```
from fastai.vision.all import *
```

```
path = untar_data(URLs.IMAGENETTE_320)
```

```
img = array(Image.open((path/'train').ls()[0].ls()[0]))
```

```
show_image(img),img.shape
```

```
(<AxesSubplot:>, (320, 463, 3))
```

Schedule: Question, This notebook, How to read a research paper, Presentation/Less formal

```
np.random.choice(np.arange(14),8,replace=False)
```

```
array([2, 5, 1, 0, 9, 8, 6, 7])
```

```
img.shape
```

```
(320, 463, 3)
```

```
img.shape[0]//2,img.shape[1]//2
```

```
(160, 231)
```

```
ex=img[:img.shape[0]//2,:img.shape[1]//2]
show_image(ex)
```

```
<AxesSubplot:>
```

```
img.shape[0]//4,-img.shape[0]//4
```

```
(80, -80)
```

```
ex=img[img.shape[0]//4:-img.shape[0]//4,
       img.shape[1]//4:-img.shape[1]//4]
show_image(ex)
```

```
<AxesSubplot:>
```

```
ex.shape,img.shape
```

```
((160, 232, 3), (320, 463, 3))
```

```
show_image(img)
```

```
<AxesSubplot:>
```

```
list(range(10,0,-1))
```

```
[10, 9, 8, 7, 6, 5, 4, 3, 2, 1]
```

```
show_image(img),img.shape
```

```
(<AxesSubplot:>, (320, 463, 3))
```

```
show_image(img[:-1:2,:-1:2]),img[:-1:2,:-1:2].shape
```

```
(<AxesSubplot:>, (160, 231, 3))
```

```
show_image(img[1::2,1::2]),img[1::2,1::2].shape
```

```
(<AxesSubplot:>, (160, 231, 3))
```

```
img[:-1:2,:-1:2]//2+img[1::2,1::2]//2
```

```
img[::2,::2]//2+img[1::2,1::2]//2
```

```
ValueError: operands could not be broadcast together with shapes (160,232,3) (160,231,3)
```

```
img[1::2,1::2].shape
```

```
(160, 231, 3)
```

```
ex= img[:-1:2,:-1:2]//2+img[1::2,1::2]//2
show_image(ex),ex.shape
```

```
(<AxesSubplot:>, (160, 231, 3))
```

```
array([1/3,1/3,1/3])
```

```
array([0.33333333, 0.33333333, 0.33333333])
```

```
(array([1/3,1/3,1/3])@img[...,None]).shape
```

```
(320, 463, 1)
```

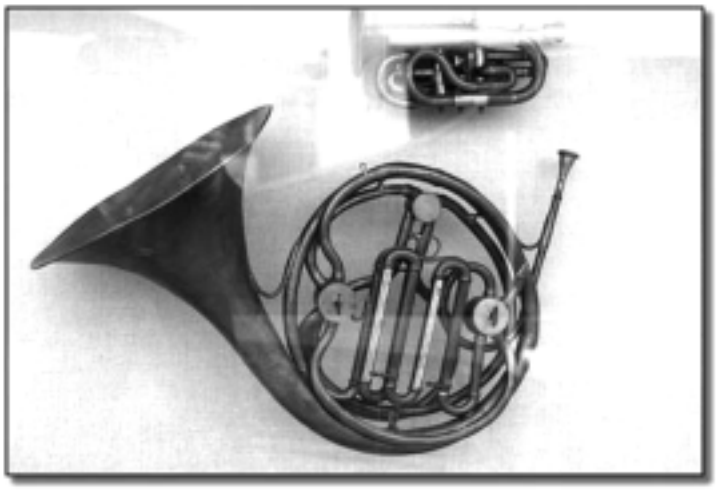
```
ex1.max(),ex2.max()
```

```
(765, 255.0)
```

```
np.
```

```
ex1= (array([1/2,1/2,1/2])@img[...,None]).clip(0,255)
ex2= (array([1/3,1/3,1/3])@img[...,None]).clip(0,255)
show_image(ex1,cmap='gray'),show_image(ex2,cmap='gray')
```

```
(<AxesSubplot:>, <AxesSubplot:>)
```



```
norm_tfm=Normalize.from_stats(*imagenet_stats,cuda=False)
def show_norm(img): show_images((norm_tfm.decode(img).clamp(0,1)),nrows=3)
```

```
norm_img = norm_tfm(TensorImage(img.transpose(2,0,1)).float()[None]/255)
```

```
noise= torch.randn_like(norm_img)
```

```
As = torch.linspace(0,1,12)[...,None,None,None]; As.squeeze()
```

```
tensor([0.0000, 0.0909, 0.1818, 0.2727, 0.3636, 0.4545, 0.5455, 0.6364, 0.7273,
        0.8182, 0.9091, 1.0000])
```

```
(As)**.5*norm_img
```

