Image Augmentation Techniques

Abhijit Singh Jowhari

Contents

1 Data Augmentation **1**
- 1.1 Introduction . . . 1
- 1.2 Challenges in Computer Vision Problems . . . 2
  - 1.2.1 Image Variations . . . 2
  - 1.2.2 Class Imbalance and Few Images . . . 2
  - 1.2.3 Domain Shift . . . 2
  - 1.2.4 Data Remembering (Overfitting) . . . 2

2 Classical Techniques for Image Augmentation **2**
- 2.1 Flipping and Rotating . . . 2
- 2.2 Cropping and Resizing . . . 2
- 2.3 Color Jittering . . . 3
- 2.4 Adding Noise . . . 3
- 2.5 Image Warping . . . 3
- 2.6 Random Erasing . . . 3

3 Advanced Techniques **3**
- 3.1 Cutout . . . 3
- 3.2 Mixup Augmentation . . . 4
- 3.3 Cutmix . . . 5
- 3.4 Augmix . . . 5

4 Summary **6**

5 References **6**

1 Data Augmentation

1.1 Introduction

Deep learning has been doing some amazing things in the field of computer vision. But there's a catch: it needs loads of images to work its magic! Collecting a ton of images is not exactly a walk in the park. However, there are some cool image augmentation techniques that can help us out. Understanding these

techniques is super important for improving your computer vision tasks. So, let's dive in!

1.2 Challenges in Computer Vision Problems

1.2.1 Image Variations

Images can vary in many ways, such as lighting, pose, scale, and occlusion. These variations can make it difficult for computer vision models to generalize to new data.

1.2.2 Class Imbalance and Few Images

In many object detection and classification tasks, the number of images in each class is not balanced, and some classes may have only a few examples. This can make it difficult for models to learn to recognize all classes equally well.

1.2.3 Domain Shift

A model trained on one dataset or in one environment may not perform well when applied to new, unseen data or environments. This is due to the distribution of the data being different in the new environment.

1.2.4 Data Remembering (Overfitting)

A larger set of learnable parameters in a deep learning model demands more data for training. If the number of parameters increases, the model may overfit by memorizing specific data points, leading to poor performance on new data.

2 Classical Techniques for Image Augmentation

Classical image augmentation techniques are still relevant and widely used in computer vision because they provide a simple yet effective way to increase the size and diversity of the training dataset. These techniques are easy to implement and computationally efficient, making them suitable for large-scale datasets and real-time applications.

2.1 Flipping and Rotating

Robust to changes in object orientation.

2.2 Cropping and Resizing

To handle changes in object scale or position within the image.

2.3 Color Jittering

To handle changes in lighting conditions and color variations in the dataset.

2.4 Adding Noise

Beneficial to handle variations in image quality and simulate noisy environments.

2.5 Image Warping

Helpful for increasing a model's ability to handle changes in object pose and simulate changes in camera viewpoint.

2.6 Random Erasing

Useful to make it more robust to occlusions or clutter in the image, or when you want to simulate missing data.

3 Advanced Techniques

3.1 Cutout

Cutout is an augmentation technique that randomly covers a region of an input image with a square.

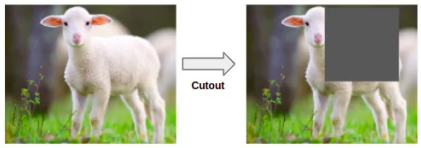

Figure 1: Cutout Illustration

Explanation Co-adaptation in neural networks refers to a situation where some neurons become highly dependent on others, affecting model performance when independent neurons receive "bad" inputs. Cutout applies a similar method on images for CNNs by dropping contiguous sections of inputs rather than individual pixels or neurons.

Advantages

- Helps in training models to recognize partial or occluded objects.
- Allows the model to consider more of the image context such as minor features rather than relying heavily on major features.

Limitations

- It can completely remove important features from an image.
- It may not work well for images with complex backgrounds.
- Significantly reduces the proportion of informative pixels used in the training process.

Hyperparameter Size and number of patches to be cut out from the image.

