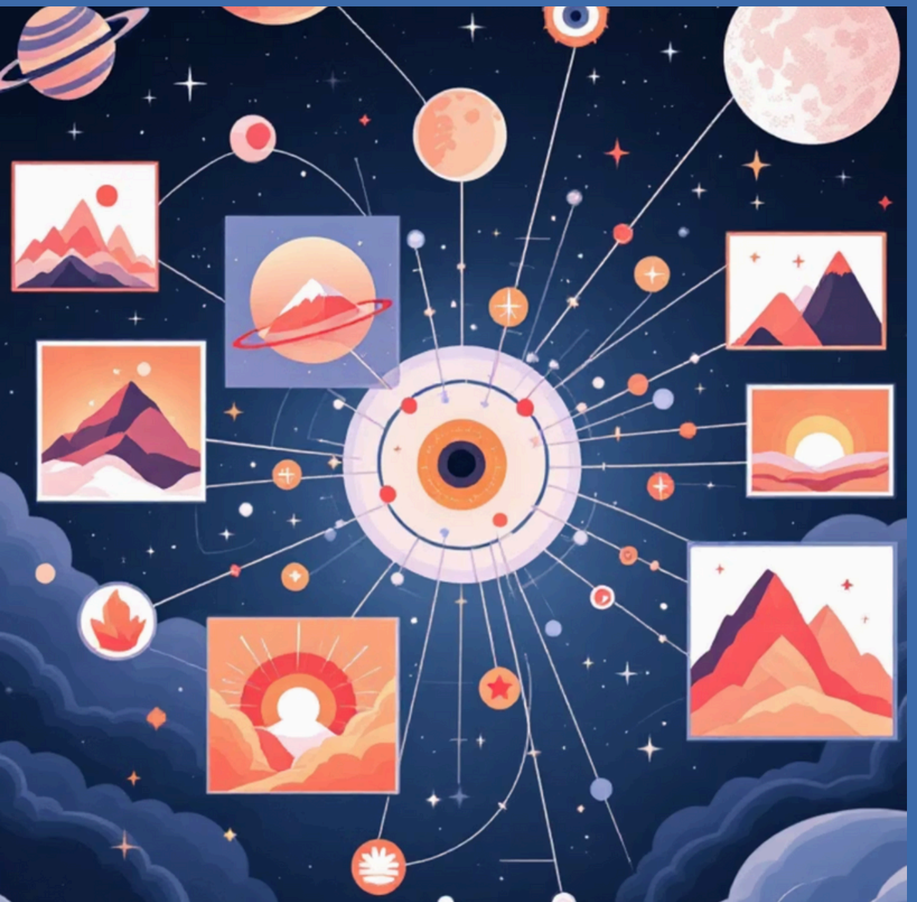
Deep Learning:

Image Classification using CNN and Transfer Learning

PRESENTED BY: AYSE K., NEKKY L., RAÍSSA F.

Overview



1. Data Preprocessing
2. CNN Model 1
3. CNN Model 2
4. MobileNetV2
5. VGG16
6. EfficientNetB0
7. CNN X Transfer Learning

Data Preprocessing

- **Validate Data Structure**

  Confirm shape (-1,32,32,3) ensuring RGB format

- **Check Class Balance**

  Verify equal distribution across all 10 classes

- **Normalize Pixel Values**

  Scale from [0,255] to [0,1] for better convergence

- **Apply Data Augmentation**

  Use ImageDataGenerator for rotation, zoom, flip operations

CIFAR-10:
A BENCHMARK DATASET CONTAINING 60,000 32×32 COLOR IMAGES ACROSS 10 CLASSES (6,000 IMAGES PER CLASS)

