Advanced Training Techniques for CNNs

Parth Agrawal

January 2026

In this report, we will discuss several techniques used to enhance CNN models.

1 Data Augmentation

When we work with images in a dataset, the model often learns to recognize features specific to the images themselves, rather than the standard features we would want it to recognize. For example, if we train our model to recognize the sun in the sky, but the sun is always positioned in the same area of the image, the model may learn that the sun is only it if it exists at that particular area.

To prevent such issues from occurring, we utilize data augmentation. In this technique, we alter/transform the images in such a way that the broad features are preserved and dominant over others, like changing orientation or lighting.

There are two categories of augmentation techniques -

- **Geometric Augmentation** - Rotation (invariance to orientation), Flipping (Horizontal, Vertical Symmetry) and cropping.
- **Photometric Augmentation** - Colour Jittering (adjusting brightness, contrast, saturation, etc. to handle variations in lighting.)

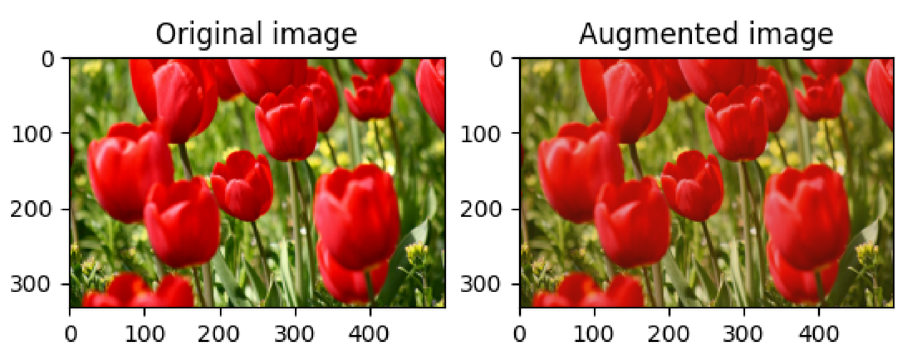

Figure 1: Random Contrast Changes (Data Augmentation Technique)

Transformations are often represented as matrix operations. For a rotation of θ:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

These techniques create a suitably diverse dataset of images.

2 Transfer Learning & Fine-Tuning

In image classification, model sizes often become quite large due to the need to capture minute details. When such models, which contain millions of parameters, are to be trained, it becomes tedious and impractical to do so for each new problem.

A solution is to use large general image classification models like MobileNet or ResNet, and build a fully connected layer on top to suit our problem. The primary benefit is that we can load the pre-trained weights into the model (which have already learned to capture the main details) and then train the top. This is called transfer learning.

There are two broad types of transfer learning -

- **Feature Extraction -** In this technique, the pre-trained model is frozen entirely and only the top layers are trained.
- **Fine-Tuning -** In this technique, either partially or entirely, the pre-trained model is unfrozen and allowed to participate in the training process. Depending on the number of unfrozen layers, the model can be partially or fully fine-tuned.

The learning process remains the same, but the initial weights W are set to $W_{pretrained}$ instead of W_{random}. During fine-tuning, we use a much smaller learning rate η to avoid destroying the pre-existing knowledge:

$$W_{new} = W_{old} - \eta \cdot \nabla L(W)$$

3 Batch Normalization

When data is trained in deep neural networks, the distribution of inputs to each layer changes as the weights of previous layers change (Internal Covariate Shift). This leads to significantly different activation values, resulting in unstable networks and vanishing/exploding gradients. To prevent this we use something called Batch Normalization.

Batch Normalization “re-centres and re-scales the data in every branch. This happens by adjusting the activations of each layer based on the mean and variance of the mini-batch, and it also learns scale and shift parameters to refine outputs.

Over long models, this makes training significantly faster and much more stable.

