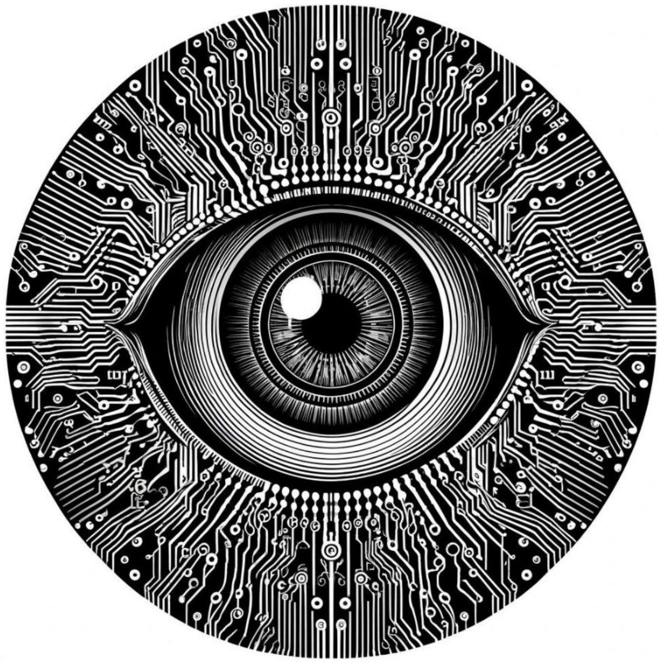
Image Data Augmentation



Antonio Rueda-Toicen

SPONSORED BY THE

Federal Ministry
of Education
and Research

Learning goals

- Use data augmentation to extend training datasets
- Gain familiarity with PyTorch’s transforms.v2: affine transformations, vertical and horizontal flipping, and random crops

