Report on Bagnet test

1.Imagenet test

Imagenet pretrained model tested on ILSVRC2012 \_val:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Test accuracy | | Time used | Reference accuracy | |
| Top1 | Top5 | Top1 | Top5 |
| Resnet-50 | 75.70% | 92.77% | 30m17s | 76.15% | 92.87% |
| VGG-16 | 71.52% | 90.46% | 23m16s | 71.59% | 90.38% |
| Bagnet-9 | 45.78% | 69.73% | 32m52s |  |  |
| Bagnet-17 | 58.52% | 81.07% | 37m14s |  |  |
| Bagnet-33 | 67.07% | 87.30% | 32m52s |  |  |

(Reference accuracy provided by torchvision.models docs.)

It seems that Bagnet has no advantage in terms of top1 accuracy but it improves quickly with the increase of patch size. Bagnet-33’s top5 accuracy can be compared with other nets in a way.

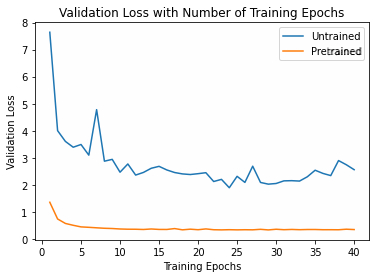
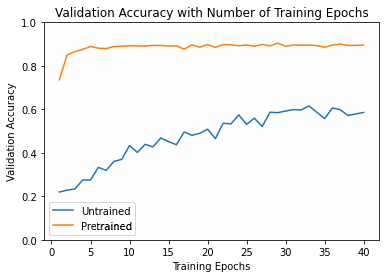
2.caltech101 test

I use both Imagenet pretrained model (transfer learning) and untrained model, change the last layer output of those nets, train them on caltech101 dataset with data augmentation and then do the validation:

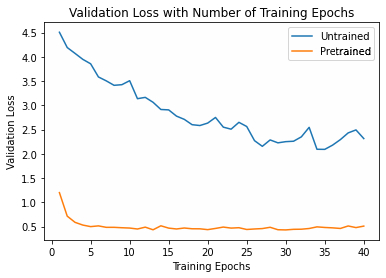
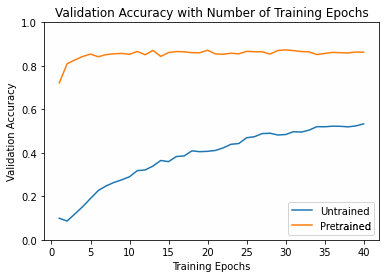
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Pretrained model | | Untrained model | |
|  | Validation accuracy | Time used | Validation accuracy | Time used |
| Resnet-50 | 90.46% | 28m36s | 61.57% | 38m3s |
| VGG-16 | 87.32% | 34m18s | 53.33% | 49m26s |
| Bagnet-9 | 65.27% | 53m46s | 42.79% | 102m45s |
| Bagnet-17 | 73.86% | 53m11s | 47.45% | 100m44s |
| Bagnet-33 | 81.74% | 51m20s | 52.46% | 197m35s |

I use google Colab to run it due to the poor GPU of my laptop, but it’s still extremely slow. For Resnet and VGG I use the batch size 128, but for Bagnet I can only use the batch size 32 or there will be the problem “CUDA out of memory”. I believe the small batch size leads to the bad converge of loss function, thus the accuracy is affected. Obviously untrained model performs worse, which is reasonable because of dataset’s small size.