CNN

Model 1

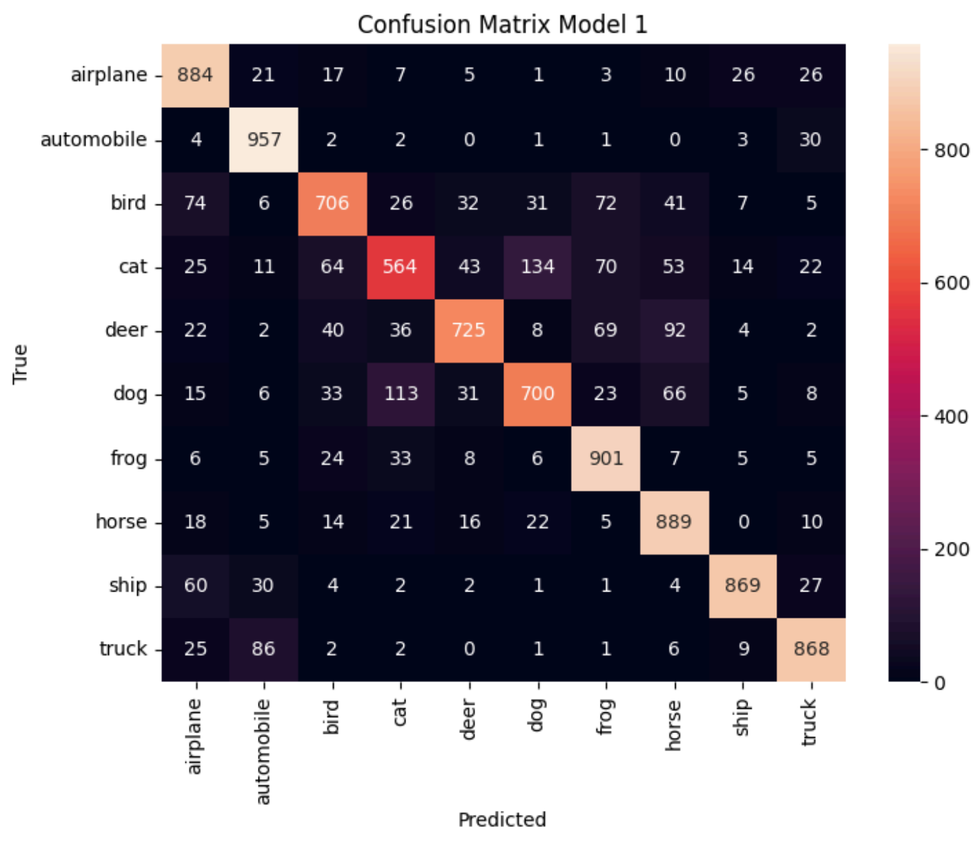
- OPTIMIZER: ADAM
- LOSS: SPARSE_CATEGORICAL_CROSS ENTROPY
- CALLBACK EARLY_STOPPING

Block	Description	Layers Included	Output Shape
1	**Conv Block 1**	Conv2D (32, ReLU) → BatchNorm → MaxPooling	(16, 16, 32)
2	**Conv Block 2**	Conv2D (64, ReLU) → BatchNorm → MaxPooling	(8, 8, 64)
3	**Conv Block 3**	Conv2D (128, ReLU) → BatchNorm → MaxPooling	(4, 4, 128)
4	**Flatten & Regularization**	Flatten → Dropout (0.5)	2048
5	**Fully Connected**	Dense (256, ReLU) → Dropout (0.5)	256
6	**Output Layer**	Dense (10, Softmax)	10 (class scores)

Evaluation

Model 1

```
Evaluation predicting with X_test
Accuracy: 0.8063
Precision: 0.8057
Recall: 0.8063
F1 Score: 0.8028
```



	precision	recall	f1-score	support
airplane	0.78	0.88	0.83	1000
automobile	0.85	0.96	0.90	1000
bird	0.78	0.71	0.74	1000
cat	0.70	0.56	0.62	1000
deer	0.84	0.72	0.78	1000
dog	0.77	0.70	0.73	1000
frog	0.79	0.90	0.84	1000
horse	0.76	0.89	0.82	1000
ship	0.92	0.87	0.89	1000
truck	0.87	0.87	0.87	1000
accuracy			0.81	10000
macro avg	0.81	0.81	0.80	10000
weighted avg	0.81	0.81	0.80	10000

