connections. These connections allow for the training of much deeper networks effectively.

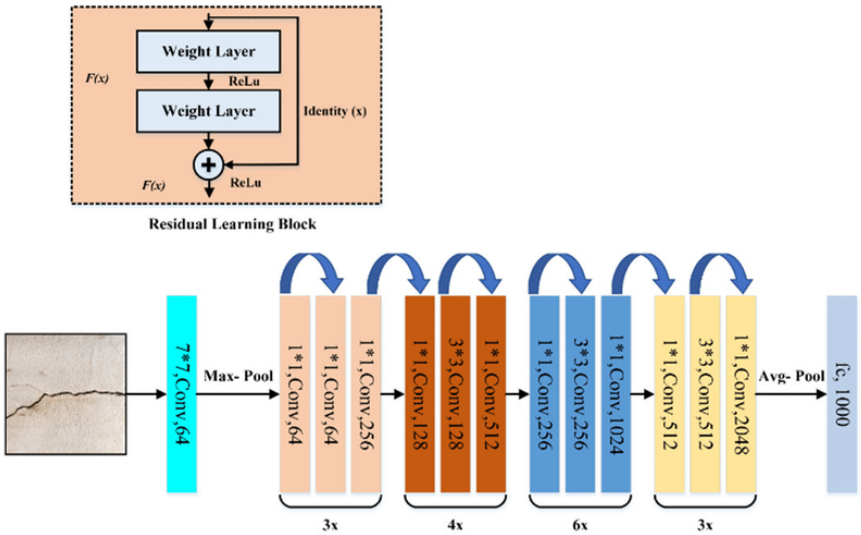

Figure 2. Resnet50 network architecture

3. ResNet152.

- ResNet-152 is a deeper variant of the ResNet architecture, also proposed by He Kaiming. It consists of 152 layers and is designed to further improve the feature learning capability of the network compared to ResNet-50.

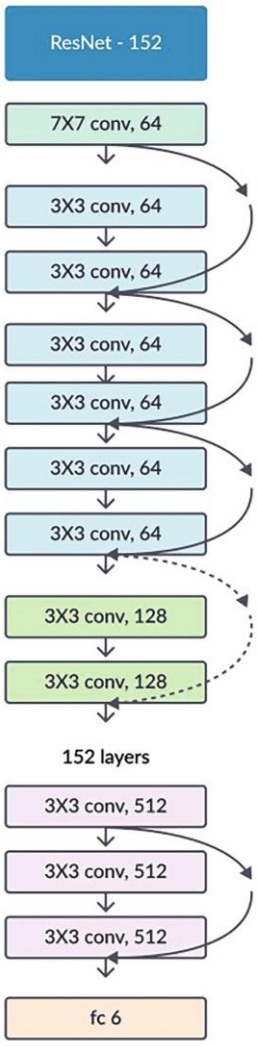

Figure 3. Resnet152 network architecture

4. CNN (Convolutional Neural Networks)

A neural network designed to process structured grid data like images. It includes convolutional layers, pooling layers, and fully connected layers. Convolutional layers extract features from input data using filters. Pooling layers reduce data size while retaining important features. Fully connected layers learn complex correlations between features for classification or prediction. CNNs are effective in learning complex pattern recognition in image data and have been successfully applied in various applications such as image recognition, natural language processing, and computer vision. The overall model structure is represented in Figure 4.

