Dataset Augmentaion

This article will cover how you can increase the size of your original dataset with the help of data augmentation. Data augmentaion is a practice of altering samples in your dataset, making them distinct enough from the original sample to be considered a new sample, and keeping alterations small enough to keep them recognizable as a part of the dataset's original data domain.
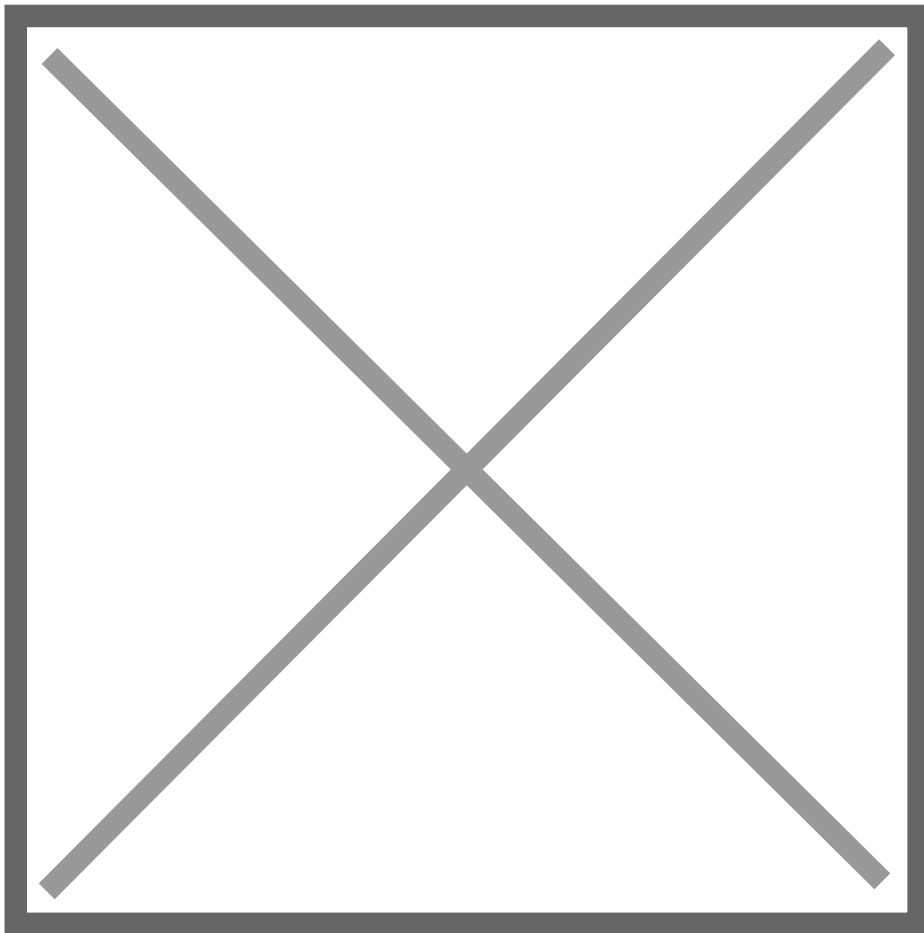**Examples:** Adding slight noise to audio samples and mirroring images.

Image augmentaion

The simplest way to add data augmentaion to your training pipeline is to use <u>Albumentations</u> library.