CNN

Model 2

- OPTIMIZER:
ADAM

- LOSS:
SPARSE_CATEGORICAL_CROSS ENTROPY

- CALLBACK
EARLY_STOPPING
REDUCELRONPLATEAU

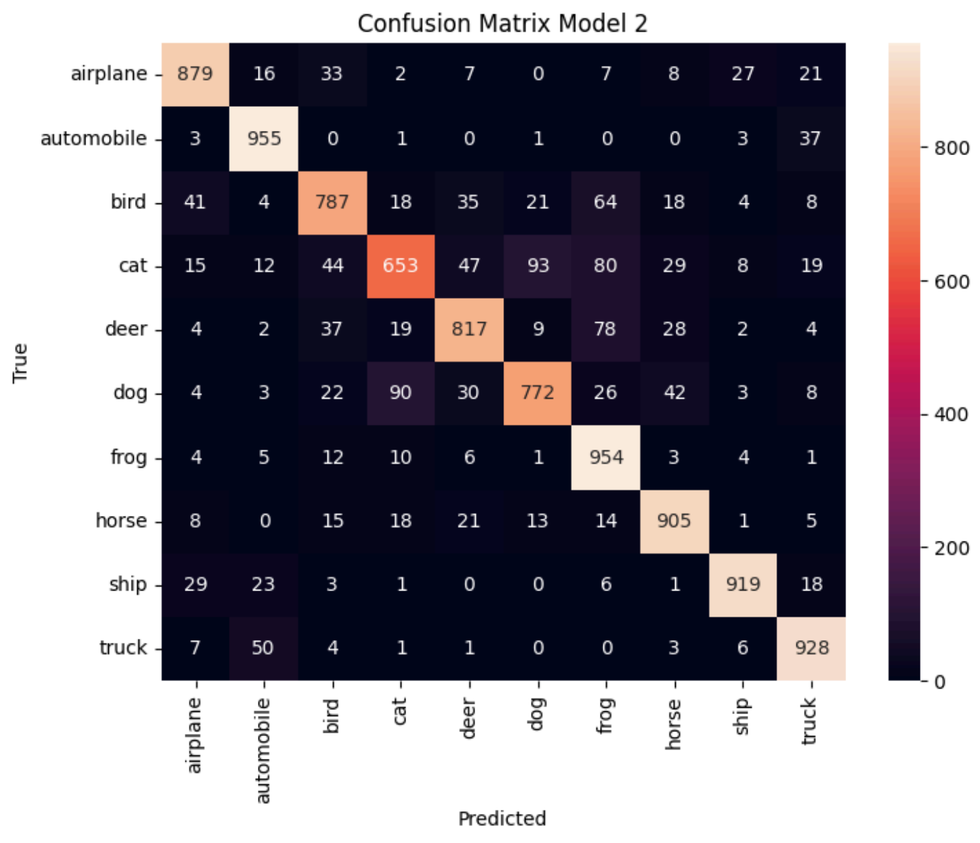
Block	Description	Layers Included	Output Shape
1	**Conv Block 1**	Conv2D (32, ReLU,L2) → BatchNorm → MaxPooling	(16, 16, 32)
2	**Conv Block 2**	Conv2D (64, ReLU, L2) → BatchNorm → MaxPooling	(8, 8, 64)
3	**Conv Block 3**	Conv2D (128, ReLU, L2) → BatchNorm → MaxPooling	(4, 4, 128)
4	**Conv Block 4**	Conv2D (256, ReLU, L2) → BatchNorm → MaxPooling	(2,2,256)
5	**Global Feat. Extract**	GlobalAveragePooling2D → Dropout (0.5)	256
6	**Fully Connected**	Dense (256, ReLU, L2) → Dropout (0.5)	256
7	**Output Layer**	Dense (10, Softmax)	10 (class scores)

Model 1 x Model 2

	Model 1	**Model 2**
Convolutional Blocks	3	4
Regularization	Dropout	Dropout L2
Final Feature Layer	Flatten (2048)	GlobalAverage (256)
Callbacks	EarlyStopping	EarlyStopping ReduceLRO

