DASC 5309 – DATA SCIENCE CAPSTONE PROJECT

MELANOMA DETECTION BASED ON DEEP NEURAL NETWORKS

Abhishek Shrinivas Joshi (1002050821)   Lakshmi Ganapathi Nukala (1002060514)
,Tirupathirao Naga (1002077760)

ABSTRACT:

Melanoma is a kind of skin cancer that can spread quickly and has a variety of symptoms. As such, it can be difficult to identify and classify early on. Deep neural networks (DNNs) have become highly effective instruments for medical image analysis in recent years, providing promising paths toward enhancing the precision and effectiveness of melanoma categorization. This study investigates the use of DNNs in the categorization of melanoma cancer, specifically examining how well they distinguish benign from malignant tumors using dermoscopic pictures. This work intends to improve melanoma detection and therapy by doing a thorough analysis of pertinent literature and experimenting with different DNN architectures.

INTRODUCTION:

Skin cancer, especially melanoma, continues to be a major global health concern that requires prompt and correct detection in order to effectively treat. Medical image analysis, particularly the classification of melanoma, has undergone a revolutionary change with the introduction of deep learning algorithms combined with transfer learning procedures. Deep neural networks (DNNs) have proven to be remarkably adept at extracting complex information from medical images, allowing for highly accurate automated diagnosis. By applying information from related tasks to melanoma classification, transfer learning overcomes the problem of data scarcity in medical imaging by utilizing pre-trained models developed on large-scale datasets.

The traditional method of melanoma identification, which mostly depends on dermatologists' subjective visual assessment, is error-prone by nature. But new developments in medical technology, especially in the field of computer vision and artificial intelligence (AI), have ushered in a revolutionary era for the detection and diagnosis of melanoma. There are promising opportunities to increase the accuracy and productivity of skin cancer diagnosis by utilizing computational techniques like image acquisition, data preprocessing, and feature extraction through sophisticated machine learning models like Convolutional Neural Networks (CNN) and Support Vector Machines (SVM). By combining medical knowledge with state-of-the-art AI techniques this project endeavors to address the critical need for more accurate and reliable tools in the early detection and diagnosis of melanoma.

In addition to enhancing dermatologists' skills, this integration of cutting-edge technologies solves the urgent demand for scalable and easily accessible diagnostic solutions. These AI-driven solutions have the potential to democratize access to high-quality healthcare, especially in underserved regions, by automating and expediting the melanoma detection process. Furthermore, these systems' continual evolution and improvement as a result of continuing research and development initiatives promise to increase their effectiveness and influence even further. As a result, this collaborative strategy is a critical step in reducing the incidence of melanoma worldwide and enhancing public health outcomes generally.

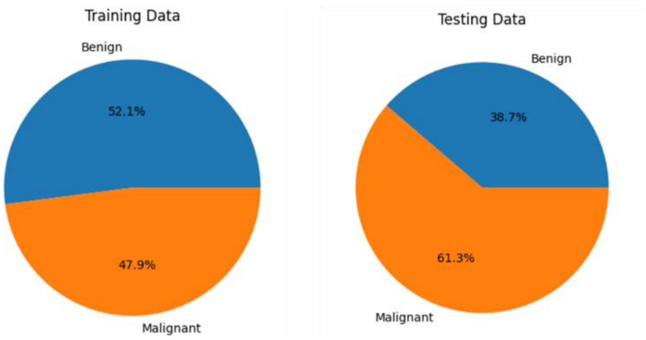
PROBLEM STATEMENT:

The project aims to address the critical need for accurate melanoma detection by developing a deep learning model from scratch and evaluating its performance against established pre-trained models, VGG16 and ResNet50. Leveraging a melanoma dataset, the study will employ evaluation metrics including sensitivity, specificity, precision, accuracy, and F1-score to comprehensively assess the efficacy of each model. By comparing the performance of the custom-built model with these pre-trained architectures, the project seeks to determine the optimal approach for melanoma detection. Ultimately, the findings will contribute to advancing automated melanoma diagnosis, potentially enhancing early detection rates and improving patient outcomes in skin cancer management.

