

---

## DATASETS, PARAMETERS, AND MODELS

### DATASETS

We used CIFAR-10 (Krizhevsky et al., 2009) that has 50,000 training  $32 \times 32$  images evenly distributed between 10 categories and 10,000 test images, CIFAR-100 (Krizhevsky et al., 2009) has the same number of images as CIFAR-10 but the images are under 100 classes. The largest dataset we used for inference is ImageNet (Russakovsky et al., 2015) with about 1.2 million training images, 50,000 validation images, and 100,000 test images that are distributed into 1000 object classes.

For all the aforementioned datasets the preprocessing phase consists of three steps summarized in the table below:

Table 1: Preprocessing Parameters

<b>CIFAR-10</b>	Horizontal Flip with P=0.5
	Images are cropped at a random place
	Normalization (three channels) mean = [0.485, 0.456, 0.406], std = [0.229, 0.224, 0.225]
<b>CIFAR-100</b>	Horizontal flip with P=0.5
	Images are cropped at a random place
	Normalization (three channels) mean=[0.507, 0.487, 0.441], std=[0.267, 0.256, 0.276]
<b>ImageNet</b>	Horizontal flip with P=0.5
	Images are cropped at a random place
	Normalization (three channels) mean=[0.485, 0.456, 0.406], std=[0.229, 0.224, 0.225]

### TRAINING PARAMETERS

For the three dataset and three models we used the training parameters are reported in the table 2.

Table 2: Training Parameters

	VGG16		ResNet152, ResNet50		MobileNetV2	
Parameters	CIFAR-10	CIFAR-100	CIFAR-10	CIFAR-100	CIFAR-10	CIFAR-100
Batch-size	128	128	128	128	128	128
Learning Rate	0.01	0.01	0.01	0.01	0.01	0.01
Stepsize	200	60	200	30	200	60
lambda	0.1	0.02	0	0.02	0.1	0.2
Momentum	0	0.9	0	0.9	0	0.9
Weight Decay	0	0.0005	0	0.0005	0	0.0005
Dropout	0.5	0.5	0.5	0.5	0.2	0.2

For inference implementations the pretrained models with the above are used.

### MODELS

We used different models for our experiments. VGG16 (Simonyan & Zisserman, 2014), ResNet152, ResNet50 (He et al., 2016) are chosen because of their enormous parameter size. However, MobileNetV2 (Sandler et al., 2018; Howard et al., 2017) is implemented and analyzed because it is the worst-case benchmark for our model as it reduces its linear operations considerably.

Model details including the description, output shape and number of parameters are reported in table 3, ??, 5.

Table 3: VGG16 Architecture

---

Layer (type)	Output Shape	Param #
input_4 (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
flatten (Flatten)	(None, 25088)	0
fc1 (Dense)	(None, 4096)	102764544
fc2 (Dense)	(None, 4096)	16781312
predictions (Dense)	(None, 1000)	4097000
Total params: 138,357,544		
Trainable params: 138,357,544		
Non-trainable params: 0		

Table 4: Architectures for ImageNet. Building blocks are shown in brackets, with the numbers of blocks stacked. Downsampling is performed by conv3\_1, conv4\_1, and conv5\_1 with a stride of 2 (He et al., 2016)

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$
conv3_x	28×28	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 8$
conv4_x	14×14	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 23$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 36$
conv5_x	7×7	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		$1.8 \times 10^9$	$3.6 \times 10^9$	$3.8 \times 10^9$	$7.6 \times 10^9$	$11.3 \times 10^9$