Data augmentation

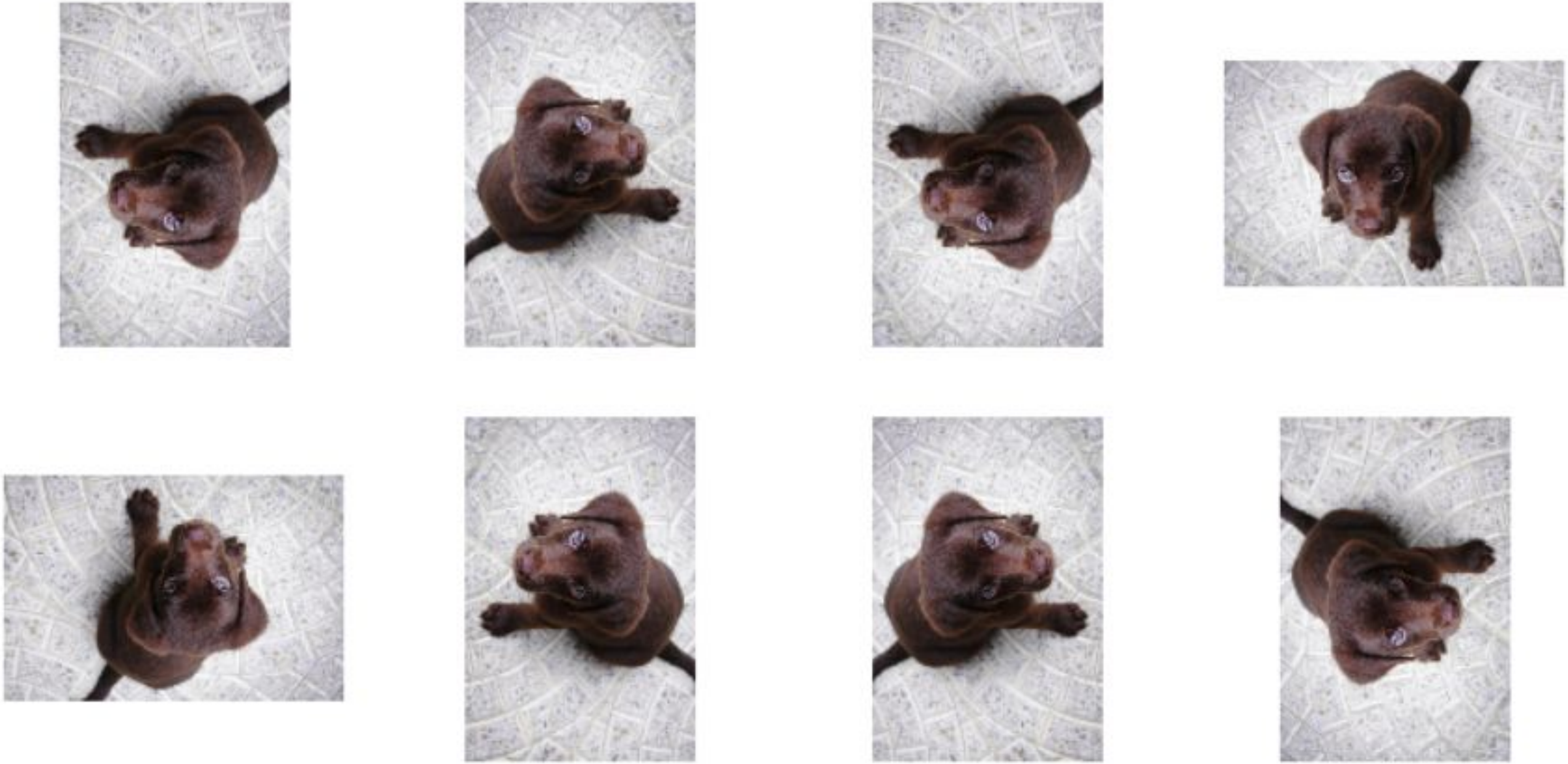
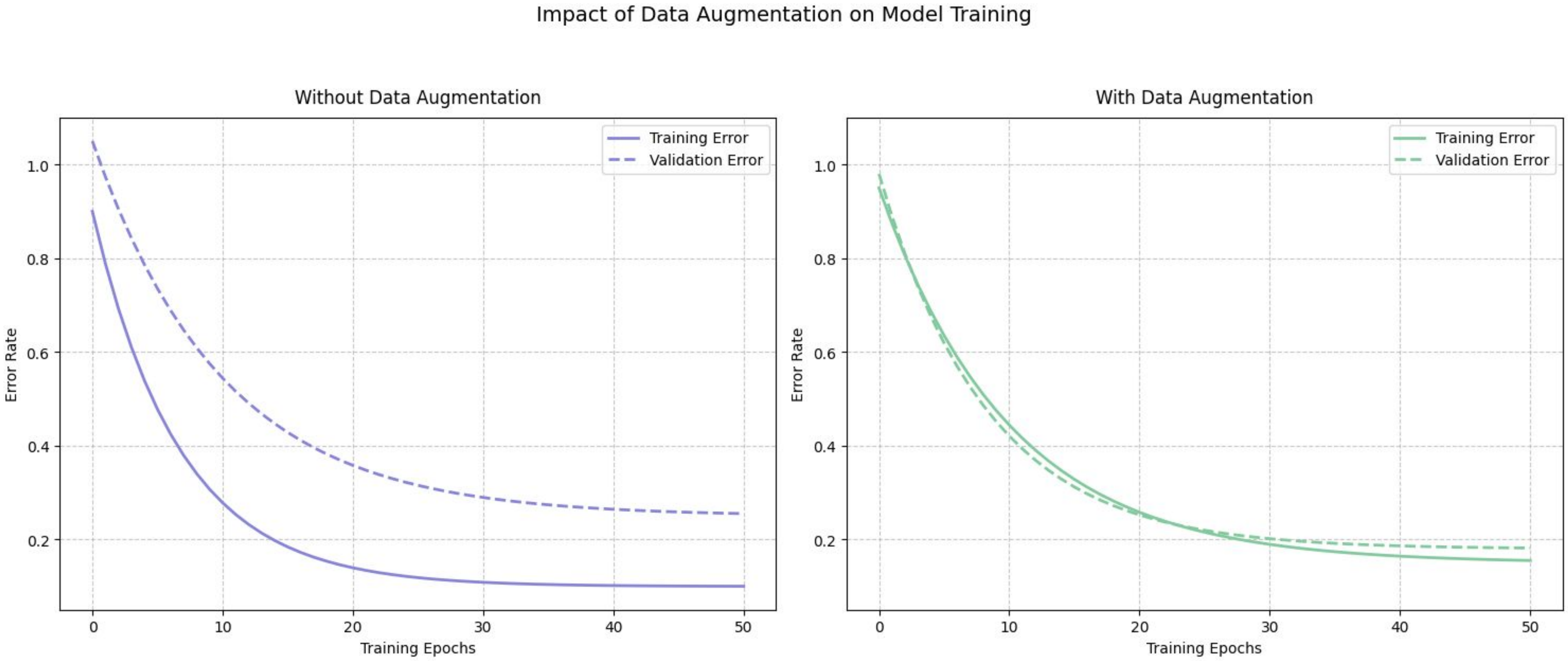


