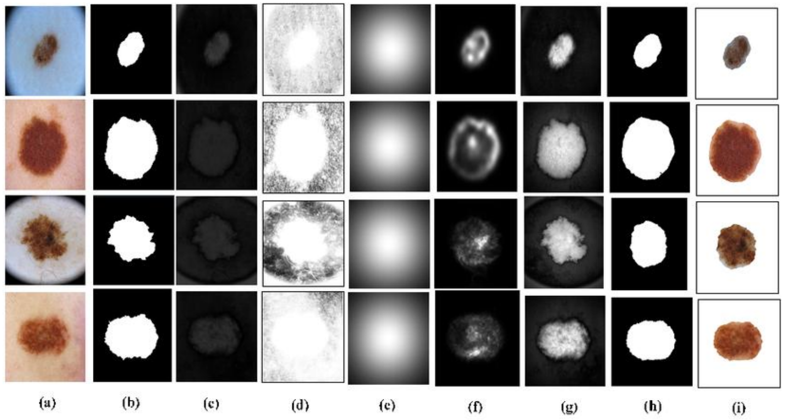
A comprehensive analysis of different transfer learning techniques for skin cancer classification

Presented by:
MD Rayhan Hossain[1], MD Rahat Al Mamun[2], Saif Nasim Bosunia[1], Jotirmoy Mollick[1], Muaaz Mohammad[1]
[1]Department of Computer Science & Engineering
[2]Department of Electrical & Electronic Engineering
INDEPENDENT UNIVERSITY, BANGLADESH

Presented To: **Md. Rashedur Rahman, D.Eng**

Background and Objective

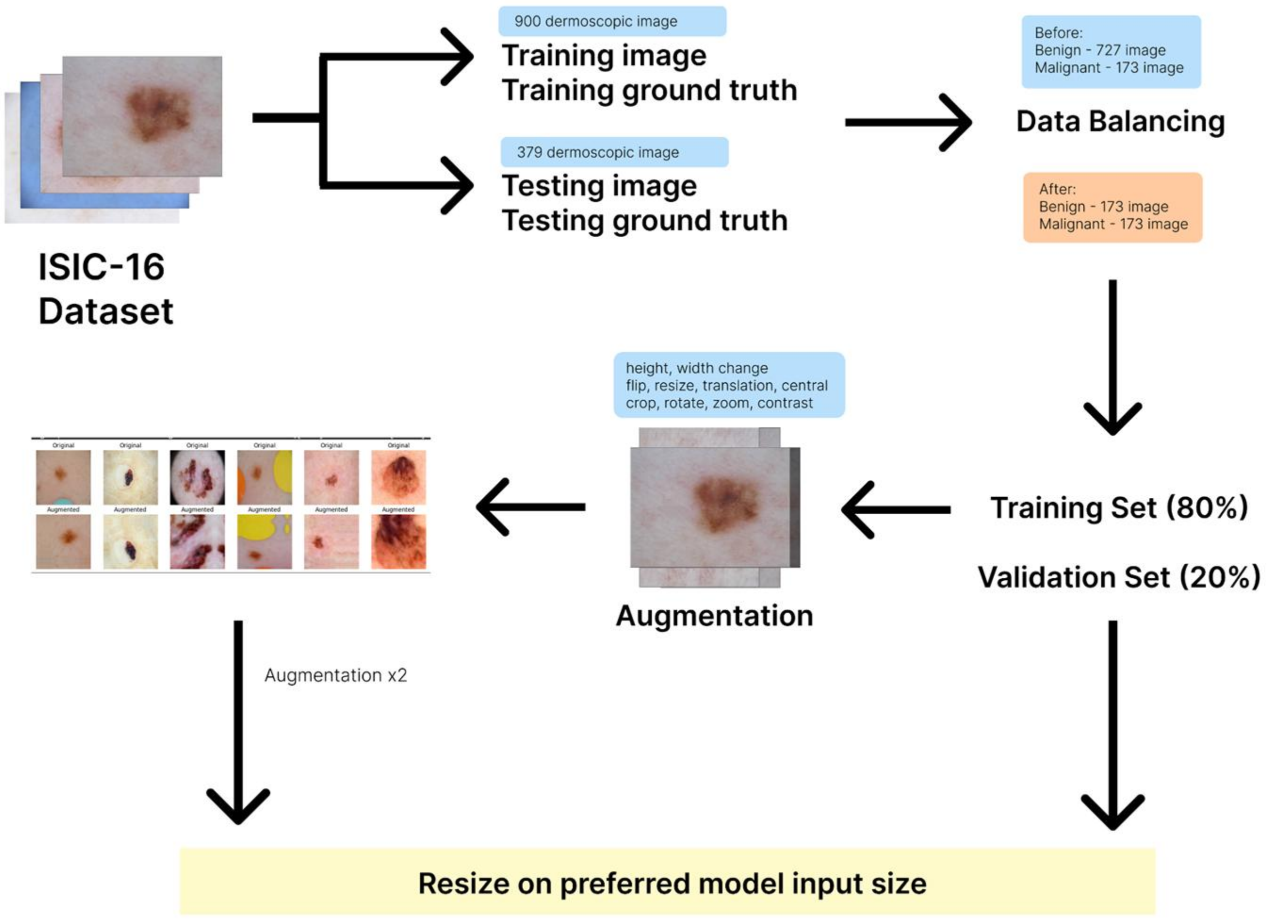
ISIC 2016 Dataset



International Skin Imaging Collaboration (ISIC) widely used benchmarking dataset used for research in automated skin lesion analysis.
It contains 900 dermoscopic image. Each image labeled for binary classification into one of two categories: benign or malignant.