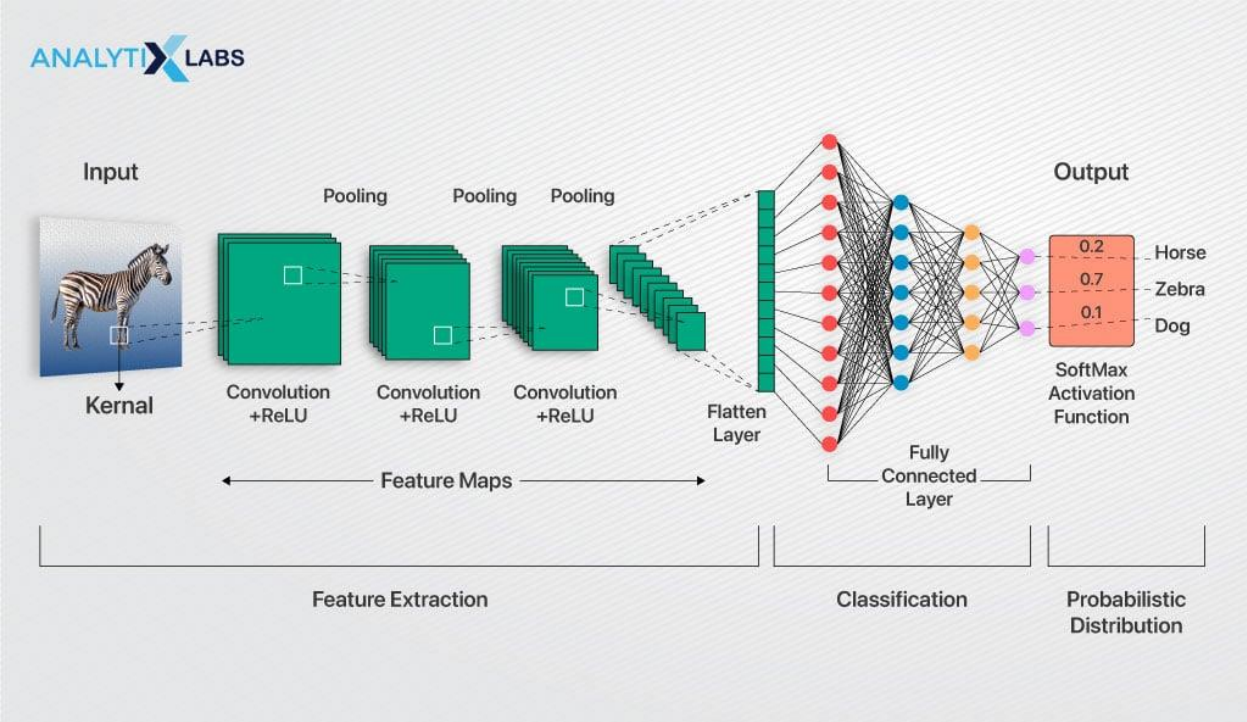

Figure 4. CNN network's architecture

5. VGG16

- VGG (Visual Geometry Group), a convolutional neural network architecture developed by the Visual Geometry Group at the University of Oxford in 2014. This model has achieved top results on multiple benchmark datasets and is widely used in image classification tasks. VGG16 has a deep and complex design structure, consisting of multiple convolutional layers and fully connected layers. Convolutional layers are used to extract local features of the image, such as edges, corners, and color patterns. Pooling layers are used to reduce data size while preserving important features. Finally, fully connected layers learn complex correlations between features to perform image classification.

.

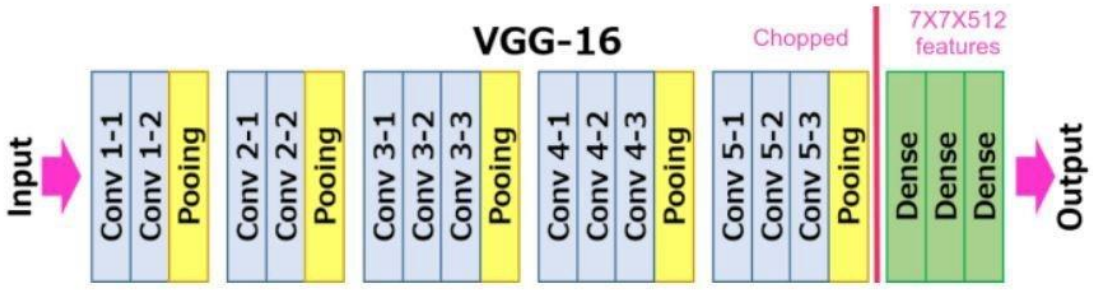

Figure 5: VGG network's architecture

6. DenseNet

- DenseNet is a neural network architecture introduced in 2016, characterized by dense connections where each layer is connected to every other layer. This allows the network to learn more complex features and improve performance in image classification tasks. DenseNet121 is a specific variant trained on the ImageNet dataset, achieving state-of-the-art results in ImageNet image classification tasks. With DenseNet121, the network can effectively learn complex features and efficiently identify patterns in images.