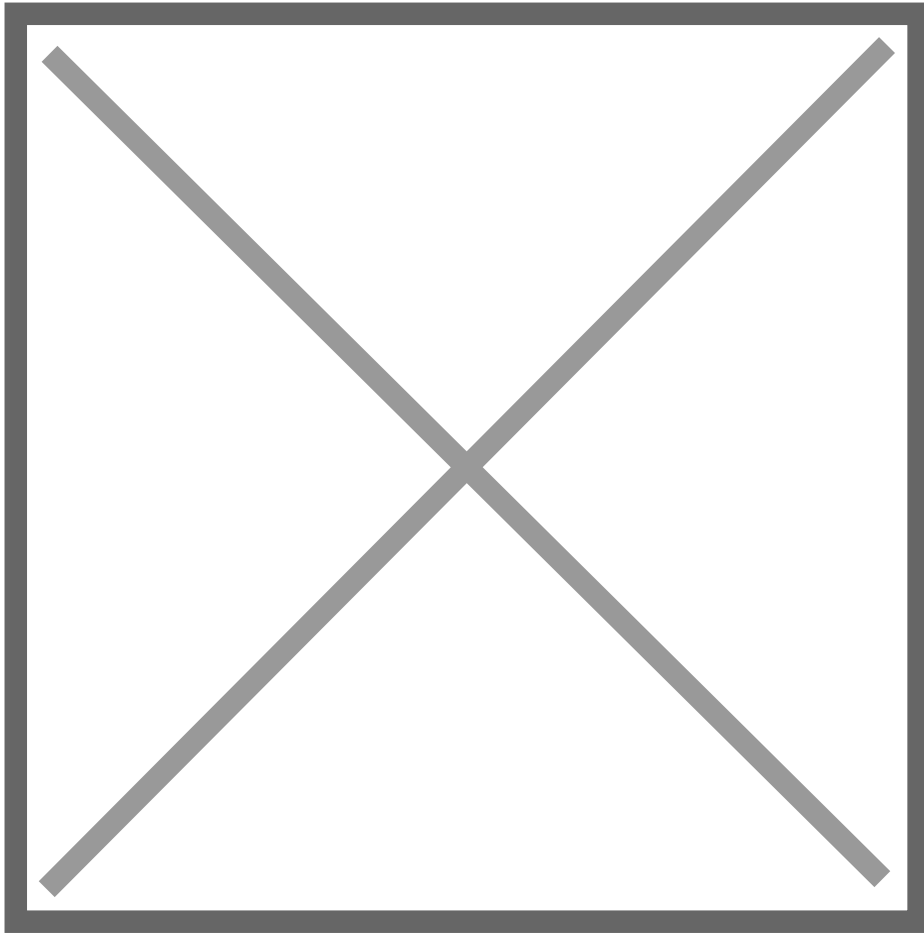
Starting from the most basic ones, here are some augmentaion tricks you can use:
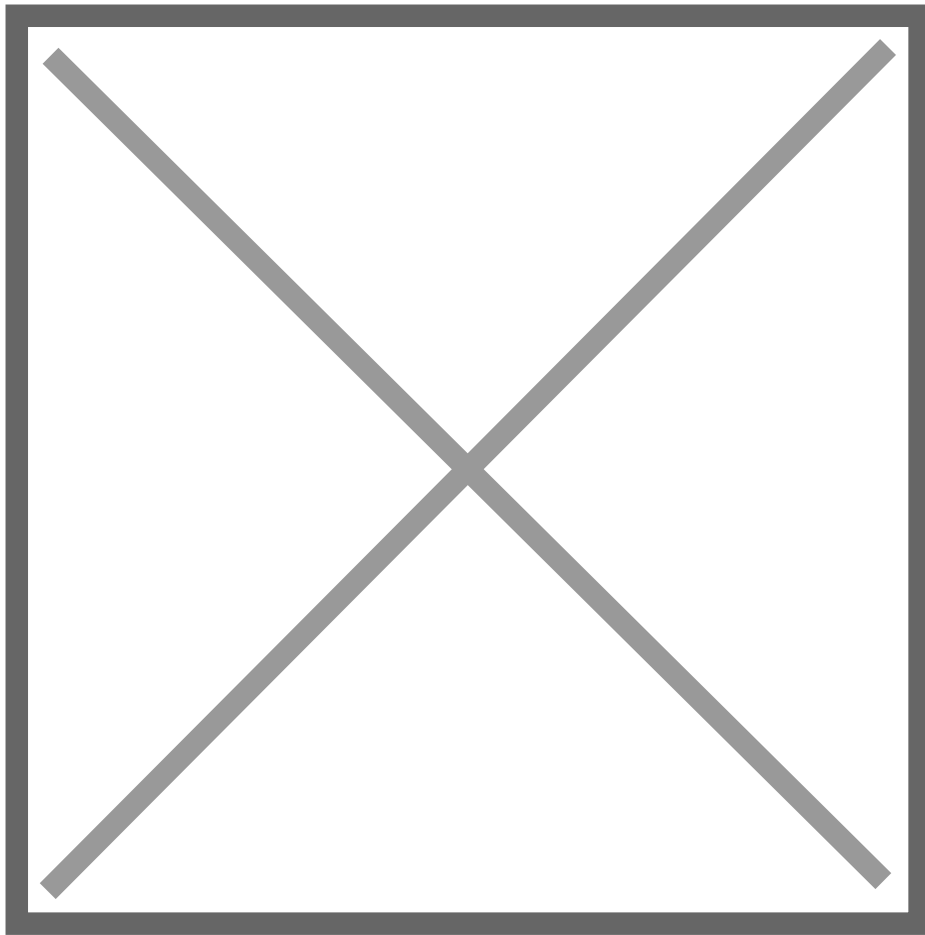
- Original image:


```
image = Image.open("testImg.jpg")
image_np = np.array(image)
```
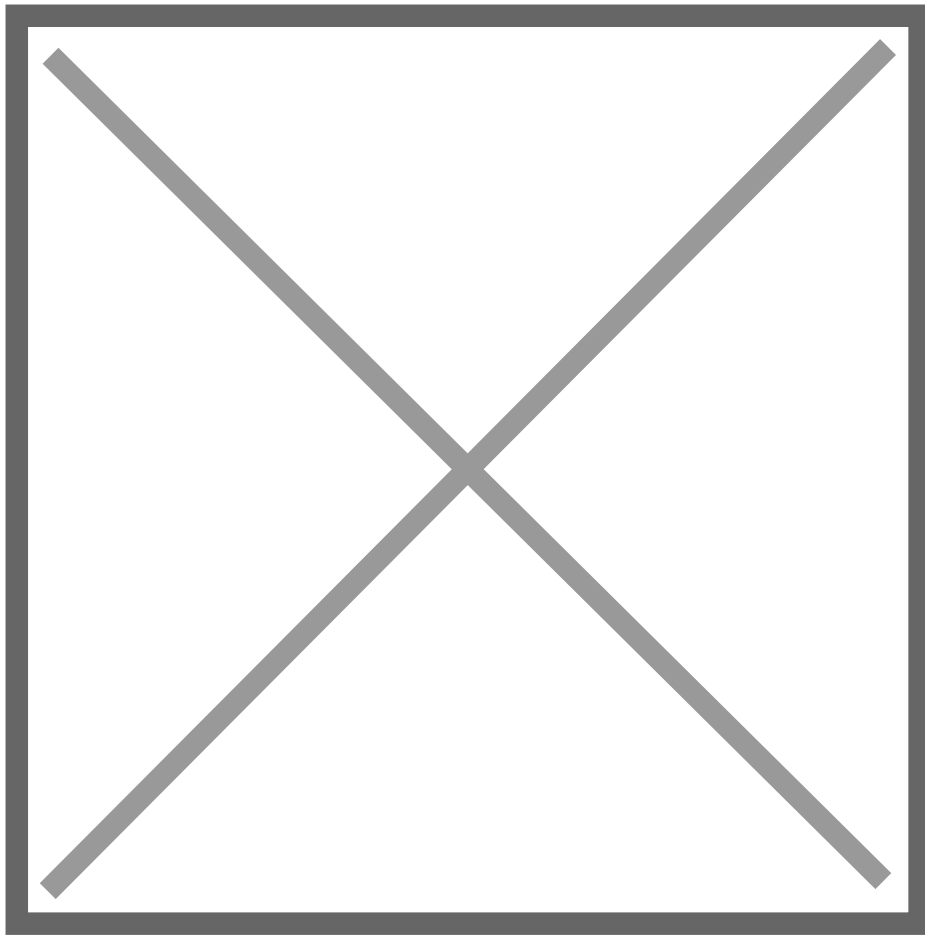
- Image flipping or mirroring:


```
transform = A.Compose([A.HorizontalFlip(p=1.0)])
transformed_image = transform(image=image_np)["image"]
```