Background:	Objective:
- Identifying skin cancer using traditional method is time consuming and financially burdening. - Computer based diagnosis produce faster and more accurate results	- To apply **transfer learning** with multiple CNN architectures, including MobileNetV2, InceptionV3, EfficientNetB0, ResNet50, and VGG16. - To determine the **most effective model architecture and training strategy** for reliable automated skin cancer detection.

Data Preprocessing



Dataset	Train: 900 Dermoscopic image and groundtruth Test: 379 image with groundtruth
Class Imbalance	Benign: 727 images Malignant: 173 images
Data Balancing	Benign: 173 images Malignant: 173 images
Augmentation (*on training set*)	Geometric transformation, color based augmentation, gaussian noise.

Later, images were resized into preferred model input size

MobileNetV2

BASE MODEL

Features	Description
Input Size	224×224×3 (RGB images)
Total Parameters	Approximately 3.49 million parameters
Trainable Parameters (before freezing)	Approximately 3.49 million parameters (since the base model is frozen)
Model Size (disk)	Approximately 13.3 MB (floating-point precision)
Top-1 accuracy on imagenet	Approximately 71.8%

What is MobileNetV2

- **Lightweight** CNN by Google for mobile and edge devices.
- **Efficient:** significantly less computation than traditional CNNs (~350 GFLOPs).
- **Input**: 224×224×3 images (ImageNet-pretrained).
- **Architecture:** 53 convolution layers + 1 average pooling layer.

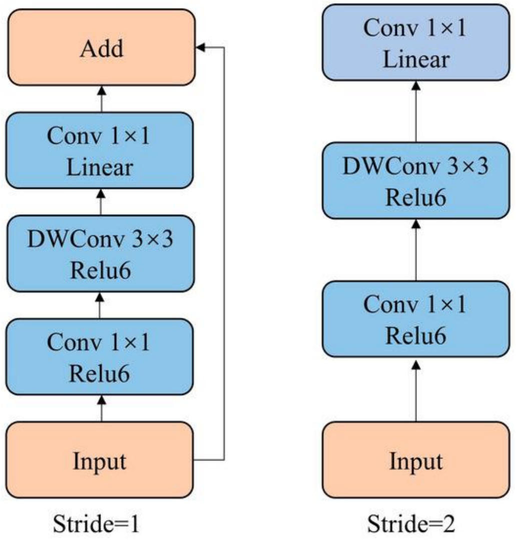


Figure title: Visualization of MobileNetV2

Core Components

Residual Blocks

Inverted Residual Bottleneck Blocks

MobileNetV2 (continued)

Transfer Learning on Raw Dataset

- Used MobileNetV2 (ImageNet pretrained, no top layer).
- kept the base of mobileNetV2 frozen, added custom layers (GAP, Dropout, Dense-256, Sigmoid).
- Compiled with Adam (1e-5), binary crossentropy, metrics (Acc, Precision, Recall).
- Handled class imbalance with weights + EarlyStopping.
- Trained on raw dataset (no augmentation).
- Evaluated Acc, Precision, Recall, training curves + confusion matrix.

Transfer Learning on Augmented Dataset

- Applied On Augmented Dataset.
- Used same MobileNetV2 (frozen base + custom layers).
- Compiled with Adam (1e-4), metrics (Acc, Precision, Recall, AUC).
- Added callbacks: ModelCheckpoint, EarlyStopping, ReduceLROnPlateau.
- Trained on augmented dataset (50 epochs).
- Evaluated Loss, Acc, Precision, Recall, AUC, training/validation curves + confusion matrix.

Fine-Tuning on Augmented Model

- Unfroze deeper layers (from block_13), kept BatchNorm frozen.
- Recompiled with Adam (1e-6) for careful fine-tuning.
- Continued training on augmented data (25 epochs).
- Same callbacks (Checkpoint, EarlyStopping, ReduceLROnPlateau).
- Evaluated final Loss, Acc, Precision, Recall, AUC.
- Plotted full training curves + final confusion matrix.

MobileNetV2 (continued)

Performance of MobileNetV2 Across Training Strategies

1. When I trained on the raw dataset, the model reached 54% accuracy with a recall of 72%, meaning it caught most positives. Precision was only 26%, showing it produced too many false alarms and its positive predictions were unreliable.

2. After live augmentation, performance improved to 65% accuracy with an AUC of 0.74. Recall stayed decent at 69%, and precision rose slightly to 32.5%, but the model still struggled with many false positives, making its positive predictions weak.