Image from https://docs.fast.ai/vision.augment

Data augmentation as regularization

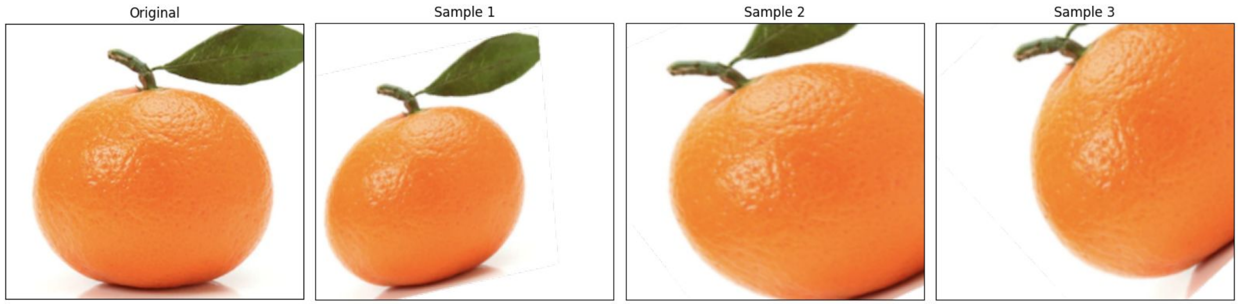


Affine transformations

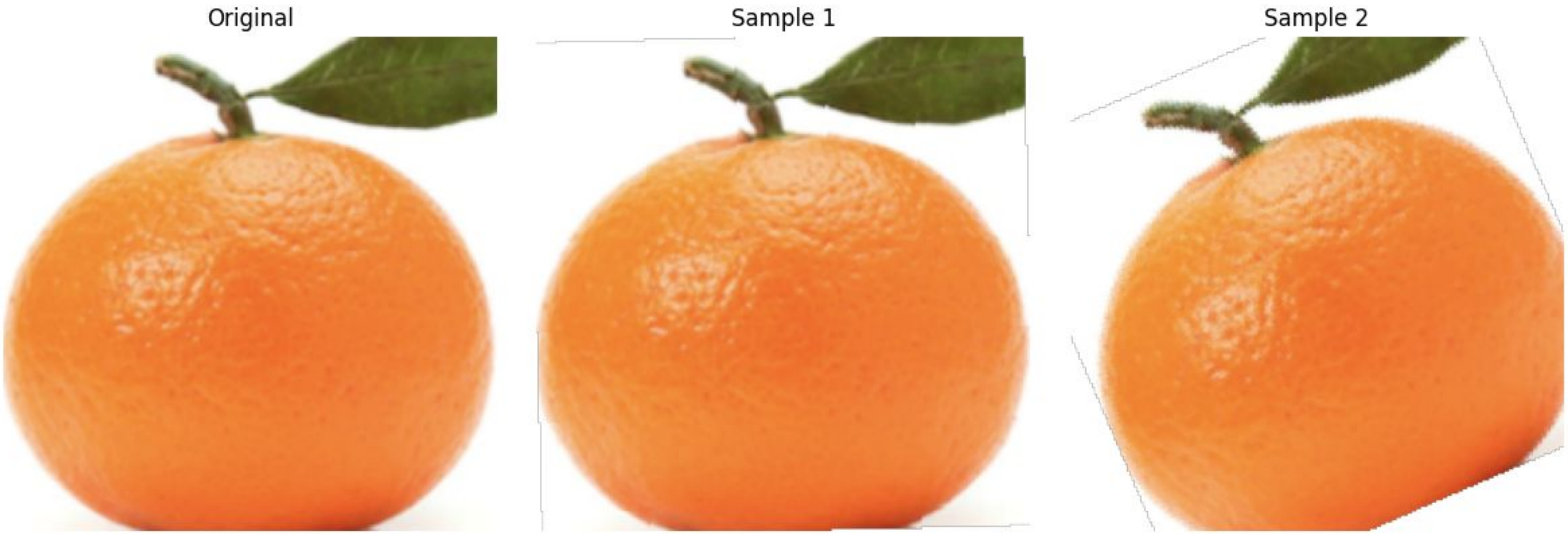
$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} \Delta_x \\ \Delta_y \end{pmatrix}$$