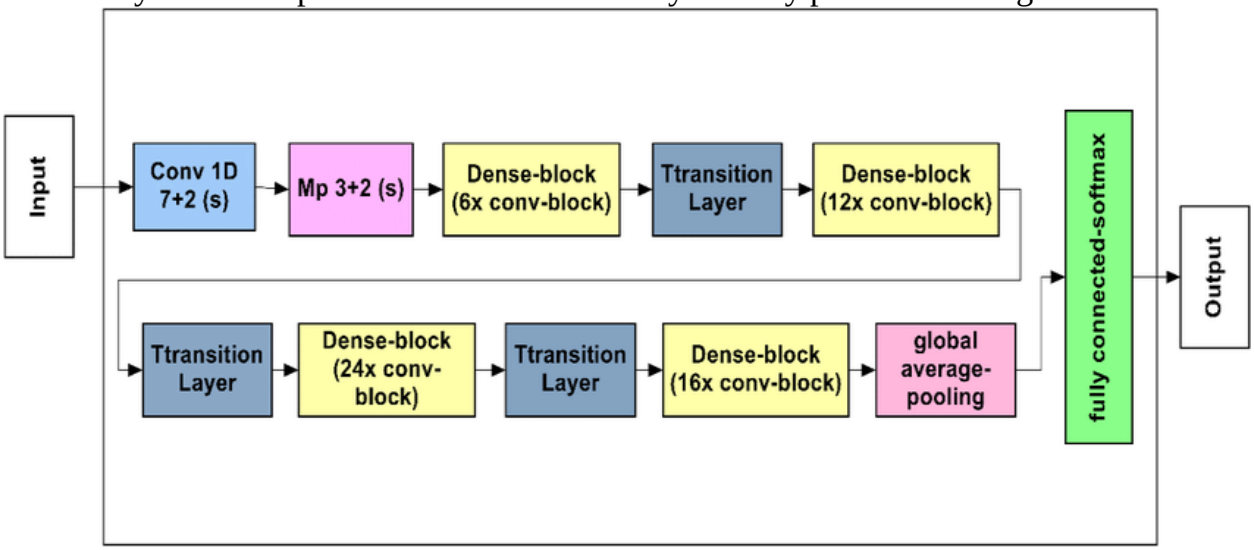

Figure 6: DenseNet network's architecture

7. CoatNet

- CoatNet is a convolutional neural network model designed specifically for image classification tasks. This model is built on a deep neural network architecture, aiming to accurately and efficiently recognize and classify objects in images. CoatNet's architecture includes multiple convolutional layers and pooling layers. Convolutional layers are used to extract local features of the image, such as edges, corners, and color patterns. Pooling layers are used to reduce data size and retain important features. By combining these convolutional and pooling layers, CoatNet can learn complex patterns in images and create an accurate feature representation of the image.

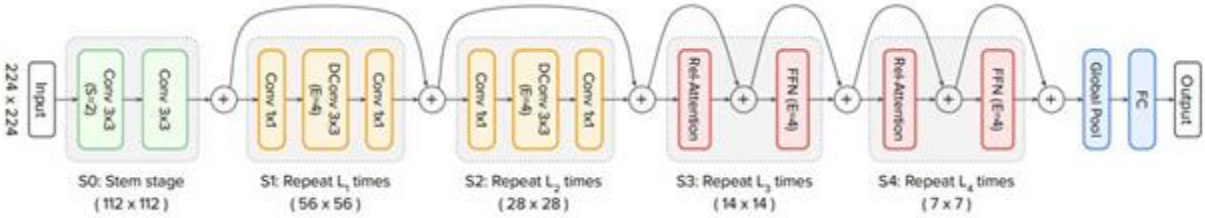

Figure 7: CoAtNet network's architecture

8. EfficientNet

- EfficientNet consists of a backbone network for extracting features from input images and a classification head for performing final classification. The backbone network is based on a sequence of convolutional layers and pooling layers, designed to achieve high performance and computational efficiency. The separated depthwise convolution layers help reduce the number of parameters in the network while enhancing the ability to learn complex features from images. The scaling method is used to automatically adjust the network size, thereby enhancing the model's performance and efficiency on different computational resources.