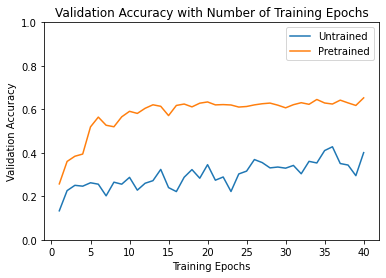
**Resnet-50**



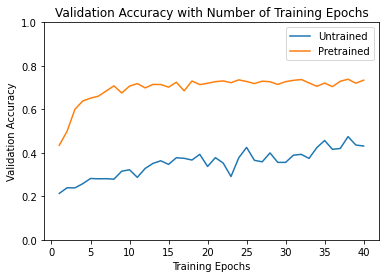
**VGG-16**



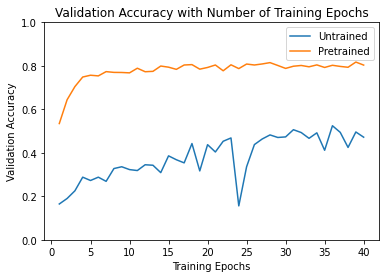
**Bagnet-9**



**Bagnet-17**

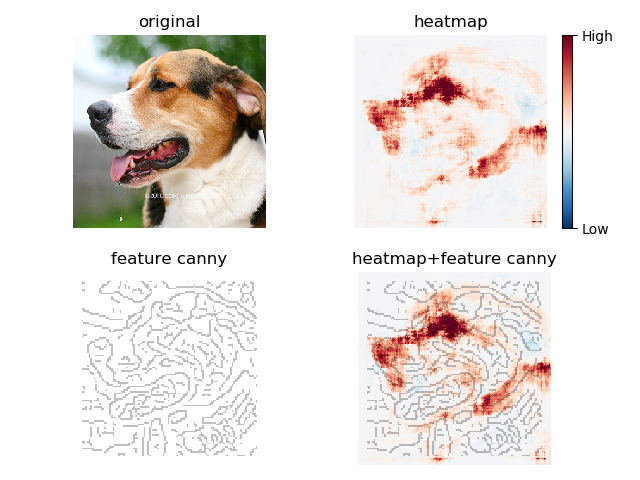


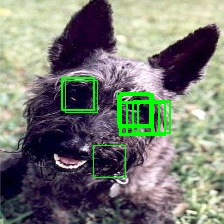
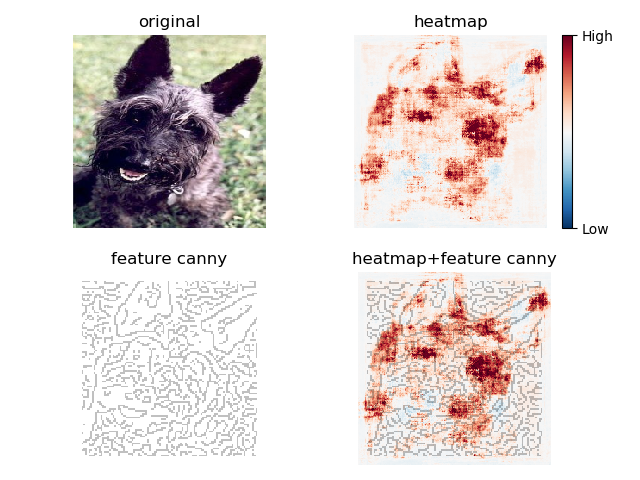
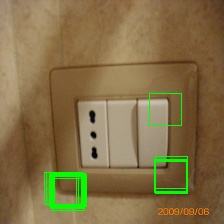
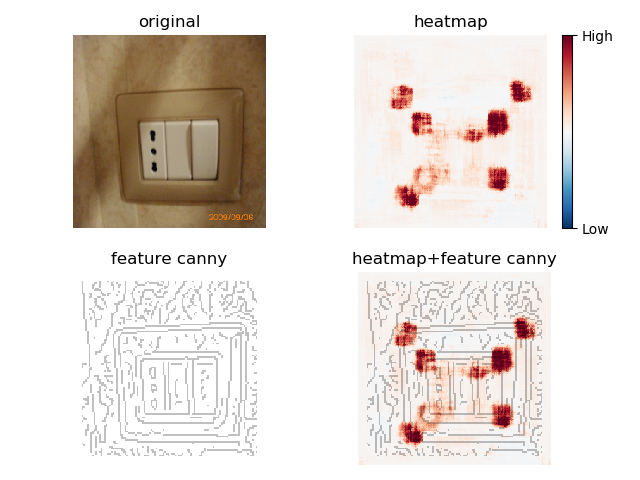
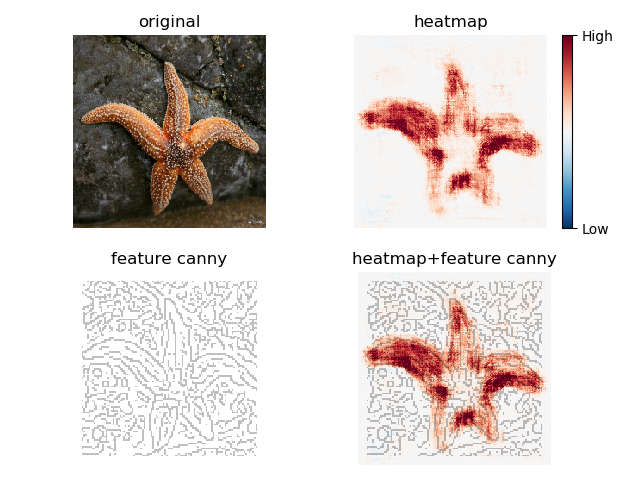
**Bagnet-33**



3.heatmap visualization

Randomly I choose 4 pictures from Imagenet to visualize. Save the heatmap tensor and show it. I also use 33\*33 boxes to mark the most important image patches, whose standard is the logits should be bigger than 99.95% logits from other patches. The visualization is based on Bagnet-33.



****