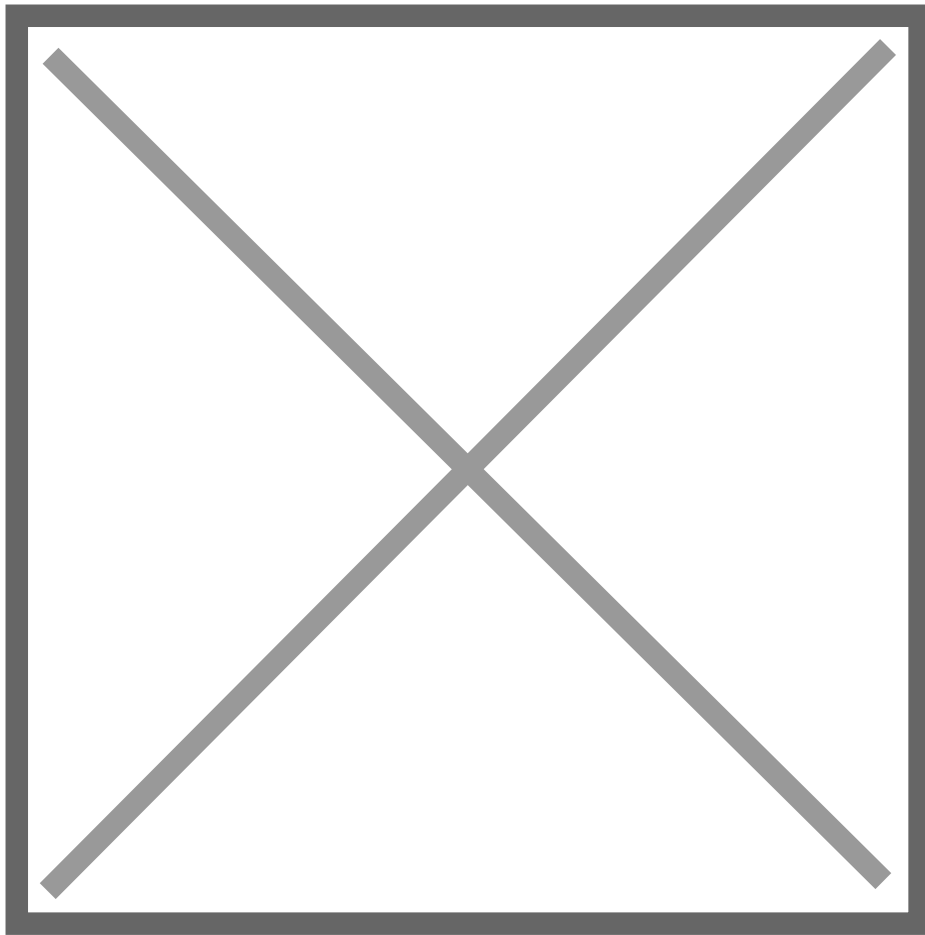
```
transform = A.Compose([A.VerticalFlip(p=1.0)])
transformed_image = transform(image=image_np)["image"]
```
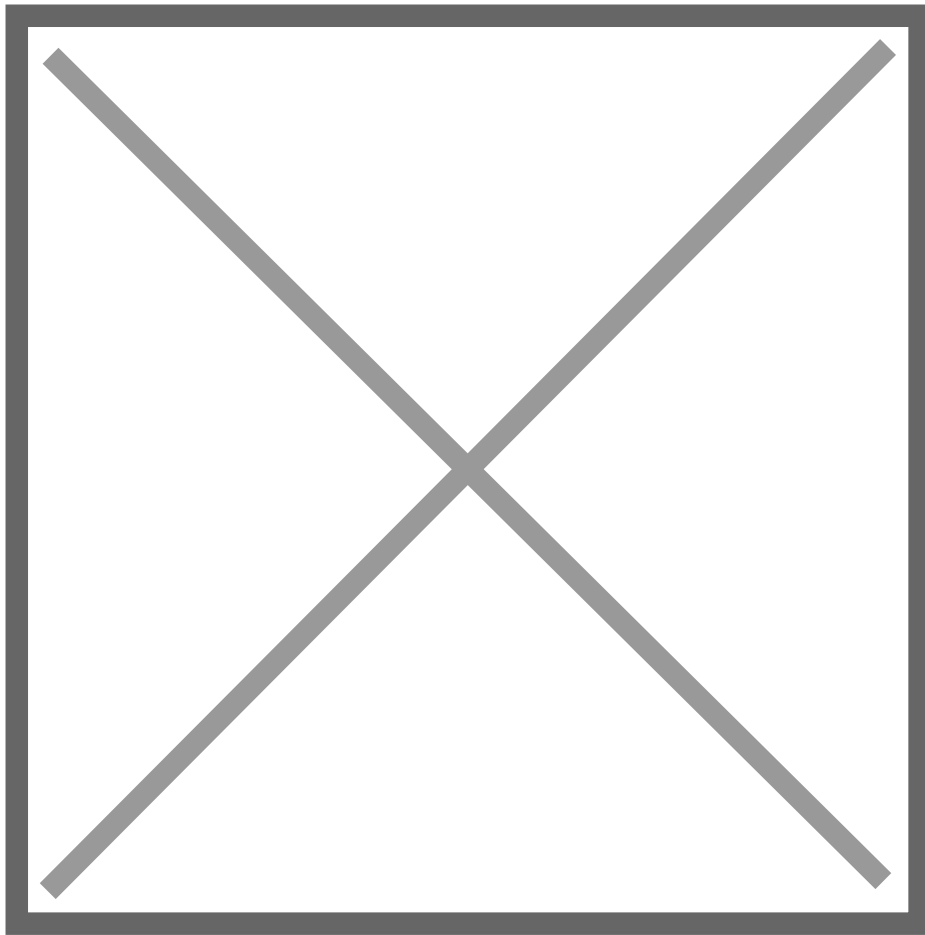
- Image rotation:

```
transform = A.Compose([A.Rotate(p=1.0, limit=45, border_mode=0)])
transformed_image = transform(image=image_np)["image"]
```
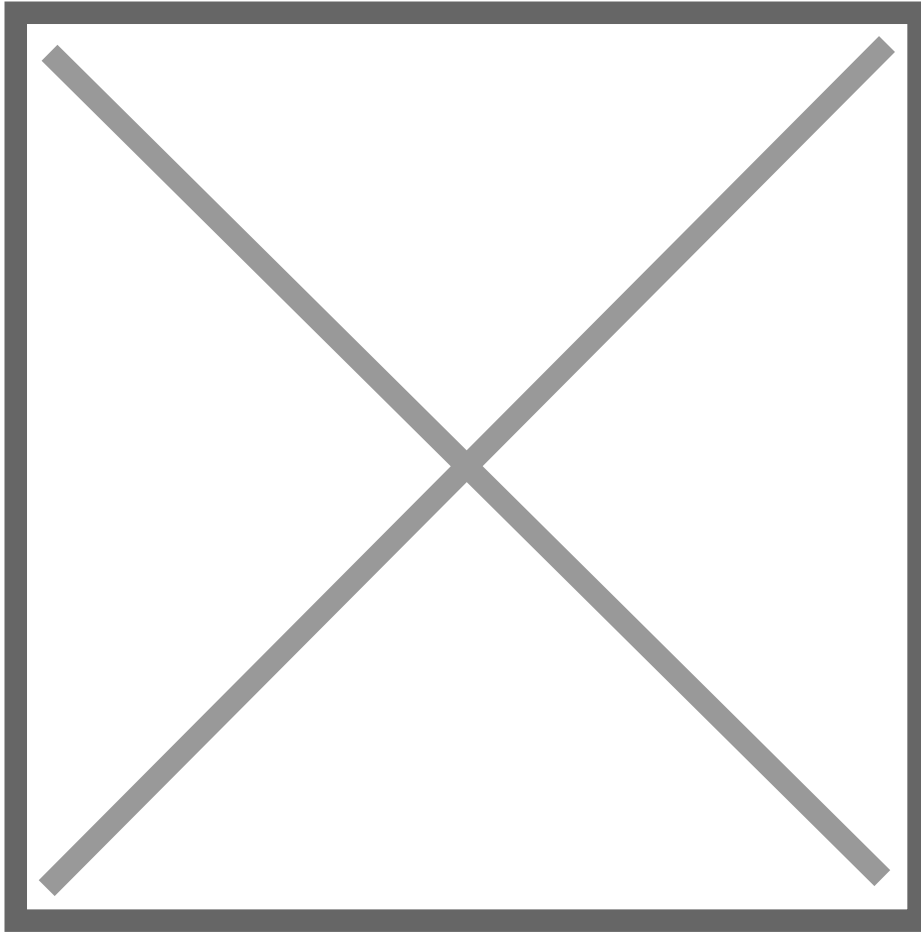
- HSV Jitter:

```
transform = A.Compose([A.ColorJitter(p=1.0)])