PROPOSED METHODOLOGY:

Our approach involves a custom hybrid CNN model trained on a curated dataset of 40,000 high-resolution dermoscopic images. The methodology involves data loading, data preprocessing, building VGG-16 and RESNET-50 through transferred learning and evaluate them with various metrics mentioned in Problem Statement. After this our own models are build and evaluated as above. The results are then compared.

1. DATA DISTRIBUTION



2. VGG-16 MODEL

Model: "model"

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	[(None, 224, 224, 3)]	0
block1_conv1 (Conv2D)	(None, 224, 224, 64)	1792
block1_conv2 (Conv2D)	(None, 224, 224, 64)	36928
block1_pool (MaxPooling2D)	(None, 112, 112, 64)	0
block2_conv1 (Conv2D)	(None, 112, 112, 128)	73856
block2_conv2 (Conv2D)	(None, 112, 112, 128)	147584
block2_pool (MaxPooling2D)	(None, 56, 56, 128)	0
block3_conv1 (Conv2D)	(None, 56, 56, 256)	295168
block3_conv2 (Conv2D)	(None, 56, 56, 256)	590080
block3_conv3 (Conv2D)	(None, 56, 56, 256)	590080
block3_pool (MaxPooling2D)	(None, 28, 28, 256)	0
block4_conv1 (Conv2D)	(None, 28, 28, 512)	1180160
block4_conv2 (Conv2D)	(None, 28, 28, 512)	2359808
block4_conv3 (Conv2D)	(None, 28, 28, 512)	2359808
block4_pool (MaxPooling2D)	(None, 14, 14, 512)	0
block5_conv1 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv2 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv3 (Conv2D)	(None, 14, 14, 512)	2359808
block5_pool (MaxPooling2D)	(None, 7, 7, 512)	0
global_average_pooling2d (GlobalAveragePooling2D)	(None, 512)	0
flatten (Flatten)	(None, 512)	0
dense (Dense)	(None, 1)	513

Total params: 14715201 (56.13 MB)
Trainable params: 14715201 (56.13 MB)
Non-trainable params: 0 (0.00 Byte)

The training dataset is made more robust by using techniques like data augmentation, normalization, and noise reduction using Gaussian Blur. This ensures that the dimensions are standardized, there is more diversity, and there is less noise, which improves generalization and lessens the risk of overfitting in the deep learning model.

The deep stack of small 3x3 filters, followed by max-pooling layers, characterizes the architecture of the VGG16 model, which consists of 16 convolutional and fully connected layers and achieves high accuracy in image classification tasks. Along with this a dense layer was added for binary classification. Further information is provided in appendix.

The ResNet-50 model architecture mitigates the vanishing gradient problem and allows for the training of very deep neural networks. It consists of 50 layers, including residual blocks with shortcut connections. Along with this a dense layer was added for binary classification. Further information is provided in appendix. Further information is provided in appendix.