Linear transformation
(Rotation, scaling, shear)

Translation
(Moving along the axes)



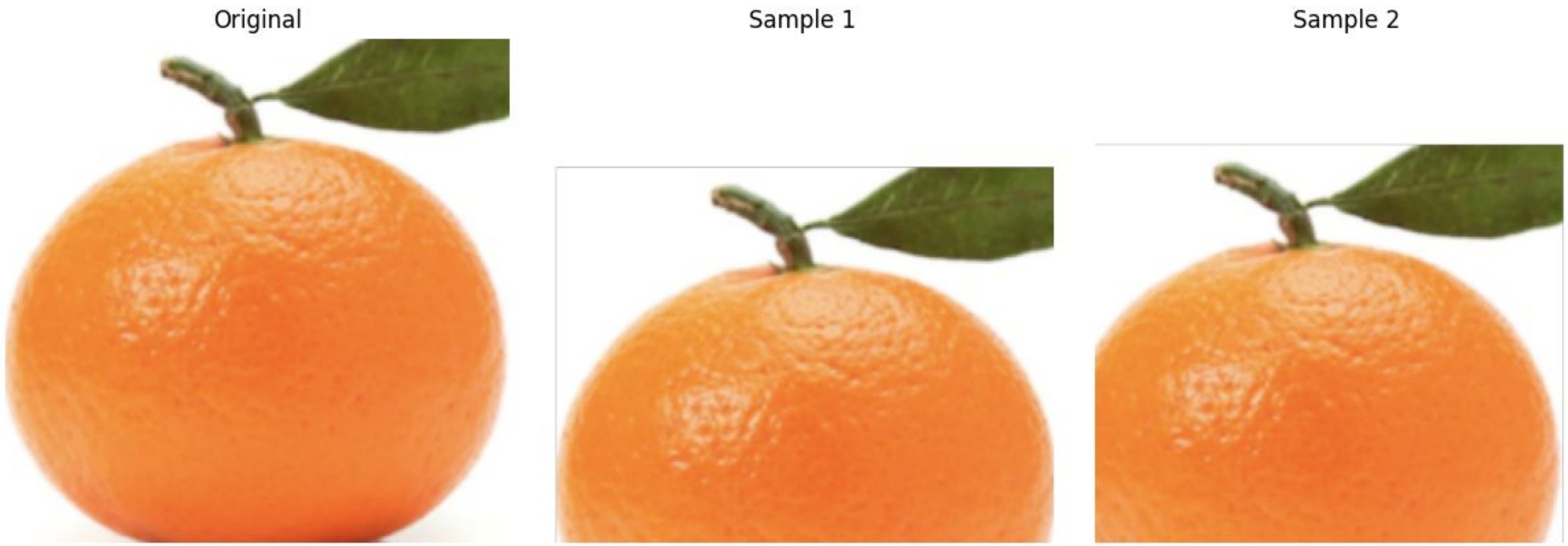
The RandomAffine transformation - rotation



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

```
transforms.RandomAffine(degrees=45)
```