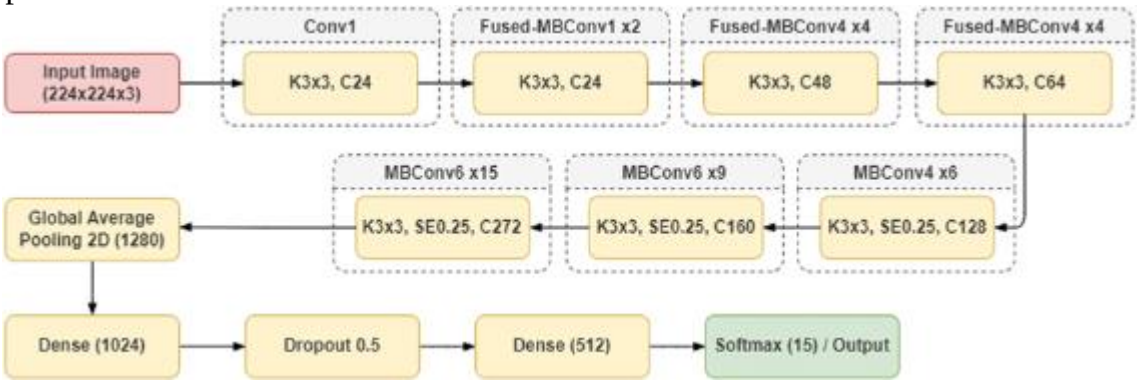

Figure 8: EfficientNetV2L network's architecture

9. PyramidNet

- PyramidNet is a deep learning model used to extract features from images using a pyramid-like structure, including convolutional layers and pooling layers. The convolutional layers are designed with different receptive field sizes, from small to large, to gather detailed and global information from the image. Using different receptive field sizes helps the model understand local and global features of the image. The output of each convolutional layer is then passed through a pooling operation to reduce the feature map size and help the model generate more abstract features. The pooling operation also helps reduce the computational complexity of the model. Finally, the output of PyramidNet is passed through a fully connected layer to map the extracted features to a probability distribution over possible image classification layer.