The Shallow CNN Model is the sequential model, which has a total of about 25.7 million parameters and is mainly used for image classification tasks, consists of two convolutional layers, max-pooling and dropout layers, a flattening layer, and a dense layer. Further information is provided in appendix.

```
Model: "sequential_4"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_6 (Conv2D)           (None, 224, 224, 64)      1792

 max_pooling2d_6 (MaxPoolin  (None, 112, 112, 64)      0
 g2D)

 dropout_8 (Dropout)         (None, 112, 112, 64)      0

 conv2d_7 (Conv2D)           (None, 112, 112, 64)      36928

 max_pooling2d_7 (MaxPoolin  (None, 56, 56, 64)        0
 g2D)

 dropout_9 (Dropout)         (None, 56, 56, 64)        0

 flatten_3 (Flatten)         (None, 200704)            0

 dense_6 (Dense)             (None, 128)               25690240

 dense_1 (Dense)             (None, 1)                 129

=================================================================
Total params: 25729089 (98.15 MB)
Trainable params: 25729089 (98.15 MB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

3. SHALLOW COMPLEX MODEL

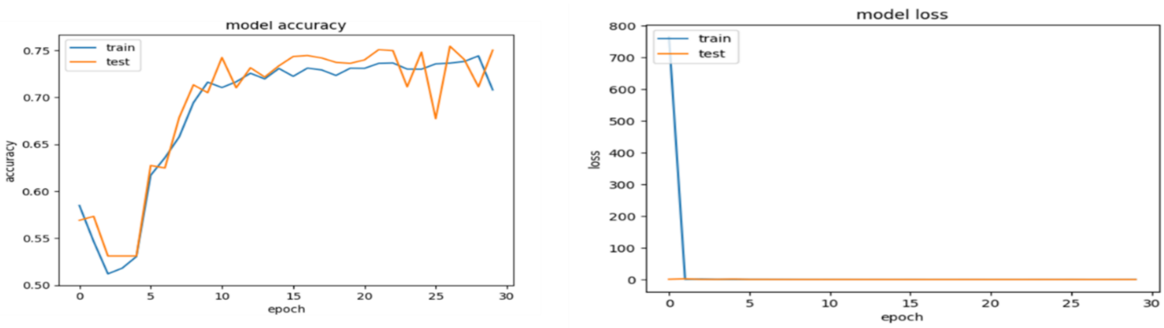
The Complex CNN Model architecture consists of 4 Conv2D layers, 4 BatchNormalization layers, 4 MaxPooling2D layers, 7 Dropout layers, 4 Flatten layers and 5 Dense layers, achieving a total of approximately 104.9 million parameters, designed for image classification tasks. Further information is provided in appendix.

ANALYSIS AND RESULTS:

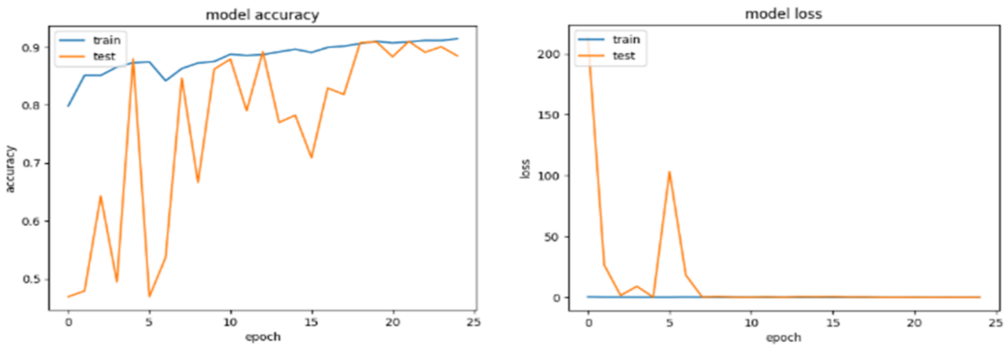
MODELS	ACCURACY		PRECISION		RECALL		F1-SCORE		AUC
	TRAINING	TESTING	BENIGN	MALIGNANT	BENIGN	MALIGNANT	BENIGN	MALIGNANT	
Visual Geometry Group Model (VGG16)	80.27	59.39	65.39	61.1	5.71	93.2	10.5	73.79	49.46
Resnet 50 Model	90.92	88.47	18.52	93.75	90.79	87	30.77	90.25	88.9
Shallow Hybrid Model	73.09	74.6	37.58	89.25	87.46	66.4	52.57	76.15	76.93
Complex Hybrid Model	95.55	87.18	22.28	95.61	94.04	82.8	36.03	88.75	88.42

4. MODEL COMPARISON TABLE (IN %)

1. The lower performance on the testing set indicates that the VGG16 model struggles with generalization, despite exhibiting respectable accuracy on the training set. The low F1-score and AUC indicate that it performs poorly in identifying malignant cases, but it recalls benign cases relatively well.



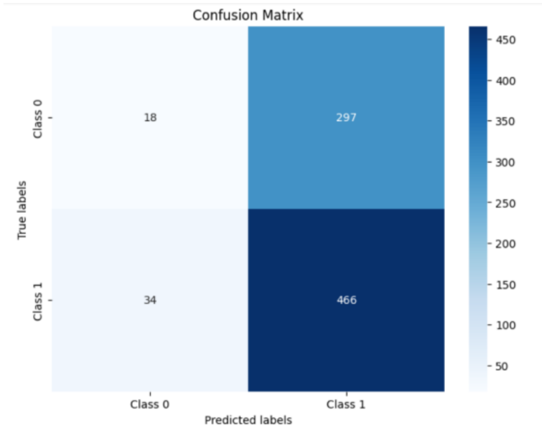
5. VGG 16 MODEL FITTING



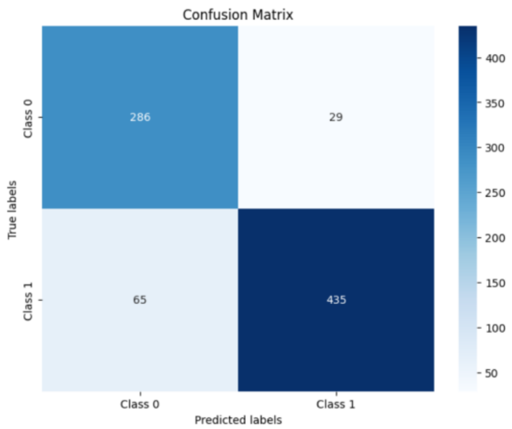
6. Resnet 50 MODEL FITTING

2. The ResNet50 model performs admirably on all metrics, particularly when it comes to correctly identifying cases that are cancerous. It does, however, have accuracy issues in benign cases, suggesting a propensity for false positives.

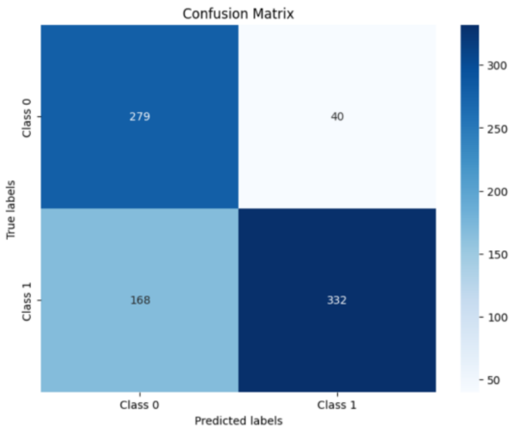
VGG-16



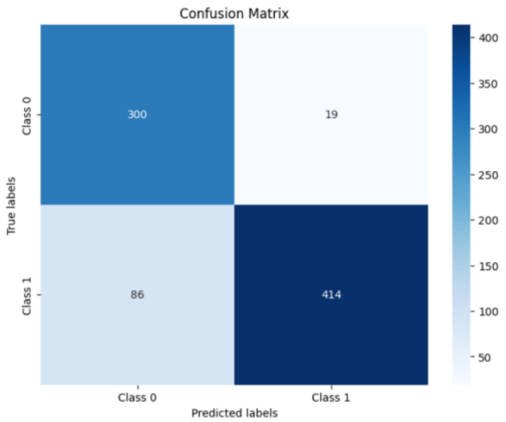
RESNET-50



SHALLOW HYBRID

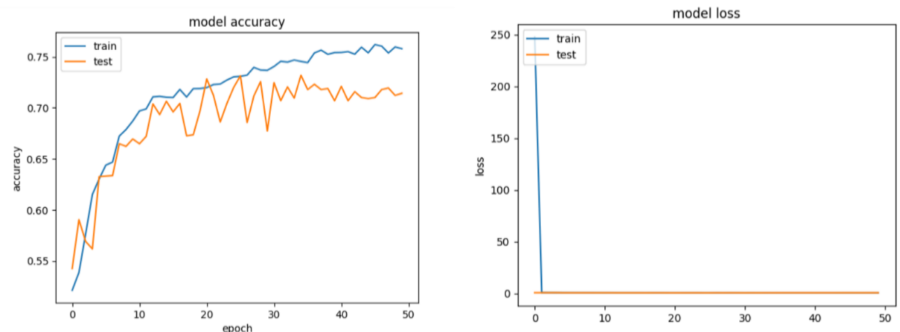


COMPLEX HYBRID



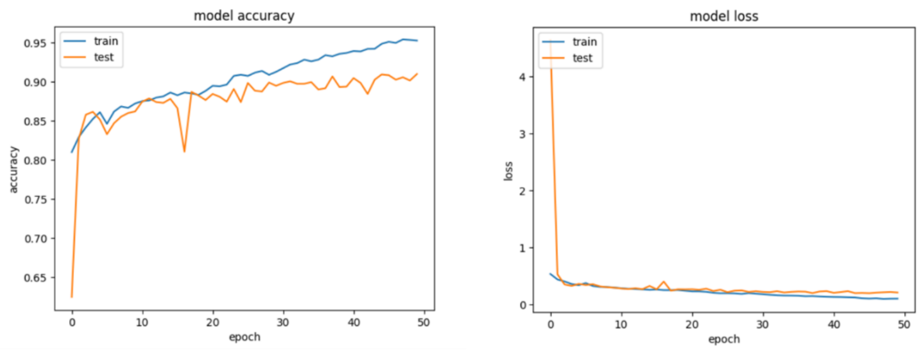