3. Finally, after fine-tuning on the augmented model, the model gave the best results: 67.5% accuracy, AUC 0.75, and precision improved to 34%. While accuracy and reliability improved slightly, the false positive issue still limits its effectiveness for medical diagnosis.

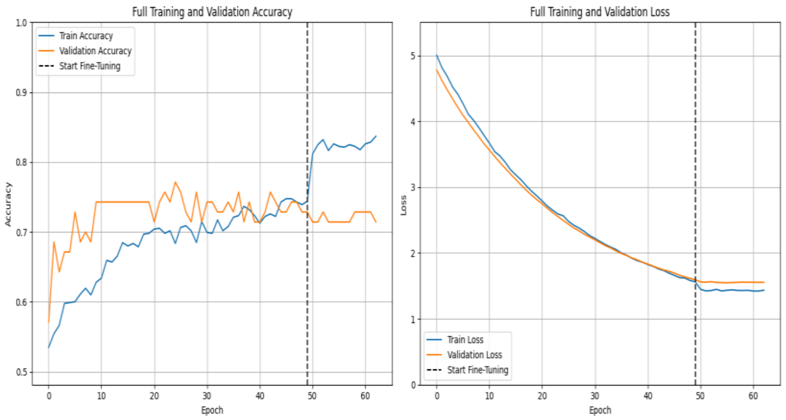


Figure title: Training with fine tuning on augmented transfer learning model

Figure title: Training with on raw data

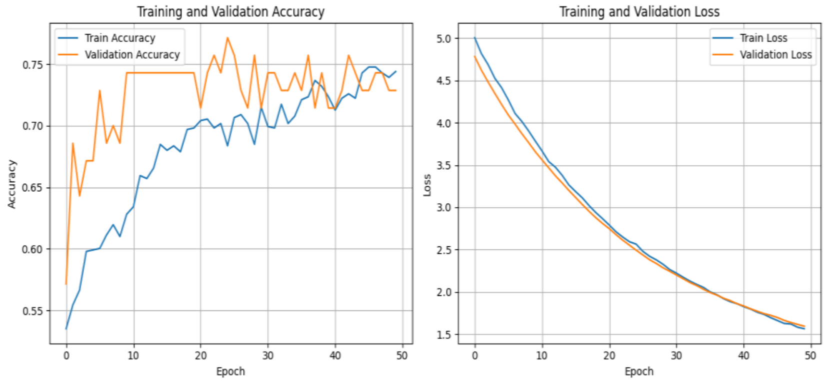


Figure title : Training with on augmented data

MobileNetV2 (continued)

Why Good for Transfer Learning:

- MobileNetV2 is pre-trained on ImageNet, so it already knows how to recognize a wide variety of features from images.
- Its base layers can be kept frozen while adding new layers specific to your task, which allow the model to quickly adapt to new problems.
- This approach works well even with small datasets and helps reduce overall training time.

Benefits for Medical Sector:

- MobileNetV2 delivers high accuracy while keeping computational costs low.
- It has been successfully used in medical imaging tasks, such as skin lesion classification.
- The model can help with early disease detection, supporting faster and more efficient diagnoses.

Inception V3

BASE MODEL

Features	Description
Input Size	299 x 299 x 3
Total Parameters	~ 23 million
Trainable Parameters (before freezing)	~0.5 million (can be reduced if layers are frozen)
Model Size (disk)	~90.9 MB (TensorFlow/Keras implementation)
Top-1 accuracy on imagenet	~77% (baseline performance)

FINE TUNE

Features	Description
Frozen layers	mixed8
Unfrozen layers	mixed9 & mixed10
Trainable Parameters	~6.5 million