Evaluation

Model 2

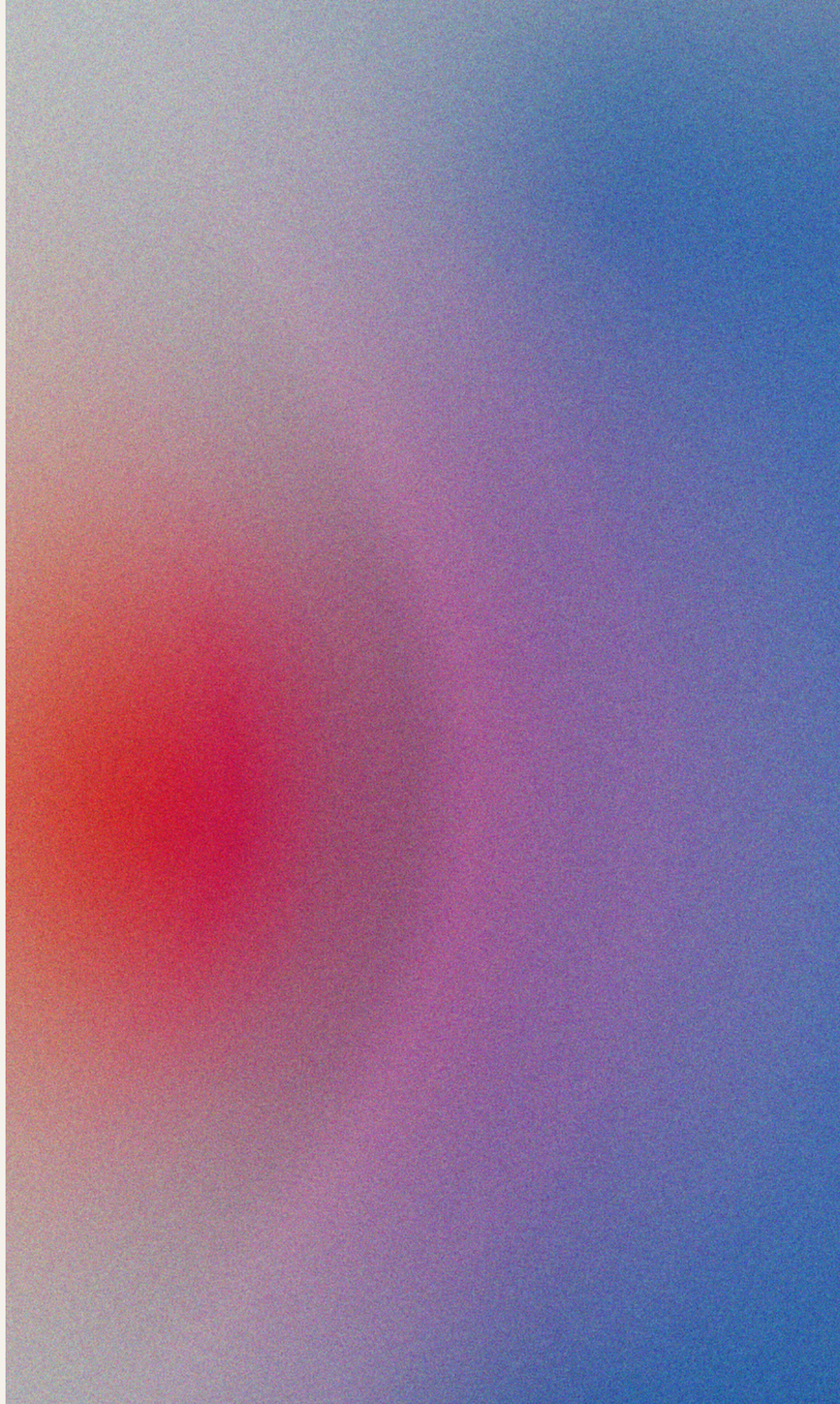



```
Evaluation predicting with X_test
Accuracy: 0.8569
Precision: 0.8572
Recall: 0.8569
F1 Score: 0.8549
```

	precision	recall	f1-score	support
airplane	0.88	0.88	0.88	1000
automobile	0.89	0.95	0.92	1000
bird	0.82	0.79	0.80	1000
cat	0.80	0.65	0.72	1000
deer	0.85	0.82	0.83	1000
dog	0.85	0.77	0.81	1000
frog	0.78	0.95	0.86	1000
horse	0.87	0.91	0.89	1000
ship	0.94	0.92	0.93	1000
truck	0.88	0.93	0.91	1000
accuracy			0.86	10000
macro avg	0.86	0.86	0.85	10000
weighted avg	0.86	0.86	0.85	10000