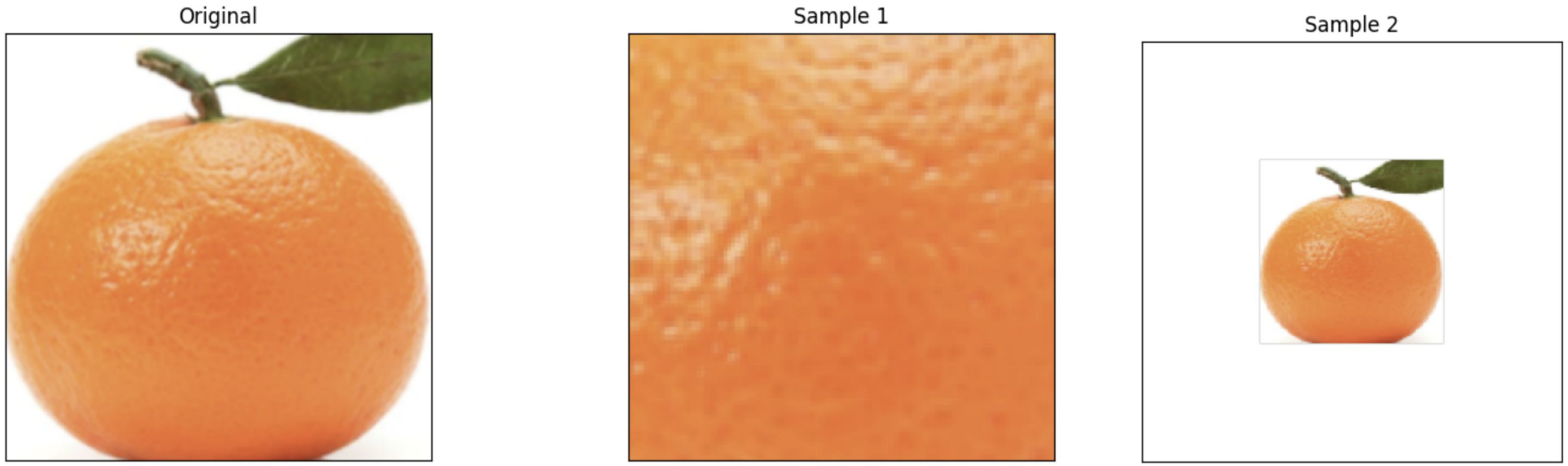
The RandomAffine transformation - translation



$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix}$$

```
transforms.RandomAffine(degrees=0, translate=(0.4, 0.4))
```

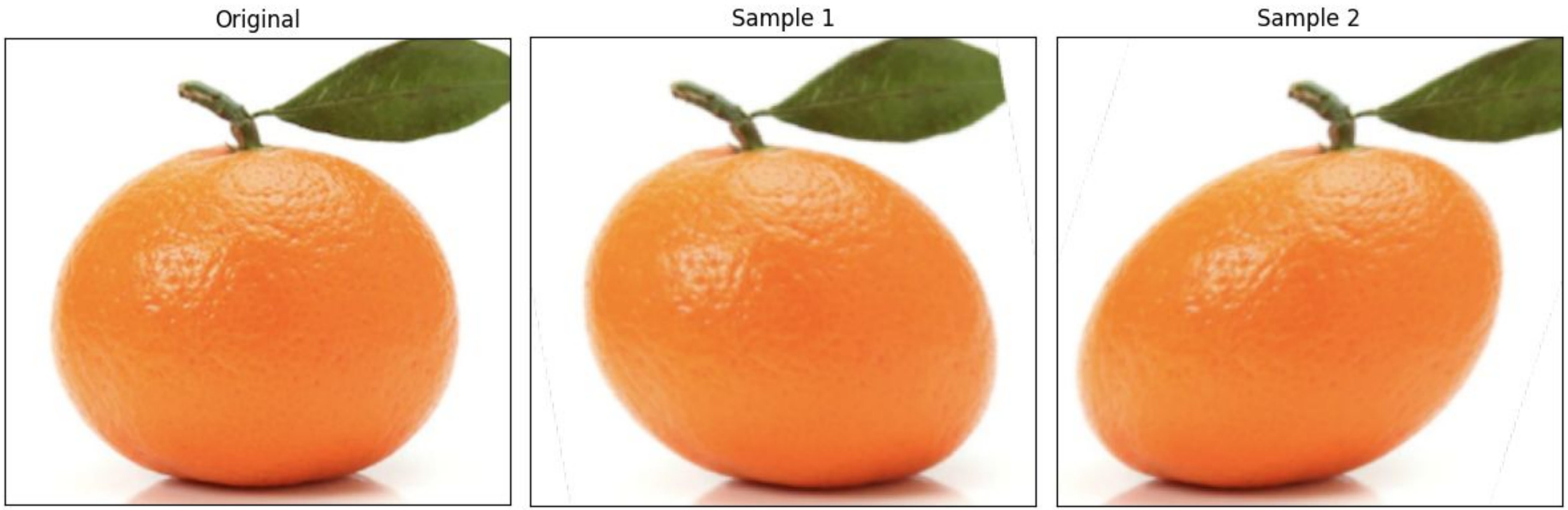
The RandomAffine transformation - scaling