7. CONFUSION MATRICES

3. While it performs in a balanced manner, the Shallow Hybrid Model is not as accurate or resilient as deeper models. For malignant cases, it shows good precision, but for benign cases, recall and precision are comparatively low.



8. SHALLOW HYBRID MODEL FITTING

4. On the training set, the Complex Hybrid Model performs well, but on the testing set, its performance slightly declines, suggesting that overfitting may have occurred. When it comes to malignant cases, it shows outstanding precision and recall, but it has trouble with precision when it comes to benign cases.



9. COMPLEX HYBRID MODEL FITTING

CONCLUSION:

1. Both of our custom models showcase promising performance, particularly when compared to the pre-trained VGG16 model.
2. The Complex Hybrid Model exhibits high precision, indicating its effectiveness in correctly identifying positive cases, comparable to the performance of ResNet-50 in terms of precision.
3. While the Shallow Hybrid Model falls short in accuracy compared to ResNet-50, it demonstrates robustness in precision and F1 score, suggesting its potential for specific applications where precision is crucial.
4. Our models show potential for further fine-tuning to match or even surpass the accuracy of ResNet-50 while maintaining or improving upon the precision and robustness of both ResNet-50 and VGG16. When it comes to overall accuracy and its capacity to correctly identify cases of malignancy, the ResNet50 model performs better than other models.

LESSONS LEARNED:

1. Transitioning to Google Colab with T4 GPU significantly eased the challenge of handling large datasets on localhost Jupyter notebooks, as discussed during team meetings.
2. The initial presentation enlightened us about the paramount importance of precision and recall in medical disease classification, surpassing mere accuracy metrics, offering a profound insight into the practical utility of these additional evaluation measures.
3. Effective time management emerged as a crucial factor in the successful completion of the project, especially highlighted by the fact that until April 1st, we were still engrossed in working with pre-trained models, leading to last-minute stress in constructing and executing our custom models from scratch.

4. Our engagement with the course modules furnished us with abundant knowledge, which we diligently endeavored to integrate into our project implementation, leveraging the insights gleaned from the courses to enhance our project outcomes.

BIBLIOGRAPHY:

- Singha, S.; Roy, P. Skin Cancer Classification and Comparison of Pretrained Models Performance using Transfer Learning. Preprints 2022, 2022090215. https://doi.org/10.20944/preprints202209.0215.v1

- https://www.kaggle.com/datasets/hasnainjaved/melanoma-skin-cancer-dataset-of-10000-images/data

- He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).