Transfer Learning

- MobileNetV2
- VGG16
- EfficientNetB0



MobileNetV2

ARCHITECTURE SPECIFICATIONS:

- 53 LAYERS
- (51 CONVOLUTIONAL + 2 FULLY CONNECTED)
- DEPTHWISE SEPARABLE CONVOLUTIONS (3X3 + 1X1 POINTWISE)
- GLOBAL AVERAGE POOLING FOR SPATIAL REDUCTION
- RELU6 ACTIVATION FUNCTIONS

Fine Tune

Training Stage	Phase 1: Transfer Learning	Phase 2: Fine-tuning
MobileNetV2 Layers Status	All layers frozen	Layers after index 100 frozen (~20% of network)
Learning Rate	1e- 3 (Adam)	1e-5 (Reduced)
Training Epochs	Up to 20 Epochs	Additional 10 epochs
Objective	Learn CIFAR-10 classification	Adapt higher-level pretrained features
Expected imrpovement	Baseline performance	3-5% Accuracy gain

Metrics

86.1%	90.4%	+4.3%
Initial Accuracy	Fine-tuned Accuracy	Performance Gain

Phase 1: Transfer Learning

- Accuracy: 86.1%
- Training time: ~6 minutes
- Converged around epoch 18
- Stable loss reduction

Phase 2: Fine-tuning

- Final accuracy: 90.4%
- Additional time: ~7 minutes
- 10 fine-tuning epochs
- Consistent improvement