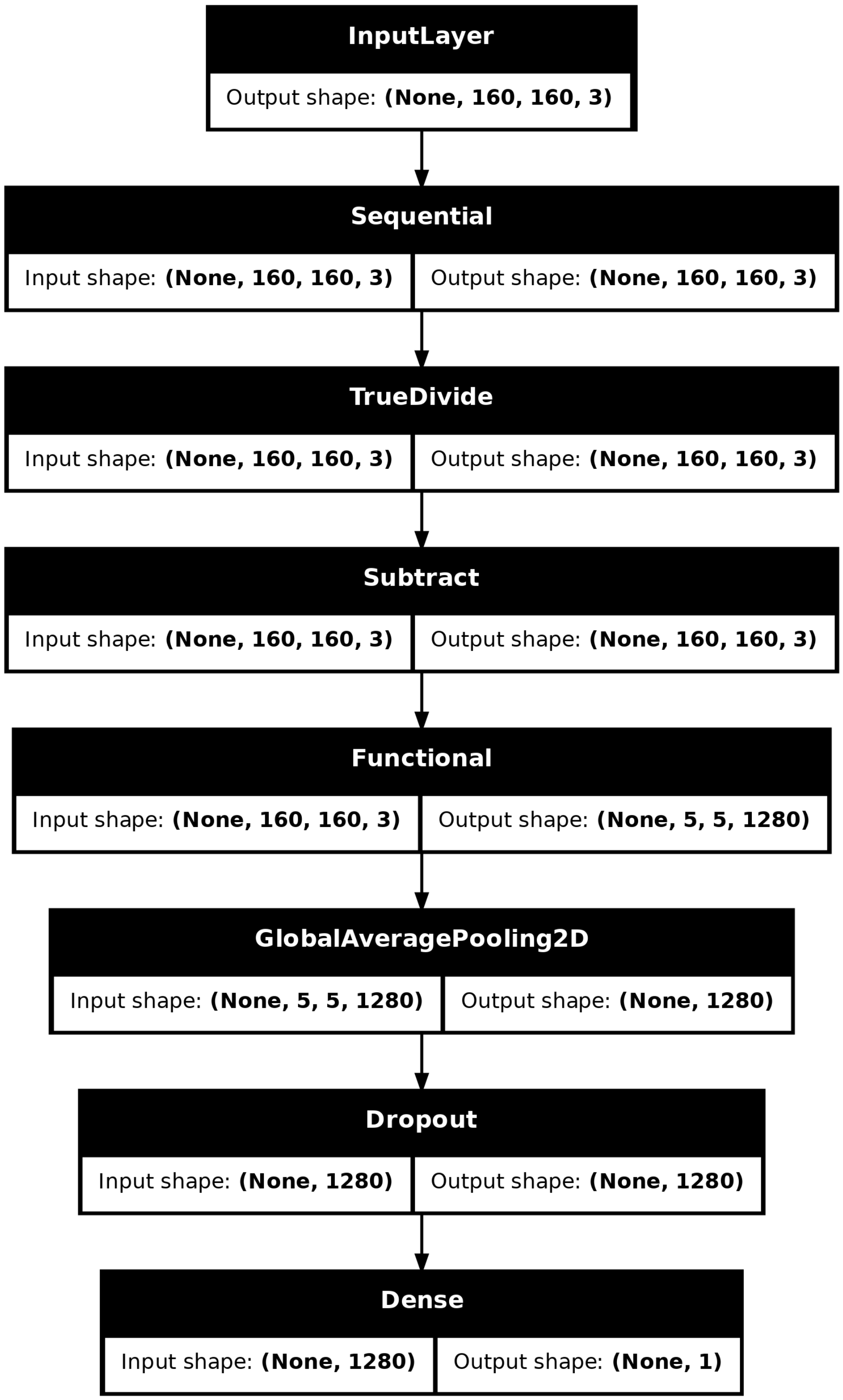

Figure 2: An example of a transfer learning model on MobileNetV2 (Functional)