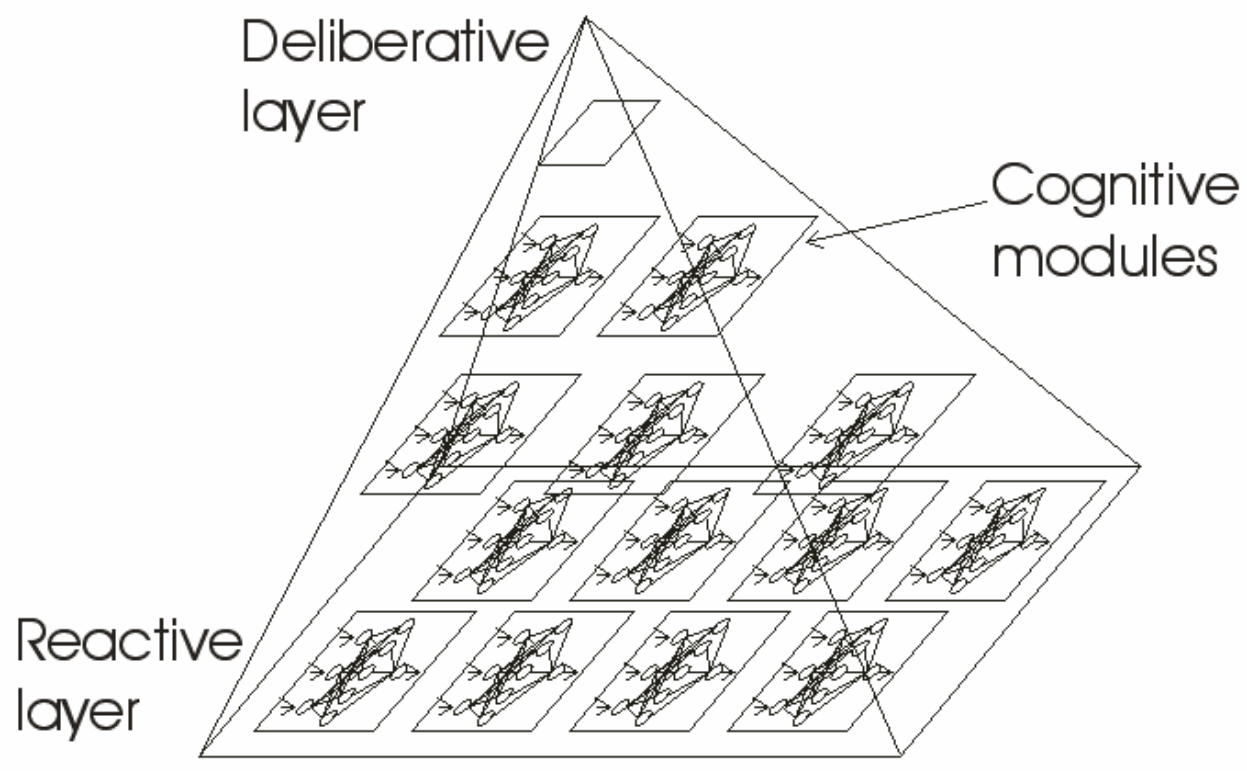

Figure 9: PyramidNet network's architecture

10. ViT-B16:

- The ViT (Vision Transformer) model is a neural network model designed based on the original Transformer architecture, addressing issues related to image classification and introducing a new approach to image processing using self-attention mechanisms. ViT's architecture is based on the core idea of Transformer, using self-attention layers to learn relationships and correlations between positions in the image. However, to apply Transformer to images, ViT divides the image into smaller patches, and each patch is treated as an input vector for the Transformer network, allowing the model to view the image as a sequence of vectors rather than individual pixels. ViT-B/16 is a specific variant of the ViT architecture, with "B" representing the model size and "16" representing the patch size in the image. ViT-B/16 uses a Transformer network with adjusted numbers of layers and patch sizes to achieve improved performance and computational efficiency. The Transformer layers in ViT-B/16 connect patches together and create an overall feature representation of the image. Finally, a fully connected layer is used to map the feature representation to possible classification layers, providing the final classification result.

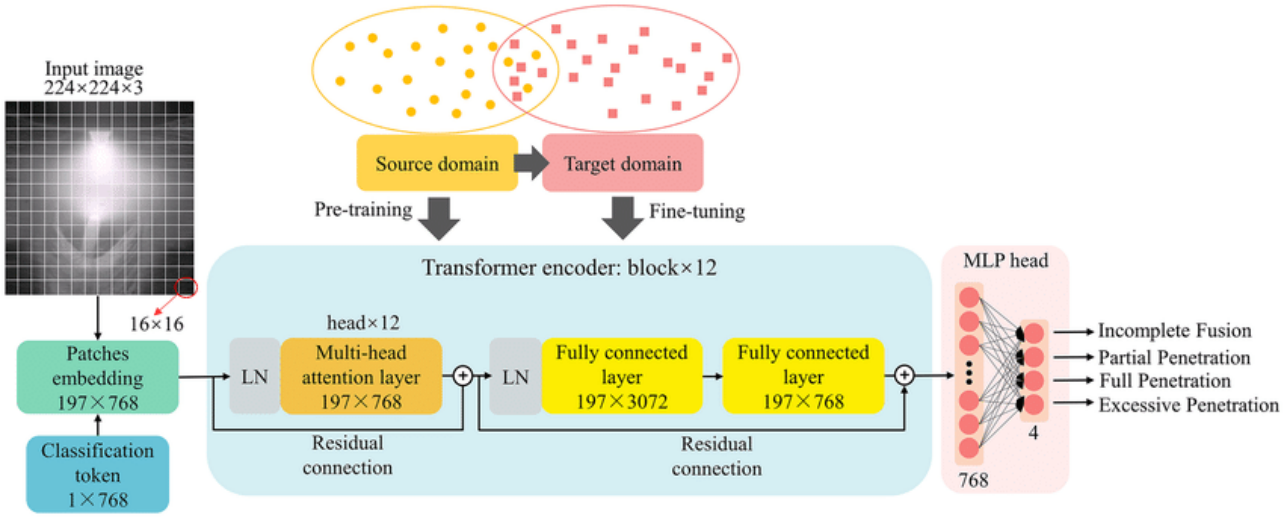

Figure 10: ViT-B/16 network's architecture