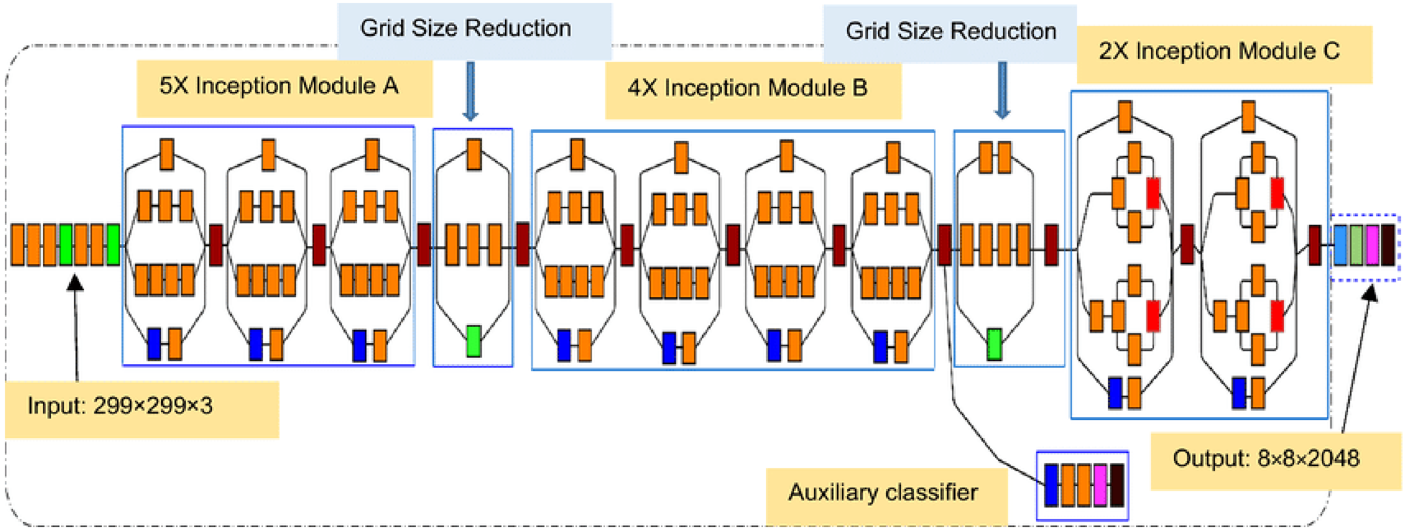


Fig Title: Inception V3 Baseline Architecture

Freeze up to mixed8

- Preserves general features (edges, textures, shapes)
- Faster training, fewer trainable parameters
- Less flexibility if your dataset differs a lot from ImageNet

Unfreeze mixed9 & mixed10

- Lets high-level abstract features adapt to your dataset
- Usually improves classification accuracy
- Risk of overfitting on small datasets, training a bit slower

Keep BatchNorm frozen

- Prevents instability and exploding gradients
- BN stats may not perfectly match your new dataset

Inception V3 (continued)

Transfer Learning on Raw Dataset

- **Base Model:** InceptionV3 (ImageNet pretrained, no top layer, input 299×299×3)
- **Base Frozen:** Yes
- **Custom Layers:** GAP → Dropout(0.2) → Dense-256 (ReLU) → Dropout(0.3) → Dense-1 (Sigmoid)
- **Optimizer & Loss:** Adam (1e-4), Binary Crossentropy
- **Metrics:** Accuracy, Precision, Recall, AUC
- **Imbalance Handling:** Class Weights + EarlyStopping
- **Training Data:** Raw dataset (no augmentation)
- **Evaluation:** Test Loss, Accuracy, Precision, Recall, AUC, Training Curves, Confusion Matrix

Transfer Learning on Augmented Dataset

- **Augmentation:** Rotations, Flips, Shifts
- **Base Model:** InceptionV3 (frozen base + custom layers)
- **Custom Layers:** GAP → Dropout(0.3) → Dense-256 (ReLU, L2=0.01) → Dropout(0.5) → Dense-1 (Sigmoid)
- **Optimizer & Loss:** Adam (1e-4), Binary Crossentropy
- **Metrics:** Accuracy, Precision, Recall, AUC
- **Callbacks:** ModelCheckpoint, EarlyStopping, ReduceLROnPlateau
- **Training:** Augmented dataset, up to 100 epochs
- **Evaluation:** Test Loss, Accuracy, Precision, Recall, AUC, Training/Validation Curves, Confusion Matrix

Fine-Tuning on Augmented Model

- **Unfreezing:** Deeper layers from mixed9/mixed10 (BatchNorm kept frozen)
- **Recompiled Optimizer:** Adam (5e-6), Binary Crossentropy
- **Metrics:** Accuracy, Precision, Recall, AUC
- **Callbacks:** ModelCheckpoint, EarlyStopping, ReduceLROnPlateau
- **Training:** Augmented dataset, up to 50 epochs (fine-tuning phase)
- **Evaluation:** Final Test Loss, Accuracy, Precision, Recall, AUC
- **Outputs:** Training/Validation Curves + Final Confusion Matrix

Inception V3 (continued)

Why Good for Transfer Learning:

- MobileNetV2 is pre-trained on ImageNet, so it already knows how to recognize a wide variety of features from images.
- Its base layers can be kept frozen while adding new layers specific to your task, allowing the model to quickly adapt to new problems.
- This approach works well even with small datasets and helps reduce overall training time.

Benefits for Medical Sector:

- MobileNetV2 delivers high accuracy while keeping computational costs low, making it ideal for real-time analysis.
- It has been successfully used in medical imaging tasks, such as skin lesion classification.
- The model can help with early disease detection, supporting faster and more efficient diagnoses.