$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} s_x & 0 \\ 0 & s_y \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

```
transforms.RandomAffine(degrees=0, scale=(0.25, 2.5))
```

The RandomAffine transformation - shear



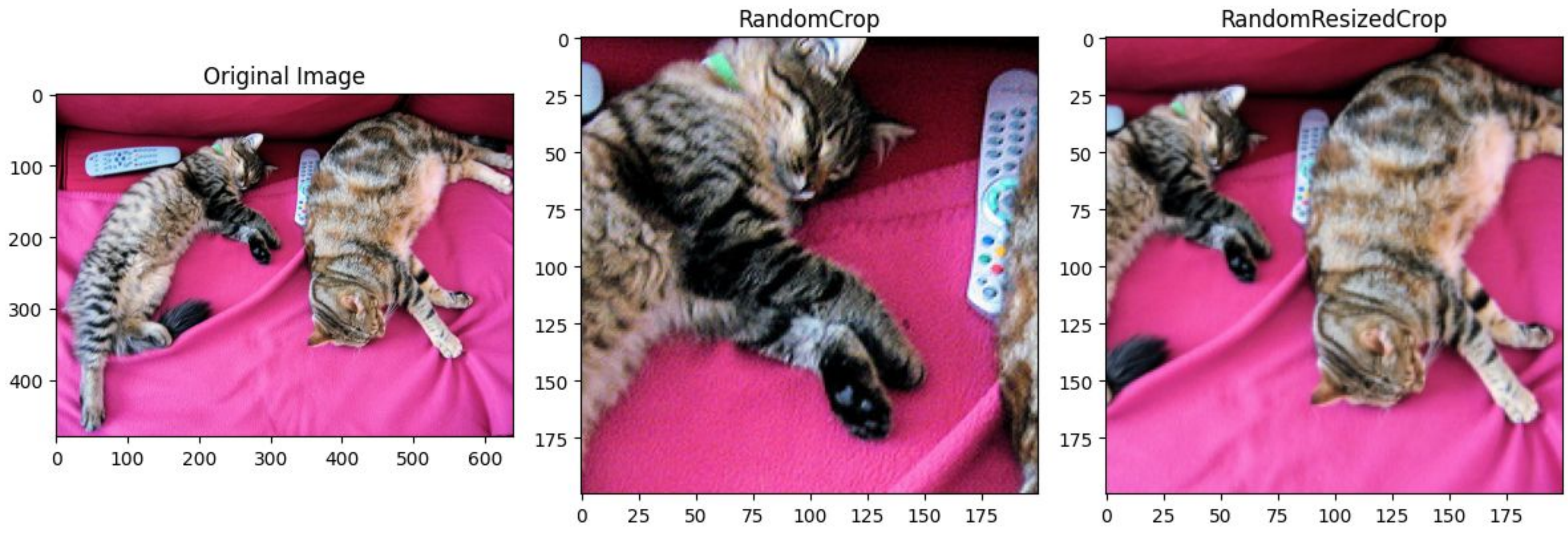
$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} 1 & s_x \\ s_y & 1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

```
transforms.RandomAffine(degrees=0,shear=45)
```

Horizontal and vertical flipping


```
transforms.Compose([
        transforms.RandomHorizontalFlip(p=0.5),
        transforms.RandomVerticalFlip(p=0.5)
   ])
```

RandomCrop and RandomResizedCrop




```python
transforms.RandomCrop(size=224)
```

```python
transforms.RandomResizedCrop(
        size=224,
        scale=(0.08, 1.0),   # Range of size of crop
        ratio=(3/4, 4/3)     # Range of aspect ratio
    )
```

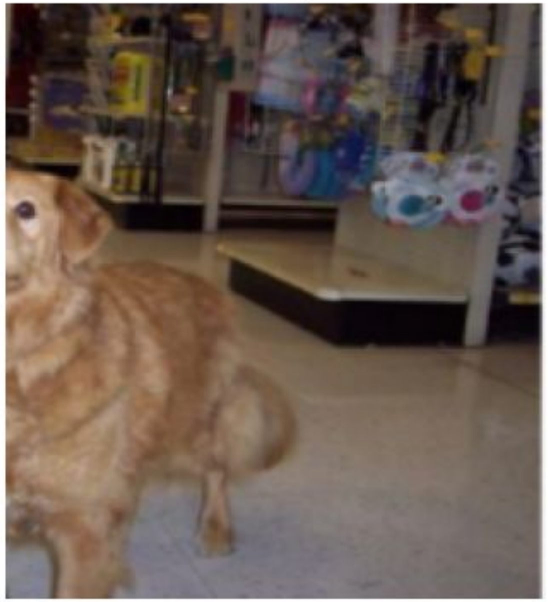
Blind data augmentation can damage your model