Table 5: MobileNetV2 Architecture

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	(None, 224, 224, 3)	0

Conv1_pad (ZeroPadding2D)	(None, 225, 225, 3)	0
Conv1 (Conv2D)	(None, 112, 112, 32)	864
Conv1_relu (ReLU)	(None, 112, 112, 32)	0
expanded_conv_depthwise (DepthwiseConvolution)	(None, 112, 112, 32)	288
expanded_conv_depthwise_relu (ReLU)	(None, 112, 112, 32)	0
expanded_conv_project (Conv2D)	(None, 112, 112, 16)	512
block_1_expand (Conv2D)	(None, 112, 112, 96)	1536
block_1_expand_relu (ReLU)	(None, 112, 112, 96)	0
block_1_pad (ZeroPadding2D)	(None, 113, 113, 96)	0
block_1_depthwise (DepthwiseCon)	(None, 56, 56, 96)	864
block_1_depthwise_relu (ReLU)	(None, 56, 56, 96)	0
block_1_project (Conv2D)	(None, 56, 56, 24)	2304
block_2_expand (Conv2D)	(None, 56, 56, 144)	3456
block_2_expand_relu (ReLU)	(None, 56, 56, 144)	0
block_2_depthwise (DepthwiseConvolution)	(None, 56, 56, 144)	1296
block_2_depthwise_relu (ReLU)	(None, 56, 56, 144)	0
block_2_project (Conv2D)	(None, 56, 56, 24)	3456
block_2_add (Add)	(None, 56, 56, 24)	0
block_3_expand (Conv2D)	(None, 56, 56, 144)	3456
block_3_expand_relu (ReLU)	(None, 56, 56, 144)	0
block_3_pad (ZeroPadding2D)	(None, 57, 57, 144)	0
block_3_depthwise (DepthwiseConvolution)	(None, 28, 28, 144)	1296
block_3_depthwise_relu (ReLU)	(None, 28, 28, 144)	0
block_3_project (Conv2D)	(None, 28, 28, 32)	4608
block_4_expand (Conv2D)	(None, 28, 28, 192)	6144
block_4_expand_relu (ReLU)	(None, 28, 28, 192)	0
block_4_depthwise (DepthwiseConvolution)	(None, 28, 28, 192)	1728
block_4_depthwise_relu (ReLU)	(None, 28, 28, 192)	0
block_4_project (Conv2D)	(None, 28, 28, 32)	6144
block_4_add (Add)	(None, 28, 28, 32)	0
block_5_expand (Conv2D)	(None, 28, 28, 192)	6144
block_5_expand_relu (ReLU)	(None, 28, 28, 192)	0
block_5_depthwise (DepthwiseConvolution)	(None, 28, 28, 192)	1728
block_5_depthwise_relu (ReLU)	(None, 28, 28, 192)	0
block_5_project (Conv2D)	(None, 28, 28, 32)	6144
block_5_add (Add)	(None, 28, 28, 32)	0
block_6_expand (Conv2D)	(None, 28, 28, 192)	6144
block_6_expand_relu (ReLU)	(None, 28, 28, 192)	0
block_6_pad (ZeroPadding2D)	(None, 29, 29, 192)	0
block_6_depthwise (DepthwiseConvolution)	(None, 14, 14, 192)	1728
block_6_depthwise_relu (ReLU)	(None, 14, 14, 192)	0
block_6_project (Conv2D)	(None, 14, 14, 64)	12288
block_7_expand (Conv2D)	(None, 14, 14, 384)	24576
block_7_expand_relu (ReLU)	(None, 14, 14, 384)	0
block_7_depthwise (DepthwiseConvolution)	(None, 14, 14, 384)	3456
block_7_depthwise_relu (ReLU)	(None, 14, 14, 384)	0
block_7_project (Conv2D)	(None, 14, 14, 64)	24576
block_7_add (Add)	(None, 14, 14, 64)	0
block_8_expand (Conv2D)	(None, 14, 14, 384)	24576
block_8_expand_relu (ReLU)	(None, 14, 14, 384)	0
block_8_depthwise (DepthwiseConvolution)	(None, 14, 14, 384)	3456