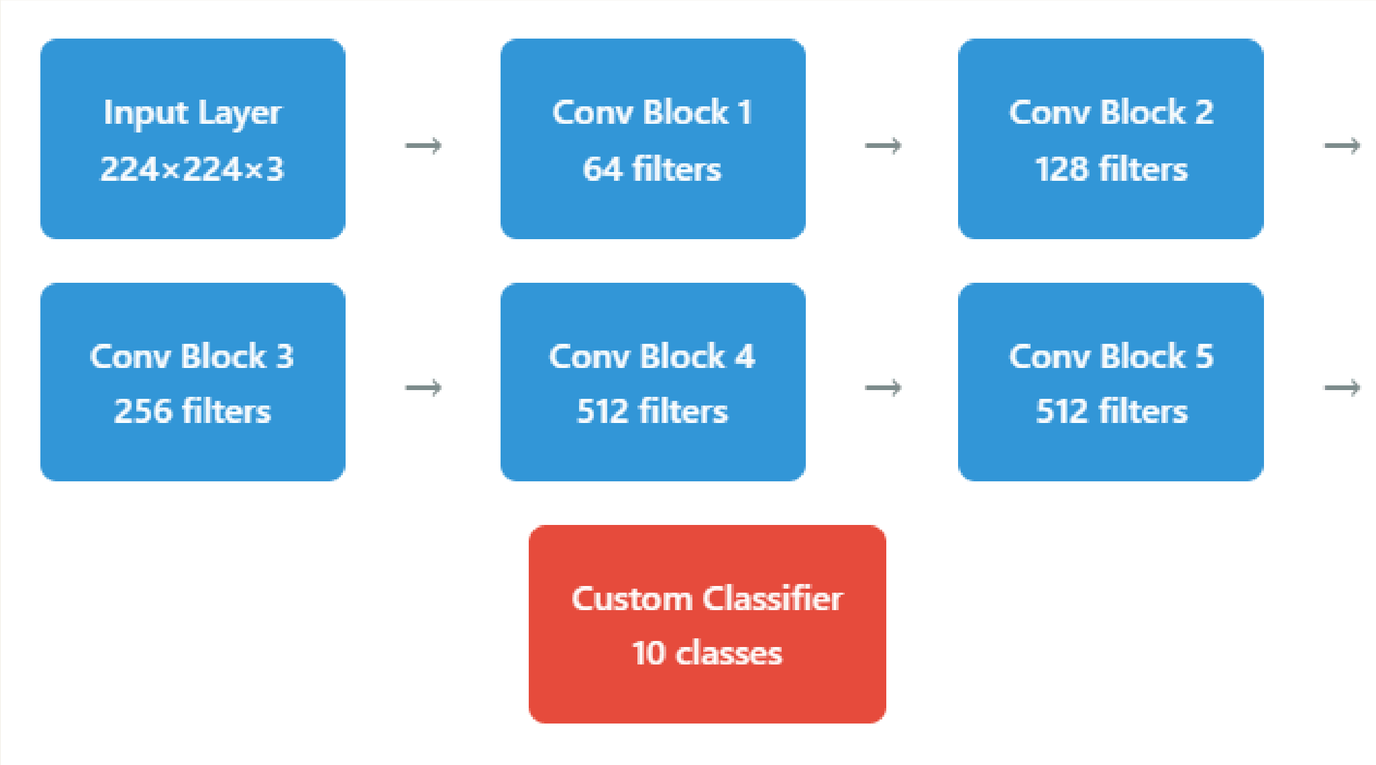
VGG16 ARCHITECTURE OVERVIEW

Architecture Specifications:

- 16 layers

(13 convolutional + 3 fully connected)

- 3×3 convolution filters throughout
- Max pooling for spatial reduction
- ReLU activation functions



VGG16 Fine Tune

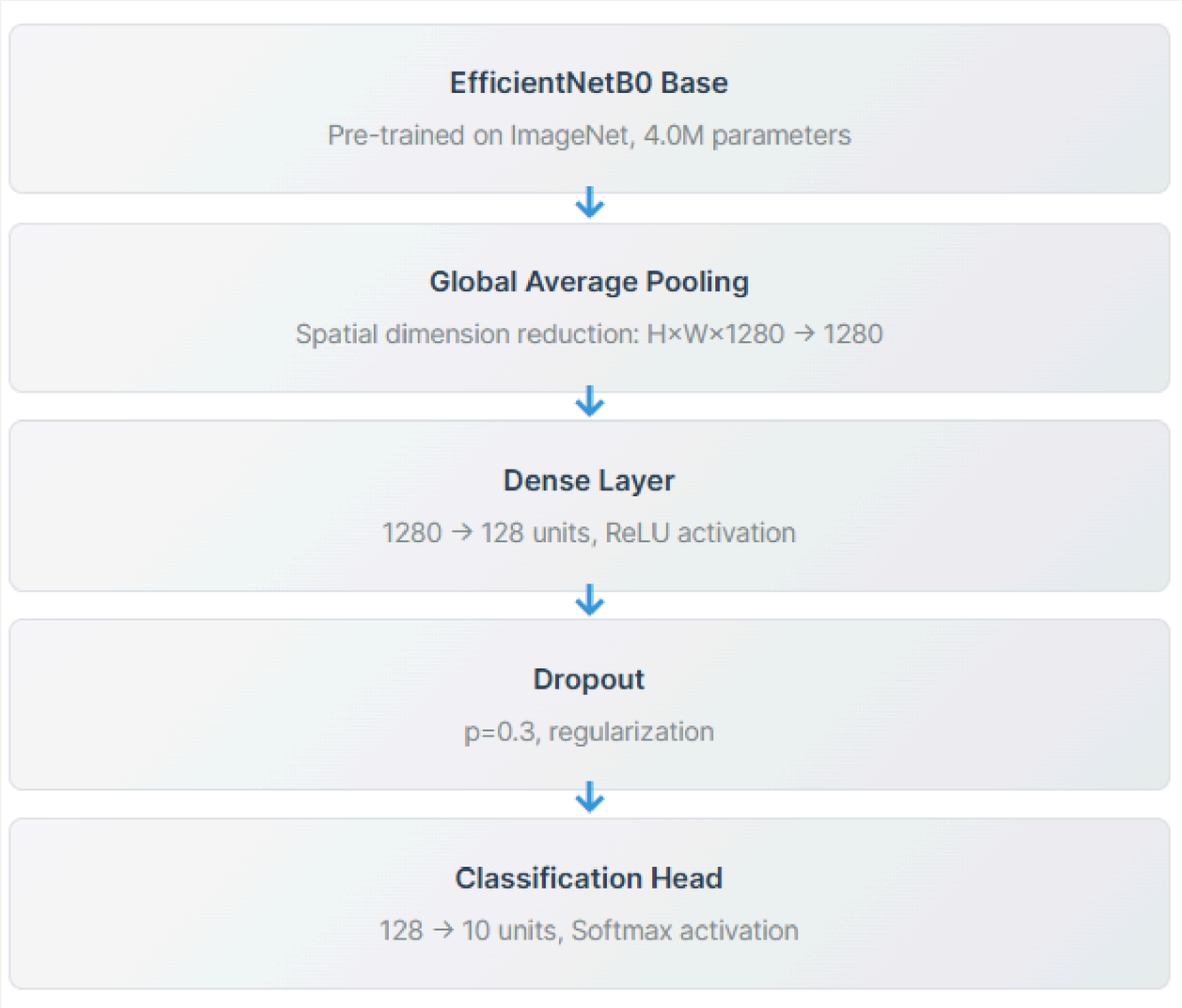
Training Stage	Phase 1: Transfer Learning	Phase 2: Fine-tuning
VGG16 Layers Status	All layers frozen	Last 4 layers unfrozen
Learning Rate	1e-3 (Standard)	1e-5 (Reduced)
Training Epochs	Up to 20 epochs	Additional 5 epochs
Objective	Learn CIFAR-10 classification	Adapt pre-trained features
Expected Improvement	Baseline performance	3-5% accuracy gain

