

Part 1

Model used: resnet 18

Images used: 250

Transform used: `transform = transforms.Compose([
 transforms.Resize(224),
 transforms.CenterCrop(224),
 transforms.ToTensor(),
 transforms.Normalize(mean=[0.485, 0.456, 0.406], std=[0.229, 0.224, 0.225])])`

The squared sum difference between the tensors with bilinear resize is `tensor(1.6001e-09)`. This may be approximated to zero if we consider the float rounding difference.

However, the squared sum difference between the first model and the `rescale_by_hand` model with out bilinear resize, the difference becomes `tensor(2846.6577)`, which is a huge difference.

After that, we compared the difference between a normalized dataset and an unnormalized one. The accuracy on the normalized dataset is 0.688, while the accuracy on the unnormalized dataset is 0.464. That is a huge difference of 0.224! This shows us how important it can be to normalize the dataset.

Part 2

Model used: resnet 18

Images used: 250

Transform used: `transform = transforms.Compose([
 transforms.Resize(280),
 transforms.FiveCrop(224),
 transforms.Lambda(lambda crops: torch.stack([transforms.ToTensor()(crop) for crop in crops])),
 transforms.Lambda(lambda crops: torch.stack([transforms.Normalize(mean=[0.485, 0.456, 0.406], std=[0.229, 0.224, 0.225])(crop) for crop in crops]))])`

The accuracy of the five-crop prediction is 0.704, which is higher than the center crop's 0.688. This shows that we should perform five-crop over center crop whenever possible, and the additional crops helps to increase the performance of the model.

However, `tencrop` is not always ideal, because sometimes the meaning of the image would be changed when the image is flipped, for example texts, numbers, and direction signs. In such cases, we should avoid `tencrop` or any other augmentation that involves flipping/mirroring of images.

Part 3

Models used: resnet18 and mobilenet_v2

Images used: 50

Transform used: `transform = transforms.Compose([
 transforms.Resize(400),
 transforms.FiveCrop(330),
 transforms.Lambda(lambda crops: torch.stack([transforms.ToTensor()(crop) for crop in crops])),
 transforms.Lambda(lambda crops: torch.stack([transforms.Normalize(mean=[0.485, 0.456, 0.406],
std=[0.229, 0.224, 0.225])(crop) for crop in crops]))])`

The resnet18 achieved accuracy of 0.72 while the mobilenet_v2 achieved accuracy of 0.64. It should be noted that the resnet18 achieved better result than in Part 1 and 2, showing that (I think) a larger crop is preferable and would improve the accuracy.

I am surprised that the resnet18 would be so much better than the mobilenet_v2. There may be other factors affecting the result too.