transformed_image = transform(image=image_np)["image"]
```
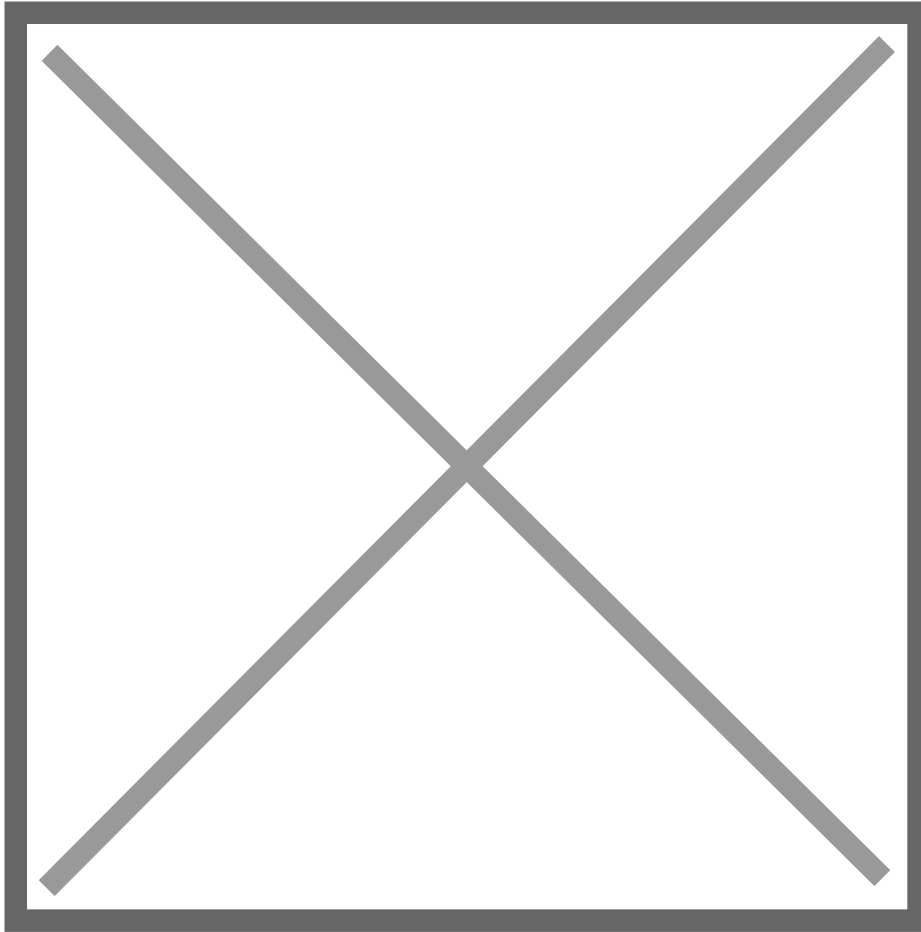
- Gaussian Noise:

```
transform = A.Compose([A.GaussNoise(p=1.0, var_limit=(1000.0, 5000.0))])
transformed_image = transform(image=image_np)["image"]
```

Augmentaions almost always combined with each other:

```
transform = A.Compose([
    A.HorizontalFlip(p=1.0),
    A.VerticalFlip(p=1.0),
    A.Rotate(p=1.0, limit=45, border_mode=0),
    A.RandomBrightnessContrast(p=1.0, brightness_limit=(0.15,0.25)),
    A.ColorJitter(p=1.0),
    A.GaussNoise(p=1.0, var_limit=(1000.0, 2000.0),),
])
transformed_image = transform(image=image_np)["image"]
```

Above is an extremecase of image augmentaion, we still want to keep the resulting images as close to the original data distribution as possible:

```
transform = A.Compose([

    A.HorizontalFlip(p=0.5),

    A.Rotate(p=0.5, limit=15, border_mode=0),

    A.RandomBrightnessContrast(p=0.5, brightness_limit=(-0.1,0.1)),

    A.ColorJitter(p=0.5),

    A.GaussNoise(p=0.5, var_limit=(50.0, 250.0),),

])

transformed_image = transform(image=image_np)["image"]
```

Further Reading

You can follow the links bellow for example use of Albumentaions library with popular AI/ML libraries.

- [Tensorflow](#).
- [PyTorch](#).

Revision #5
Created 15 April 2024 14:27:19 by Ilia Pavlov
Updated 29 May 2024 10:04:53 by Ilia Pavlov