3.2 Mixup Augmentation

Mixup generates a weighted combination of random image pairs from the training data. Given two images and their ground truth labels: $(x_i, y_i), (x_j, y_j)$, a synthetic training example (x, y) is generated as:

$$x = \lambda x_i + (1 - \lambda) x_j$$

$$y = \lambda y_i + (1 - \lambda) y_j$$

where λ is sampled from the Beta distribution.

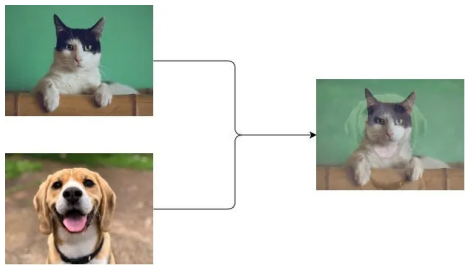

Figure 2: Mixup Illustration

Explanation Mixup constructs virtual training examples, extending the training distribution by incorporating the prior knowledge that linear interpolations of feature vectors should lead to linear interpolations of the associated targets.

Advantages

- Reduces overfitting by combining different features and labels.
- Provides a smoother estimate of uncertainty.
- Robustness to adversarial examples and stabilized GAN training.
- Domain-agnostic technique applicable to various data modalities.

Limitations

- Only inter-class mixup.
- The examples are not real representations of classes.
- Does not work well with Supervised Contrastive Learning and label smoothing.

3.3 Cutmix

Cutmix replaces a square region of an input image with a patch of similar dimensions from another image. The ground truth labels are mixed proportionally to the number of pixels from each image.

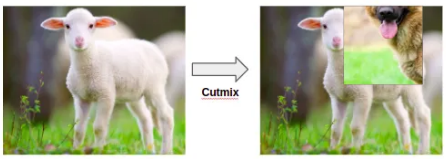

Figure 3: Cutmix Illustration

Explanation

$$x = M \odot x_i + (1 - M) \odot x_j$$

$$y = \lambda y_i + (1 - \lambda) y_j$$

where M is a binary mask indicating the Cutout and fill-in regions from the two randomly drawn images.

Advantages

- Avoids uninformative pixels during training.
- Efficient training process.

Limitations

- Complex to implement.
- May not work well with certain datasets.

3.4 Augmix

Augmix uses three separate chains of one to three randomly chosen augmentation operations, combined with the original image using different weights.

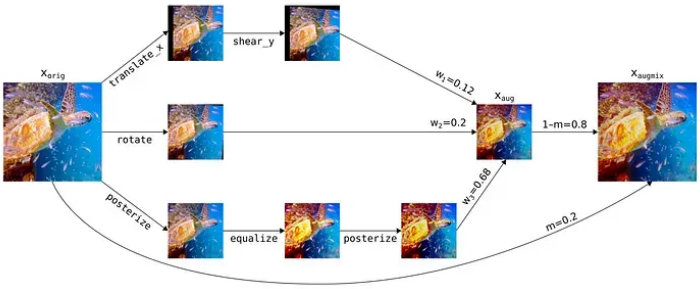

Figure 4: Augmix Illustration

Explanation Creates multiple sources of randomness, including the selection of operations, their intensity, the length of the chains, and the mixing weights. Combined with a consistency loss to ensure the augmented images are semantically meaningful.

Jensen-Shannon Consistency Loss

$$JS(p, q) = \frac{1}{2}\left(KL(p\|m) + KL(q\|m)\right)$$

$$L_{JS} = \lambda \cdot JS(p, q)$$

where $m = \frac{1}{2}(p + q)$ and λ is the weight of JSC in the loss.

Hyperparameters

- Number of augmented versions.
- Alpha: weight of JSC in loss.

4 Summary

Data augmentation significantly improves the performance of image classification and object detection models by increasing the variability of the training data and reducing overfitting. Advanced augmentations such as CutMix, MixUp, and AugMix generate more diverse and realistic training data, encouraging models to learn more robust features.

5 References

- Improved Regularization of Convolutional Neural Networks with Cutout
- CutMix: Regularization Strategy to Train Strong Classifiers with Localizable Features

- Mixup: Beyond Empirical Risk Minimization
- AugMix: A Simple Data Processing Method to Improve Robustness and Uncertainty
- A Comprehensive Survey of Image Augmentation Techniques for Deep Learning
- https://keras.io/examples/vision/cutmix/
- https://albumentations.ai