- Simonyan, K., & Zisserman, A. (2014). Very Deep Convolutional Networks for Large-Scale Image Recognition. arXiv preprint arXiv:1409.1556.

- Esteva, A., Kuprel, B., Novoa, R. A., Ko, J., Swetter, S. M., Blau, H. M., & Thrun, S. (2017). Dermatologist-level classification of skin cancer with deep neural networks. Nature, 542(7639), 115-118.

- Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., ... & Rabinovich, A. (2015). Going deeper with convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1-9).

- Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. In International Conference on Medical image computing and computer-assisted intervention (pp. 234-241). Springer, Cham.

- Litjens, G., Kooi, T., Bejnordi, B. E., Setio, A. A. A., Ciompi, F., Ghafoorian, M., ... & Sánchez, C. I. (2017). A survey on deep learning in medical image analysis. Medical image analysis, 42, 60-88.

- https://machinelearningmastery.com/precision-recall-and-f-measure-for-imbalanced-classification/

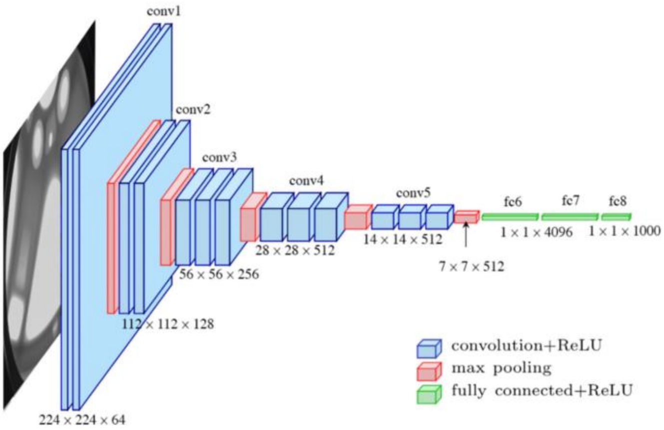
- Chollet, F. (2017). Xception: Deep learning with depthwise separable convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1251-1258).

- OFFICIAL DOCUMENTATION OF KERAS, TENSORFLOW, PANDAS AND NUMPY

APPENDIX:

A. VGG-16 MODEL ARCHITECTURE –

The VGG-16 (Visual Geometry Group 16) model architecture is a convolutional neural network (CNN) designed for image classification tasks. It was developed by the Visual Geometry Group at the University of Oxford. Here's a detailed breakdown of the VGG-16 model architecture:



10. VGG-16 MODEL

1. Input Layer:

- The input layer receives images with dimensions 224x224 pixels, typically in the RGB (red, green, blue) color space.

2. Convolutional Blocks:

- The model consists of 13 convolutional layers, organized into five blocks, each followed by a max-pooling layer.
- Each convolutional layer uses a small 3x3 filter with a stride of 1 and 'same' padding to preserve the spatial dimensions of the input.
- The number of filters gradually increases from 64 to 512 as we move deeper into the network.

3. Max Pooling Layers:

- After each block of convolutional layers, a max-pooling layer with a 2x2 window and stride of 2 is applied to reduce the spatial dimensions of the feature maps while retaining important information.

4. Fully Connected Layers:

Following the convolutional layers, there are three fully connected (dense) layers.

- The first two dense layers have 4096 neurons each, followed by a final dense layer with 1000 neurons, corresponding to the number of output classes in the original VGG-16 model.
- Each dense layer is activated by the rectified linear unit (ReLU) activation function, except for the output layer, which typically uses softmax for multi-class classification tasks.

5. Flattening Layer:

Before the first dense layer, a flattening layer reshapes the output of the last convolutional layer into a 1-dimensional vector, enabling it to be fed into the dense layers.

6. Dropout Layers:

- Dropout layers are added after each fully connected layer to prevent overfitting by randomly dropping a fraction of the neurons during training.

7. Output Layer:

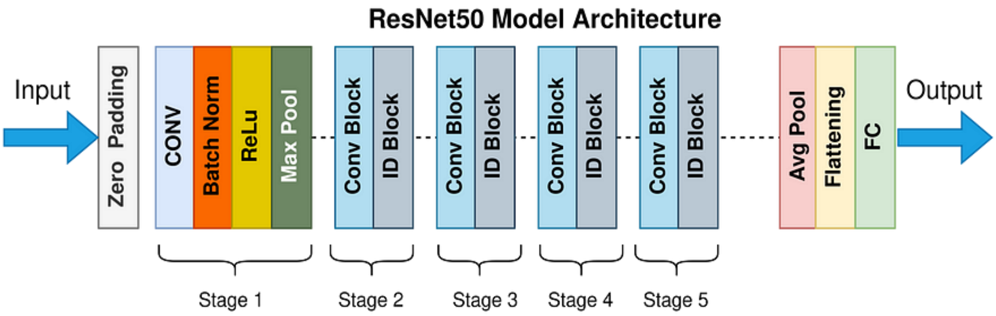
- The output layer produces the final predictions, with the number of neurons corresponding to the number of classes in the classification task.
- For tasks with binary classification, such as melanoma detection, the output layer typically has a single neuron with a sigmoid activation function.

8. Parameters:

The VGG-16 model has approximately 138 million parameters, making it a relatively large and computationally expensive model compared to earlier architectures like AlexNet.

Overall, the VGG-16 architecture's simplicity, with stacked small convolutional filters, enables effective feature extraction and learning, contributing to its success in various image classification tasks.

B. RESNET-50 MODEL ARCHITECTURE –



11. RESNET-50 MODEL

Microsoft Research created the deep convolutional neural network architecture known as ResNet-50 (Residual Network with 50 layers). Through the introduction of skip connections, also referred to as residual connections, it solves the issue of vanishing gradients in very deep networks. An extensive synopsis of the ResNet-50 architecture is provided below:

1. Input Layer: Typically operating in the RGB (red, green, blue) color space, the input layer can handle images up to 224 by 224 pixels in size.
2. Convolutional Layers:
   - A 7x7 filter with a stride of 2 is used in the first convolutional layer.
   - This is followed by max-pooling with a 3x3 window and a stride of 2.
   - ResNet-50 consists of four stages after the first layer, each of which has a number of convolutional blocks.
3. Convolutional Blocks:
   - There are several convolutional blocks in each stage, and each block has several convolutional layers.

- The use of residual or skip connections, which allow the network to learn residual functions with respect to the input, is the main innovation in ResNet.
- By doing this, the degradation issue brought on by vanishing gradients is mitigated.
- The bottleneck block, which consists of three convolutional layers with 1x1, 3x3, and 1x1 filters, is the fundamental building block. These building blocks support lowering the computational complexity while maintaining the expressive capability of the network.

4. Identity Blocks: ResNet-50 contains identity blocks in addition to bottleneck blocks. These blocks have fewer layers and are utilized when a block's input and output dimensions match.
5. Global Average Pooling Layer: To extract the most significant features and reduce the spatial dimensions of the feature maps, a global average pooling layer is applied after the convolutional stages.