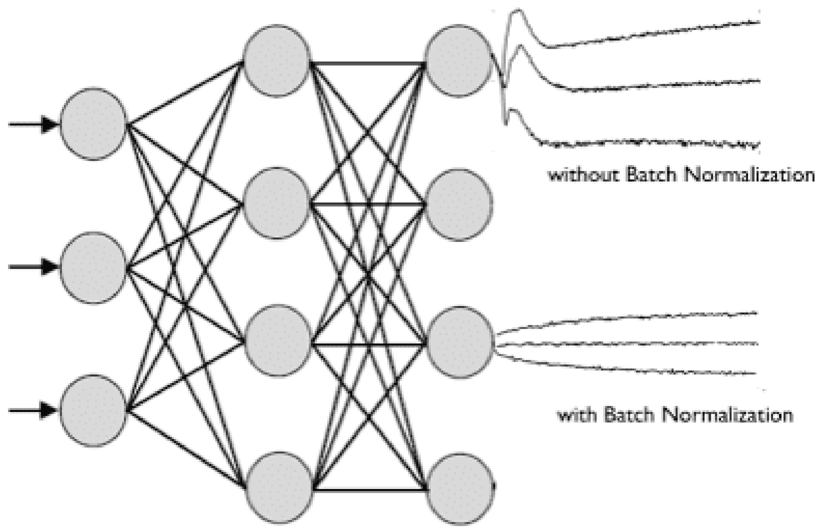

Figure 3: Effect of Batch Normalization on Neurons Output Distribution

4 Dropout

Dropout is a technique used primarily to tackle the overfitting problem. As the network size increases, especially at the fully connected layers in CNNs, the problem of overfitting becomes increasingly evident.

What dropout does is randomly "kill" (set activation to zero) some neurons during the training process. This forces the neural network to learn broad features rather than memorize the specific form of any image.

Usually, BatchNorm and Dropout are used together in CNNs to make a fast and accurate model. However, one must apply dropout carefully and not be overly reliant on it, as it may cause adverse effects on the model.

5 The Degradation Problem and ResNet

As models become deeper, we encounter the problem of degradation, where accuracy initially increases but then decreases. This is not *overfitting* since training accuracy also decreases.

ResNet was designed to solve this problem. It allows the network to learn "residuals" (changes in input and output) rather than entirely new mappings. This is achieved using skip connections.

Skip connections involve skipping one or more layers during the training process.

ResNet, developed by Kaiming He and his team at Microsoft Research, was the

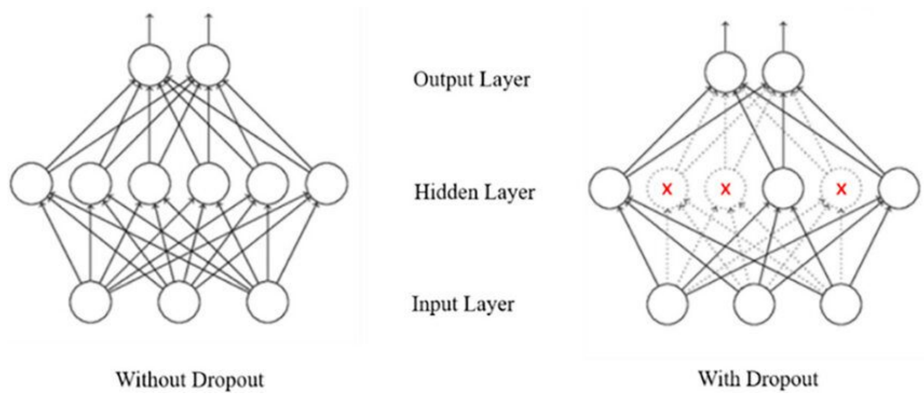

Figure 4: Impact of Dropout on Neural Netwrok Connections

first to utilize this solution. This led ResNet to win the first position at ILSVRC 2015.

6 References

- TensorFlow. (2024, July 19). Data augmentation (Tutorial). TensorFlow Core. https://www.tensorflow.org/tutorials/images/data_augmentation
- TensorFlow. (2024, August 16). Transfer learning and fine-tuning. TensorFlow. https://www.tensorflow.org/tutorials/images/transfer_learning
- Shubham. (2025, April 28) Normalization and Dropout: A Regularisation Approach. LearnOpenCV. https://learnopencv.com/batch-normalization-and-dropout-as-regularizers/
- He, K., Zhang, X., Ren, S., & Sun, J. (2015). Deep residual learning for image recognition (arXiv:1512.03385) [Preprint]. arXiv. https://doi.org/10.48550/arXiv.1512.03385