ResNet18

Features	Description
Model & Input Size	ResNet-18 with 224 × 224 × 3 input
Trainable Layers	Early layers frozen (`layer1`, `layer2`, `layer3`) depending on configuration
FC Head	1–3 hidden layers (512, 256, 128) with BatchNorm, ReLU, Dropout (0.2–0.5)
Training	30 epochs; learning rates 1e-5 (frozen layers), 1e-4 (FC head)
Performance Tracking	Train/val loss & accuracy, learning curves, confusion matrices

- **Model**: ResNet-18 (11.7M params, 512-dim output)
- **Task**: 2-class classification on custom dataset
- **Layer Freezing**: Up to `layer1`, `layer2`, or `layer3`
- **FC Head**: 1–3 hidden layers (512 → 256 → 128) with BN, ReLU, Dropout (0.2–0.5)
- **Training**: 30 epochs; LR = 1e-5 (frozen layers), 1e-4 (FC head)
- **Monitoring**: Training/validation loss & accuracy per epoch; learning curves
- **Evaluation**: Confusion matrices on validation & test sets
- **Focus**: Impact of layer freezing and FC head design on transfer learning

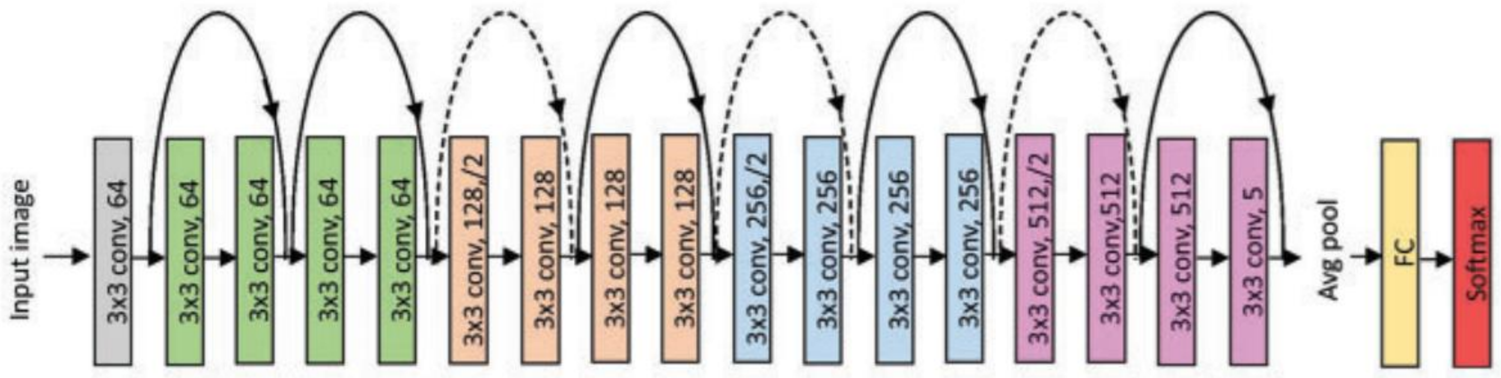


Fig title: ResNet18 baseline architecture

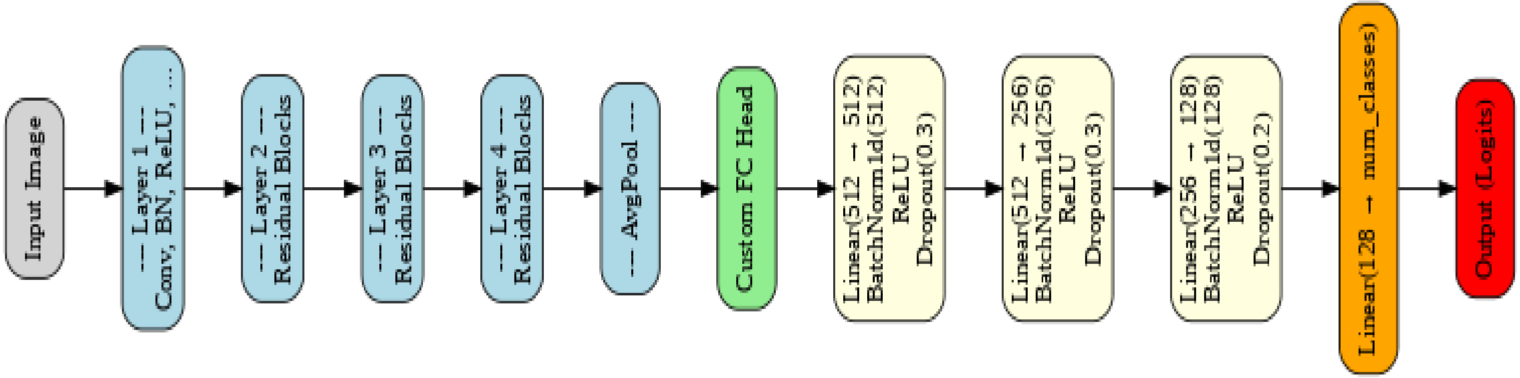


Fig title: ResNet18 Custom Architecture with logits

ResNet18 with sigmoid Activation

- **Model**: Fine-tuned ResNet-18 for binary classification
- **Training**: 12 epochs per configuration, loss = **BCEWithLogitsLoss**
- **Strategy**: Freeze different parts of ResNet-18 backbone to leverage ImageNet features
- **Configurations**:
  1. Freeze up to **layer2** → train layer3, layer4, FC head
  2. Freeze up to **layer1** → train layers 2–4, FC head
  3. Freeze up to **layer3** → train layer4, FC head
- **Evaluation**: Tested thresholds **0.3, 0.5, 0.7** on validation set
- **Final Metrics**: Chosen threshold applied to validation & test accuracies
- **Focus**: Effect of **layer freezing** + **threshold tuning** on transfer learning performance