6. Fully Connected Layer: After the global average pooling layer, there is a dense (fully connected) layer that usually has 1000 neurons, which is the same number of output classes as in the original ResNet-50 model.
7. Repaired linear units (ReLU) are utilized as activation functions across the network, with the exception of the final output layer, which usually uses softmax for multi-class classification tasks.

Overall, the residual connection architecture of ResNet-50 reduces the vanishing gradient issue and allows for the training of very deep networks, which improves performance on image classification tasks.

C. SHALLOW HYBRID MODEL ARCHITECTURE –

The provided model architecture consists of two parts: a feature extraction backbone (sequential_2) and a classification head (sequential_3). Let's break down each part in detail:

Feature Extraction Backbone (sequential_2):

1. Convolutional Layers:
   - The model begins with a convolutional layer (conv2d_2) with 64 filters, each having a size of 3x3. This layer extracts features from the input images.
   - Batch normalization (batch_normalization) is applied after each convolutional layer to stabilize and accelerate the training process.
2. Max Pooling Layers:
   - After each convolutional block, max-pooling (max_pooling2d) is performed with a 2x2 window size and stride of 2, reducing the spatial dimensions of the feature maps.
3. Dropout Layers:
   - Dropout layers (dropout) are incorporated after each max-pooling layer to prevent overfitting by randomly dropping a fraction of the neurons during training.
4. Flattening Layer:
   - The feature maps are flattened into a 1-dimensional vector using a flattening layer (flatten_1), preparing them to be fed into the subsequent dense layers.

5. Dense Layers:
    - Two dense layers (dense_2 and dense_3) follow the flattening layer. These layers have 1024 and 512 neurons, respectively, with ReLU activation functions, enabling the model to learn complex patterns in the feature space.
6. Batch Normalization and Dropout:
    - Batch normalization and dropout layers are applied after each dense layer to further enhance the model's stability and prevent overfitting.

Classification Head (sequential_3):

1. Feature Extraction Backbone Integration:
    - The feature extraction backbone (sequential_2) is included as the first layer in the classification head (sequential_3), serving as a feature extractor.
2. Dense Layer for Classification:
    - Following the feature extraction backbone, a dense layer (dense_5) with a single neuron is added for binary classification. It outputs the probability of the input belonging to the positive class (e.g., malignant for cancer detection).

Summary:

- Total Parameters: The entire model has approximately 104.9 million trainable parameters, making it a large and complex neural network.