VGG16 Model Performance Metrics

```
🔍 VGG16 Transfer Learning Results:
Accuracy:  0.7696
Precision: 0.7864
Recall:    0.7696
F1 Score:  0.7715
```

```
🔍 Metrics after Fine-Tuning VGG16:
Accuracy:  0.8517 (Improvement: +0.0821)
Precision: 0.8566 (Improvement: +0.0702)
Recall:    0.8517 (Improvement: +0.0821)
F1 Score:  0.8521 (Improvement: +0.0807)
```

EfficientNetB0 ARCHITECTURE OVERVIEW



EfficientNetB0 Fine Tune

Stage 1: Transfer Learning

Objective: Adapt pre-trained features to CIFAR-10

Base Model	Frozen (EfficientNetB0)
Trainable Params	5,130
Learning Rate	1e-3
Epochs	10
Optimizer	Adam

Stage 2: Fine-Tuning

Objective: Optimize high-level feature representations

Unfrozen Layers	Top 10% (40+ layers)
Trainable Params	404,444
Learning Rate	1e-5
Additional Epochs	3
Strategy	Conservative fine-tuning

EfficientNetB0 Model Performance Metrics

Transfer Learning Results:		After fine tuning:
Accuracy 83.10%		Accuracy 83.5%
Precision 83.18%		Precision 84.2%
Recall 83.10%		Recall 85.4%
F1-Score 83.11%		F1-Score 84.7%

Transfer Learning X CNN

CONVOLUTIONAL NEURAL NETWORKS:

- LEARN LOW-LEVEL → HIGH-LEVEL IMAGE FEATURES
- REQUIRE LARGE DATASETS + LONG TRAINING TIME

TRANSFER LEARNING:

- USES A PRETRAINED CNN
- LEVERAGES KNOWLEDGE FROM IMAGENET
- ONLY RETRAINS TOP ALYERS OR FINE-TUNES LAST’ BLOCKS

OUR RESULTS:

- CUSTOM CNNS ACHIEVED ~70-80% ACCURACY
- TRANSFER LEARNING BOOSETED TO ~85-90%+ ACCURACY
- FINE-TUNING ADDED AN EXTRA 3-5% PERFORMANCE GAIN BEYOND BASELINE TRANSFER LEARNING
- CNN FEATURE EXTRACTION + TRANSFER LEARNING
  → BEST BALANCE OF SPEED, ACCURACY AND GENERALIZATION

Thank you