block_8_depthwise_relu (ReLU)	(None, 14, 14, 384)	0
block_8_project (Conv2D)	(None, 14, 14, 64)	24576
block_8_add (Add)	(None, 14, 14, 64)	0
block_9_expand (Conv2D)	(None, 14, 14, 384)	24576
block_9_expand_relu (ReLU)	(None, 14, 14, 384)	0
block_9_depthwise (DepthwiseConvolution)	(None, 14, 14, 384)	3456
block_9_depthwise_relu (ReLU)	(None, 14, 14, 384)	0
block_9_project (Conv2D)	(None, 14, 14, 64)	24576
block_9_add (Add)	(None, 14, 14, 64)	0
block_10_expand (Conv2D)	(None, 14, 14, 384)	24576
block_10_expand_relu (ReLU)	(None, 14, 14, 384)	0
block_10_depthwise (DepthwiseConvolution)	(None, 14, 14, 384)	3456
block_10_depthwise_relu (ReLU)	(None, 14, 14, 384)	0
block_10_project (Conv2D)	(None, 14, 14, 96)	36864
block_11_expand (Conv2D)	(None, 14, 14, 576)	55296
block_11_expand_relu (ReLU)	(None, 14, 14, 576)	0
block_11_depthwise (DepthwiseConvolution)	(None, 14, 14, 576)	5184
block_11_depthwise_relu (ReLU)	(None, 14, 14, 576)	0
block_11_project (Conv2D)	(None, 14, 14, 96)	55296
block_11_add (Add)	(None, 14, 14, 96)	0
block_12_expand (Conv2D)	(None, 14, 14, 576)	55296
block_12_expand_relu (ReLU)	(None, 14, 14, 576)	0
block_12_depthwise (DepthwiseConvolution)	(None, 14, 14, 576)	5184
block_12_depthwise_relu (ReLU)	(None, 14, 14, 576)	0
block_12_project (Conv2D)	(None, 14, 14, 96)	55296
block_12_add (Add)	(None, 14, 14, 96)	0
block_13_expand (Conv2D)	(None, 14, 14, 576)	55296
block_13_expand_relu (ReLU)	(None, 14, 14, 576)	0
block_13_pad (ZeroPadding2D)	(None, 15, 15, 576)	0
block_13_depthwise (DepthwiseConvolution)	(None, 7, 7, 576)	5184
block_13_depthwise_relu (ReLU)	(None, 7, 7, 576)	0
block_13_project (Conv2D)	(None, 7, 7, 160)	92160
block_14_expand (Conv2D)	(None, 7, 7, 960)	153600
block_14_expand_relu (ReLU)	(None, 7, 7, 960)	0
block_14_depthwise (DepthwiseConvolution)	(None, 7, 7, 960)	8640
block_14_depthwise_relu (ReLU)	(None, 7, 7, 960)	0
block_14_project (Conv2D)	(None, 7, 7, 160)	153600
block_14_add (Add)	(None, 7, 7, 160)	0
block_15_expand (Conv2D)	(None, 7, 7, 960)	153600
block_15_expand_relu (ReLU)	(None, 7, 7, 960)	0
block_15_depthwise (DepthwiseConvolution)	(None, 7, 7, 960)	8640
block_15_depthwise_relu (ReLU)	(None, 7, 7, 960)	0
block_15_project (Conv2D)	(None, 7, 7, 160)	153600
block_15_add (Add)	(None, 7, 7, 160)	0
block_16_expand (Conv2D)	(None, 7, 7, 960)	153600
block_16_expand_relu (ReLU)	(None, 7, 7, 960)	0
block_16_depthwise (DepthwiseConvolution)	(None, 7, 7, 960)	8640
block_16_depthwise_relu (ReLU)	(None, 7, 7, 960)	0
block_16_project (Conv2D)	(None, 7, 7, 320)	307200
Conv_1 (Conv2D)	(None, 7, 7, 1280)	409600
out_relu (ReLU)	(None, 7, 7, 1280)	0

---

global_average_pooling2d (GlobalAveragePool)	(None, 1280)	0
Logits (Dense)	(None, 1000)	1281000
Total params: 3,538,984		
Trainable params: 3,504,872		
Non-trainable params: 34,112		

## REFERENCES

- Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.
- Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.
- Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. online: <http://www.cs.toronto.edu/kriz/cifar.html>, 2009.
- Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.
- Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4510–4520, 2018.
- Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.