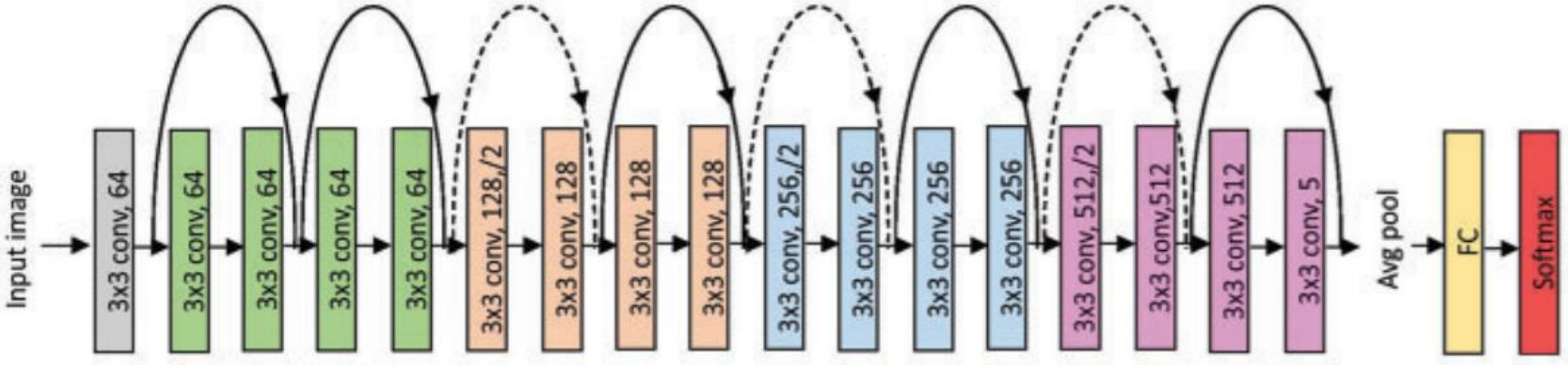


Fig title: ResNet18 baseline architecture

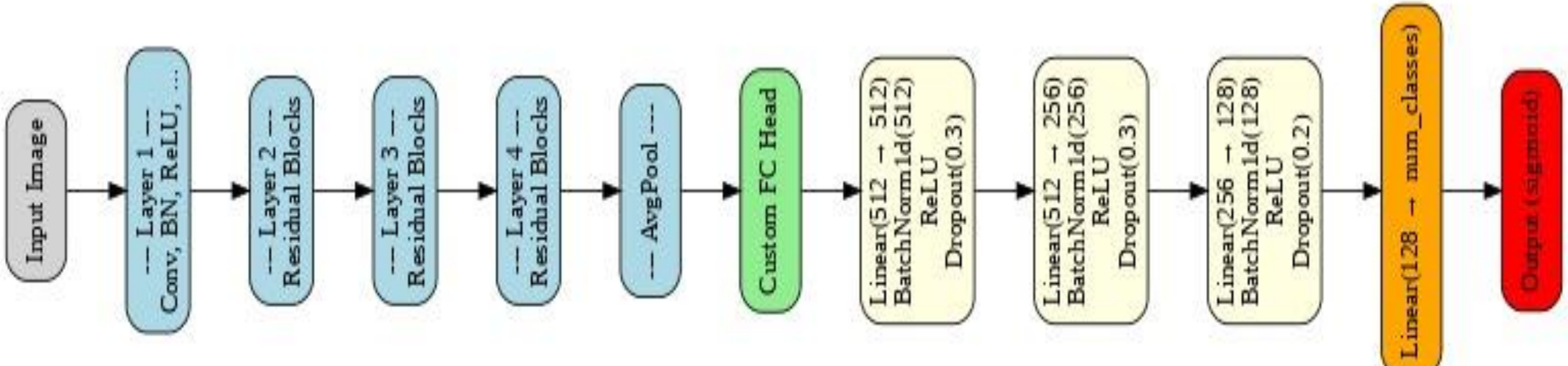


Fig title: ResNet18 Custom Architecture with sigmoid

VGG Architecture

MODEL SUMMARY

Features	Description
Input Size	299 × 299 × 3 RGB (adapted from 224×224)
Total Parameters	~138 million
Trainable Parameters (before freezing)	~120 million (after freezing first 15 layers ~18M frozen)
Model Size (disk)	~528 MB (full VGG16 weights)
Top-1 accuracy on imagenet	71.5%