$\hat{y} =$ golden retriever (unit 207 on Imagenet)

Original

Sample 1

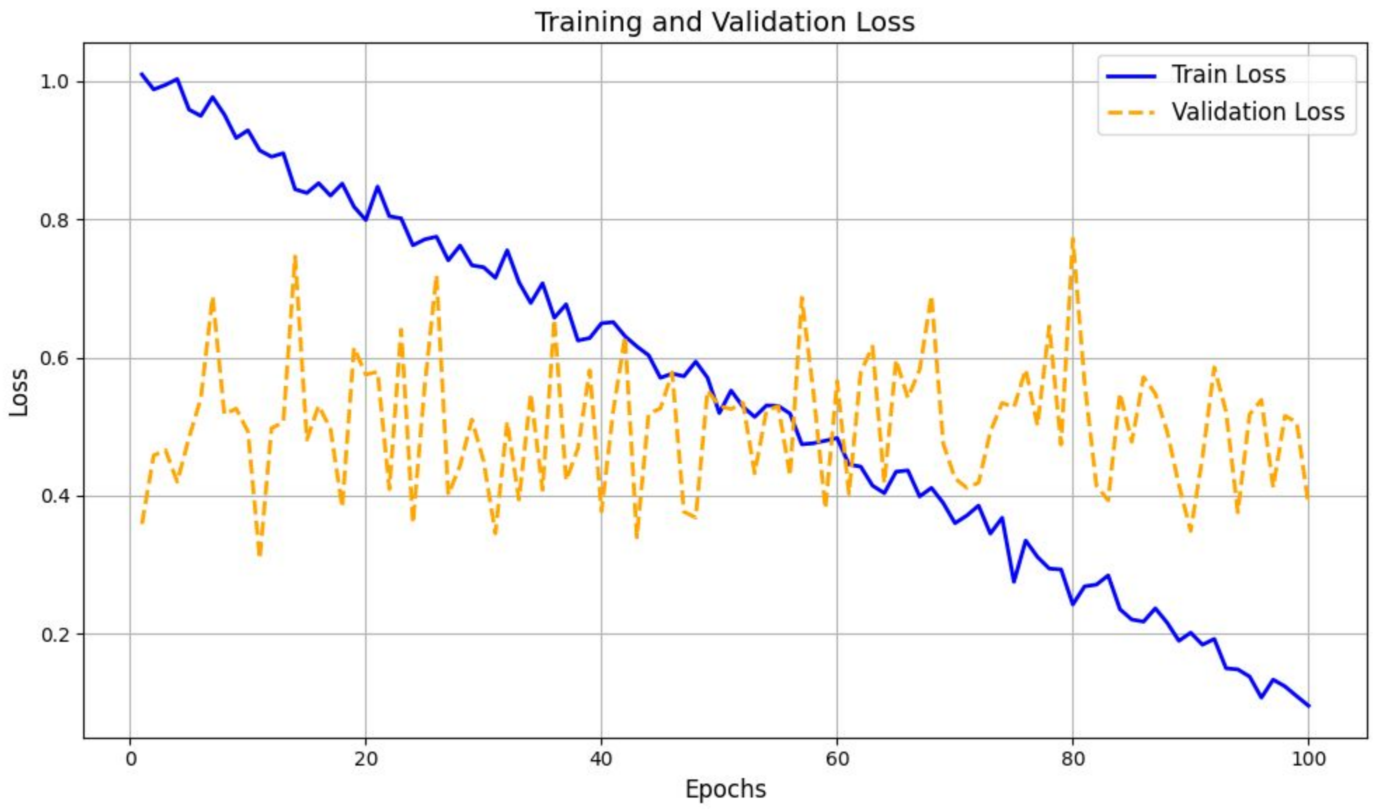
Sample 2



$$\text{Cross Entropy Loss} = -\log(0.003) = 5.809$$

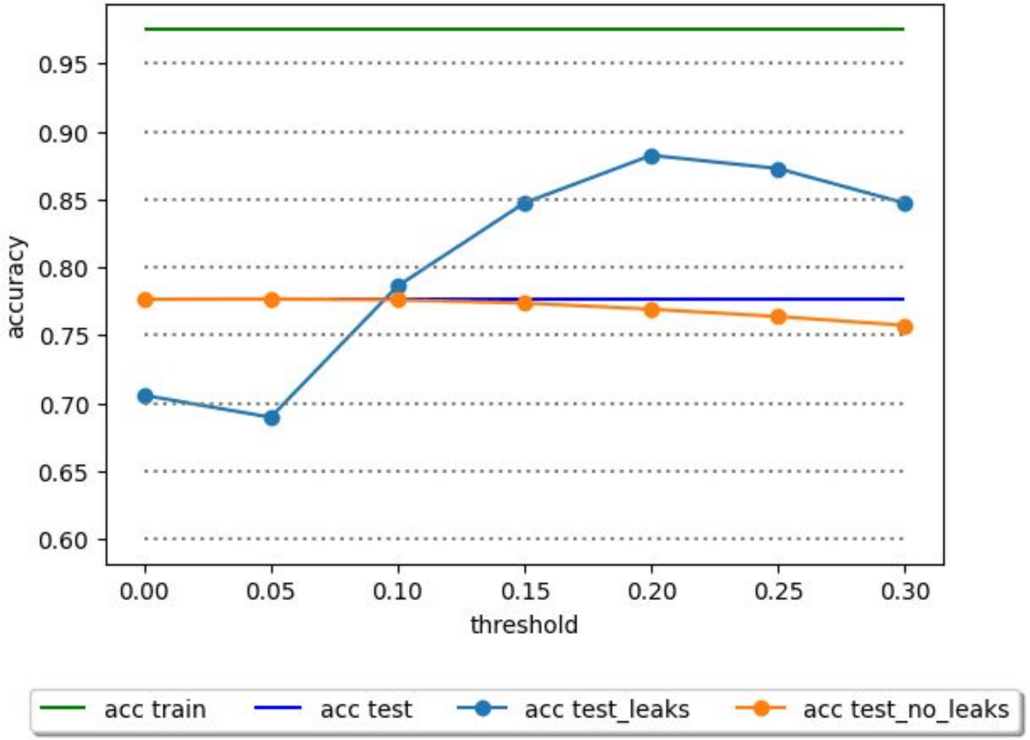
$$-\log(0.6) = 0.511$$

The validation set should not be augmented during training




```python
import torchvision.transforms.v2 as T

imagenet_mean = [0.485, 0.456, 0.406]
imagenet_std  = [0.229, 0.224, 0.225]

transform = T.Compose([

# Resize to 224x224
T.Resize((224, 224)),

# Convert to torch Image
T.ToImage(),

# Convert to scaled float tensor
T.ToDtype(torch.float32, scale=True),

# Apply ImageNet normalization
T.Normalize(mean=imagenet_mean,
            std=imagenet_std)
])
```

Augmentation should happen <u>after</u> train, validation, and test splitting



Images from "[On Leaky Datasets and a Clever Horse]"

Summary

Data augmentation creates synthetic input data

- Data augmentation can improve the robustness of a model and prevent overfitting

PyTorch transforms are powerful and composable

- `transforms.v2` provides rotation, scale, crop, and flip (+other) operations
- Each transform has specific parameters for fine control

Avoiding data leakage: best practices

- Apply augmentations after train, validation, and test split. Never augment validation data. Test augmentation effects through observation and loss values

SPONSORED BY THE
Federal Ministry of Education and Research

Further reading and references

Data augmentation is still data curation

- https://voxel51.com/blog/data-augmentation-is-still-data-curation/

Getting started with PyTorch's transforms v2

- https://pytorch.org/vision/main/auto_examples/transforms/plot_transforms_getting_started.html

Illustration of transforms

- https://pytorch.org/vision/main/auto_examples/transforms/plot_transforms_illustrations.html#sphx-glr-auto-examples-transforms-plot-transforms-illustrations-py

SPONSORED BY THE
Federal Ministry of Education and Research