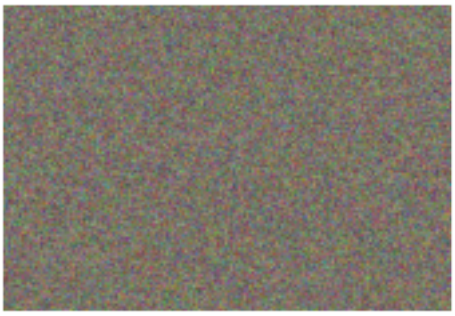
```
(1-As**.5).squeeze()
```

```
tensor([1.0000, 0.6985, 0.5736, 0.4778, 0.3970, 0.3258, 0.2615, 0.2023, 0.1472,
        0.0955, 0.0465, 0.0000])
```

```
((1-As)**.5).squeeze()
```

```
tensor([1.0000, 0.9535, 0.9045, 0.8528, 0.7977, 0.7385, 0.6742, 0.6030, 0.5222,
        0.4264, 0.3015, 0.0000])
```

```
show_norm((As)**.5*norm_img+(1-As)**.5*noise)
```



```
show_norm((As)**.5*norm_img+(1-As**.5)*noise)
```

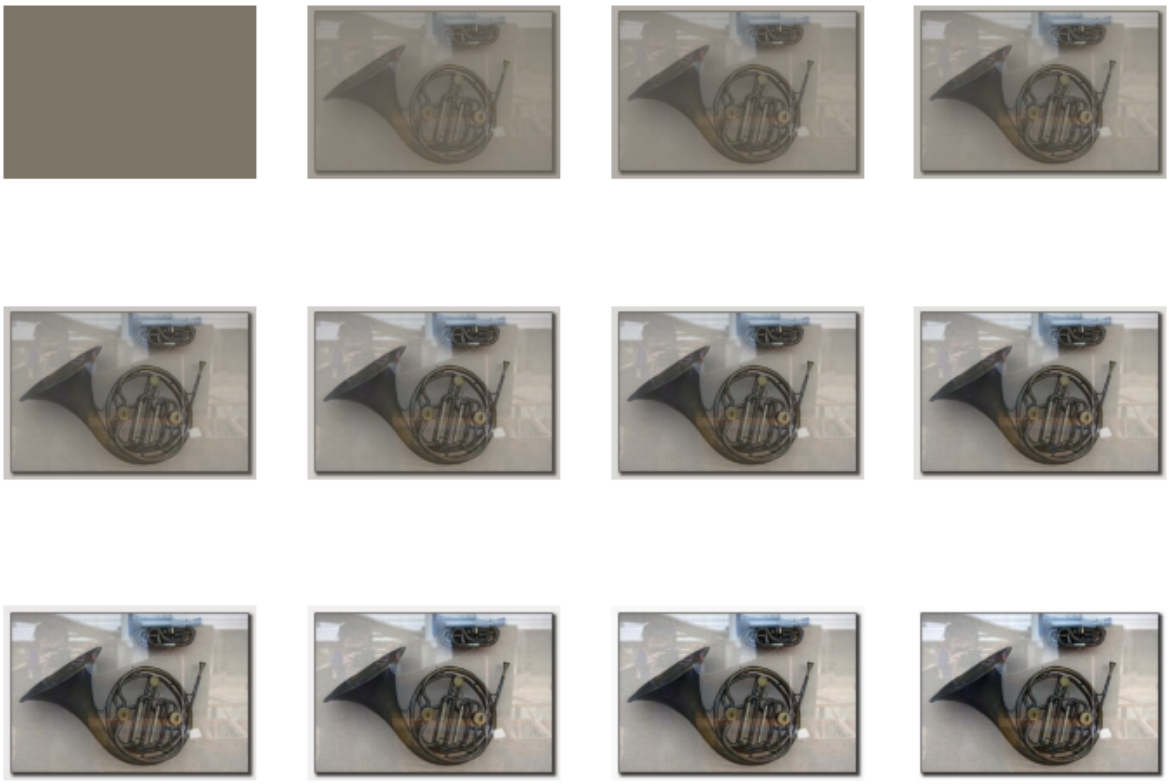
```
As.squeeze(),As.shape
```

```
(tensor([0.0000, 0.0909, 0.1818, 0.2727, 0.3636, 0.4545, 0.5455, 0.6364, 0.7273,
         0.8182, 0.9091, 1.0000]),
 torch.Size([12, 1, 1, 1]))
```

```
norm_img.shape
```

```
torch.Size([1, 3, 320, 463])
```

```
show_norm((As)**.5*norm_img)
```



```
1-(As)**.5
```

```
tensor([[[[1.0000]]],

        [[[0.6985]]],

        [[[0.5736]]],

        [[[0.4778]]],

        [[[0.3970]]],

        [[[0.3258]]],
```

```
      [[[0.2615]]],

      [[[0.2023]]],

      [[[0.1472]]],

      [[[0.0955]]],

      [[[0.0465]]],

      [[[0.0000]]]])
```

```
show_norm((1-As)**.5*noise)
```