- Non-trainable Parameters: Additionally, there are 4992 non-trainable parameters, likely associated with batch normalization layers.
- Trainable Parameters: Most of the parameters (104.9 million) are trainable, enabling the model to learn representations directly from the data during training.

D. COMPLEX HYBRID MODEL ARCHITECTURE -

The provided model architecture consists of two sequential models, where the first model serves as a feature extractor (sequential_2), and the second model (sequential_3) acts as a classifier. Let's delve into each part in detail:

Feature Extraction Backbone (sequential_2):

- Convolutional Layers:
    - The model begins with a convolutional layer (conv2d_2) with 64 filters and a 3x3 kernel size, producing feature maps of size 224x224x64.
    - Batch normalization (batch_normalization) is applied to stabilize and accelerate the training process by normalizing the activations.
- Max Pooling Layers:
    - Max-pooling (max_pooling2d_2) is performed with a 2x2 window size and stride of 2, reducing the spatial dimensions by half, resulting in feature maps of size 112x112x64.

- Dropout Layers:
    - Dropout layers (dropout_2, dropout_3, dropout_4, dropout_5, dropout_6, dropout_7) are applied after each max-pooling layer to prevent overfitting by randomly dropping a fraction of the neurons during training.
- Flattening Layer:
    - The feature maps are flattened into a 1-dimensional vector using a flattening layer (flatten_1), preparing them to be fed into the subsequent dense layers.
- Dense Layers:
    - Two dense layers (dense_2 and dense_3) follow the flattening layer. These layers have 1024 and 512 neurons, respectively, with ReLU activation functions, enabling the model to learn complex patterns in the feature space.
- Batch Normalization Layers:
    - Batch normalization (batch_normalization_4 and batch_normalization_5) is applied after each dense layer to further stabilize and accelerate training.

Classification Head (sequential_3):

- Feature Extraction Backbone Integration:
    - The feature extraction backbone (sequential_2) is included as the first layer in the classification head (sequential_3), serving as a feature extractor.
- Dense Layer for Classification:
    - Following the feature extraction backbone, a dense layer (dense_5) with a single neuron is added for binary classification. It outputs the probability of the input belonging to the positive class (e.g., malignant for cancer detection).

Summary:

- Total Parameters: The entire model has approximately 104.9 million trainable parameters, making it a large and complex neural network.
- Non-trainable Parameters: Additionally, there are 4992 non-trainable parameters, likely associated with batch normalization layers.
- Trainable Parameters: Most of the parameters (104.9 million) are trainable, enabling the model to learn representations directly from the data during training.