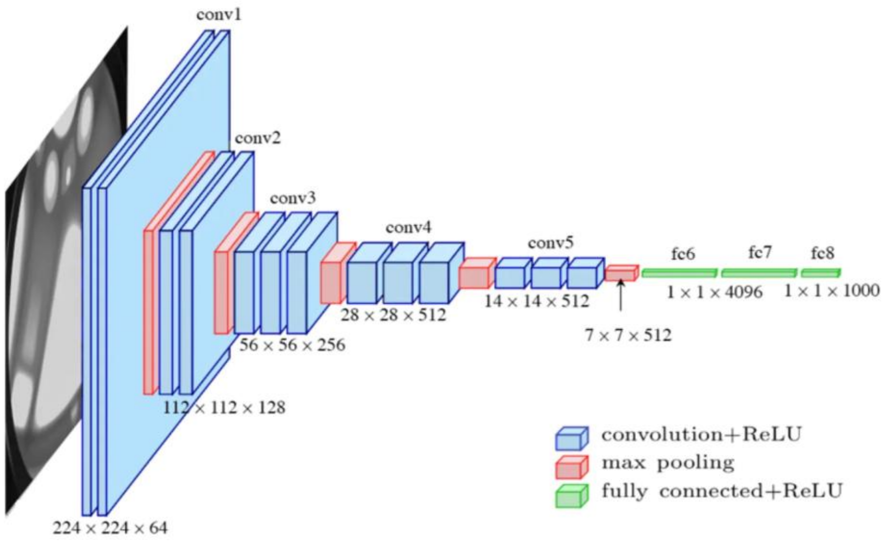


Fig Title: Diagram VGG16 architecture

- VGG (Visual Geometry Group) developed at Oxford University (2014)
- Known for **deep convolutional neural networks** with simple architecture
- Two main versions: **VGG16** (16 layers) and **VGG19** (19 layers)
- Key idea: Use **small 3×3 convolution filters** stacked to increase depth

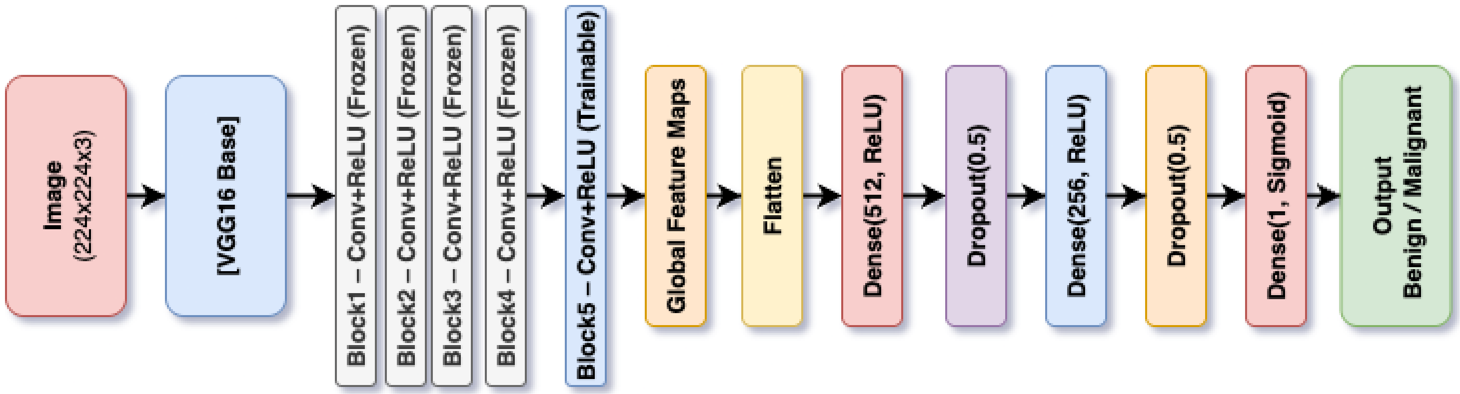


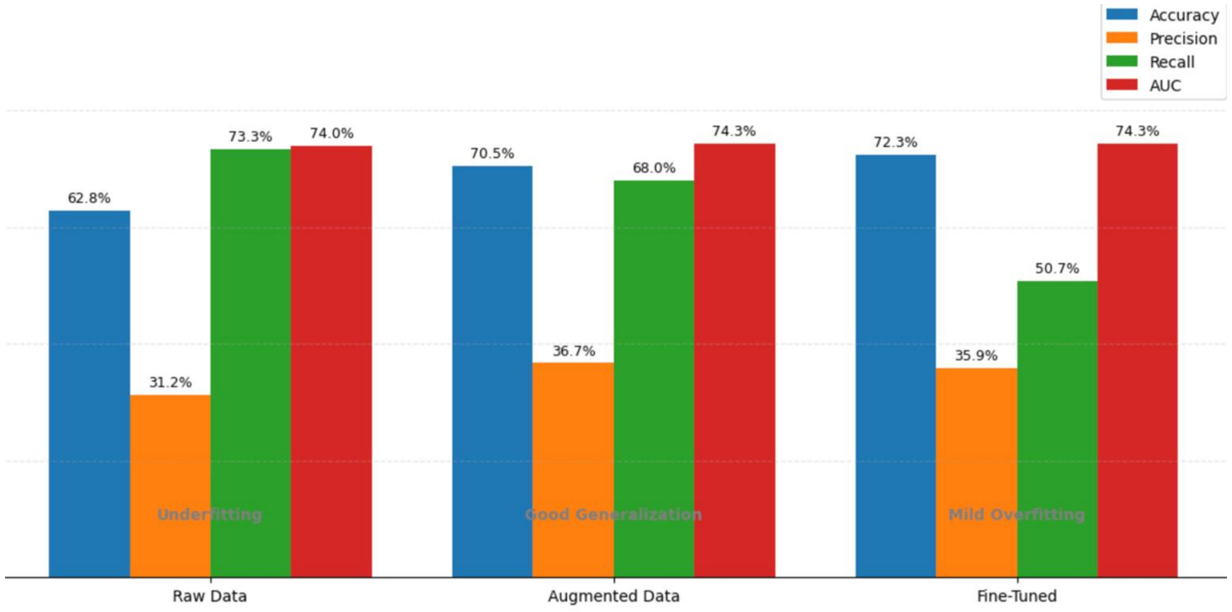
Fig Title: Fine-Tuned Model Block Diagram VGG16

VGG Architecture(continued)

Result

Case	Accuracy	Precision	Recall	AUC
Raw Data	0.6280	0.3125	0.7333	0.7395
Augmented Data	0.7045	0.3669	0.6800	0.7426
Fine-Tuned Model	0.7230	0.3585	0.5067	0.7432

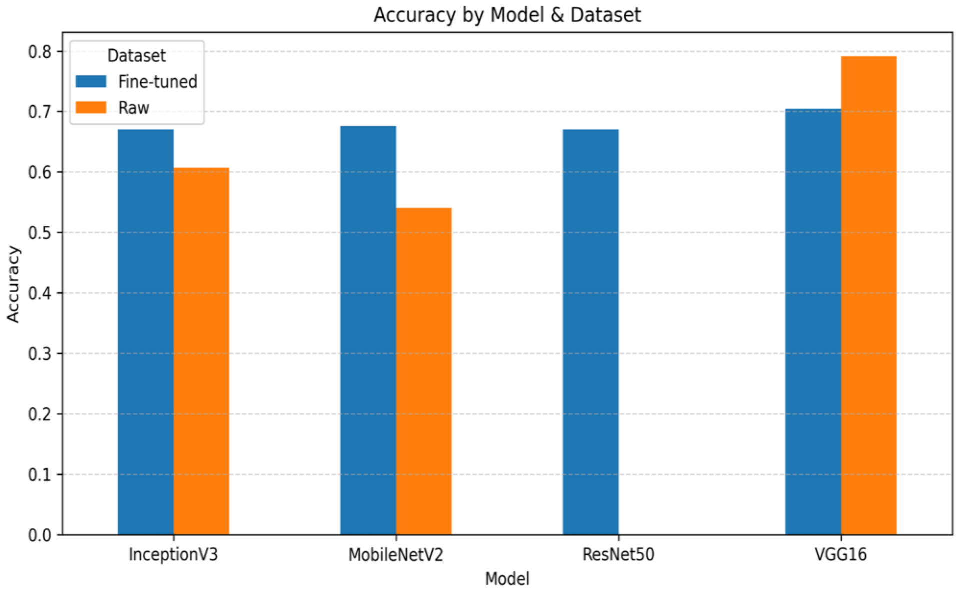
- **Raw Data** – **baseline** performance, **high recall** but low precision.
- **Augmented Data** – **improved** accuracy and **balance** between recall & precision.
- **Fine-Tuned Model** – **best accuracy** and **AUC after unfreezing** deeper layers.



- **Data augmentation** significantly improved generalization.
- **Fine-tuning** increased accuracy (**72.3%**) and AUC (**0.7432**).
- Trade-off observed between **precision vs recall** across settings.

Stage	Accuracy	Precision	Recall	AUC	Interpretation
Raw Data	62.80%	31.25%	73.33%	73.95%	Underfitting
Augmented Data	70.45%	36.69%	68.00%	74.26%	Good Generalization
Fine-Tuned	72.30%	35.85%	50.67%	74.32%	Mild Overfitting

Result & Discussion



Model	Accuracy	Precision	Recall	AUC
InceptionV3 (Raw)	0.6069	0.2989	0.7333	0.7157
InceptionV3 (Aug)	0.6702	0.3288	0.64	0.7084
InceptionV3 (Fine-tuned)	0.6702	0.3355	0.68	0.7368
ResNet18	0.6702	0.3355	0.68	0.7368
MobileNetV2 (Raw)	0.5409	0.2609	0.72	0.7402
MobileNetV2 (Aug)	0.6544	0.325	0.6933	0.7402
MobileNetV2 (Fine-tuned)	0.6755	0.3421	0.6933	0.7527
VGG16 (Raw)	0.7916	0.4773	0.56	0.7828
VGG16 (Aug)	0.6544	0.325	0.6933	0.7402
VGG16 (Fine-tuned)	0.7045	0.3669	0.68	0.7426

Any questions, comments or suggestions?