

# MSBD 5012 Machine Learning Homework 4 Report

Student: XXXXX ID: 12345678

In this homework, I use different pretrained model to classify our test set. I summarize the result in the coming chapters. The first part talks about VGG16, the second part is about ResNet50, the third part shows InceptionV3, the fourth part demonstrates DenseNet121, the fifth part talks about EfficientNetB2. The last part is about MobileNetV2.

## Part One: VGG16

In this part I use a pretrained VGG16 model to classify our test set. We show 5 classes with the maximum probability. The result is as follows:



'rapeseed', 0.99995124

'barn', 2.05e-05

'school bus', 1.01e-05

'worm fence', 5.47e-06

'hay', 2.34e-06



'peacock', 0.9999999

'indigo bunting', 7.09e-08

'fountain', 9.51e-09

'bubble', 8.02e-09

'jay', 7.34e-09



'yurt', 1.0

'lampshade', 3.23e-08

'table lamp', 5.19e-09

'dome', 4.03e-09

'mountain tent', 1.14e-09



'hourglass', 1.0

'guillotine', 1.9e-12

'pedestal', 1.0e-13

'table lamp', 9.5e-14

'China cabinet', 3.6e-14



'water tower', 0.9999951

'beacon', 3.39e-06

'hourglass', 4.44e-07

'radio telescope', 2.81e-07

'dome', 2.54e-07



'zebra', 0.9999989

'gazelle', 4.08e-07

'ostrich', 3.42e-07

'impala', 2.72e-07

'warthog', 2.47e-08



'school bus', 0.9999964

'passenger car', 1.54e-07

'minibus', 3.40e-08

'recreational vehicle', 2.73e-08

'trolleybus', 1.73e-08



'pillow', 0.9998894

'quilt', 3.19e-05

'velvet', 2.36e-05

'pencil box', 1.03e-05

'studio couch', 9.16e-06



'fireboat', 1.0

'fountain', 5.98e-09

'container ship', 1.40e-09

'liner', 7.07e-10

'pier', 4.99e-10



'carousel', 1.0

'toyshop', 1.21e-10

'throne', 4.31e-11




'vestment', 3.53e-11

'slot', 1.69e-11

All the input pictures are resized (224, 224). From the result above, we can see that **VGG16 has 100% accuracy on our 10 pictures. This model can fully cover our task. Besides, from the other result with top 5 probabilities, I think maybe VGG classifies the picture by the texture and the color.**

## Part Two: ResNet50

In this part I use a pretrained ResNet50 model to classify our test set. We show 5 classes with the maximum probability. The result is as follows:

				
'rapeseed', 1.0	'peacock', 1.0	'yurt', 1.0	'hourglass', 1.0	'water tower', 0.9999
'barn', 1.29e-11	'drake', 3.12e-09	'mountain tent', 2.22e-09	'guillotine', 7.24e-13	'airship', 2.16e-07
'fly', 1.04e-11	'fountain', 3.37e-10	'dome', 4.15e-10	'China cabinet', 5.24e-15	'radio telescope', 1.15e-07
'barrel', 8.73e-12	'feather boa', 1.22e-10	'mobile home', 9.87e-11	'beaker', 3.79e-15	'liner', 3.67e-08
'hay', 8.31e-12	'purse', 9.63e-11	'hoopskirt', 8.98e-11	'corkscrew', 1.05e-15	'drum', 3.29e-08
				
'zebra', 0.99999875	'school bus', 0.99999964	'pillow', 0.999976	'fireboat', 1.0	'carousel', 0.99975
'cheetah', 4.29e-06	'minibus', 4.00e-06	'quilt', 7.42e-08	'fountain', 1.84e-06	'Arabian camel', 1.41e-06
'impala', 3.01e-06	'passenger car', 4.56e-07	'studio couch', 6.64e-08	'liner', 9.05e-07	'Indian elephant', 4.75e-07
'gazelle', 1.29e-06	'cab', 3.54e-07	'purse', 5.78e-08	'paddlewheel', 7.22e-07	'mask', 1.39e-07
'hartebeest', 5.30e-07	'trolleybus', 1.30e-07	'pencil box', 1.15e-08	'pier', 6.29e-07	'tusk', 1.30e-07

All the input pictures are resized (224, 224). From the result above, we can see that **ResNet50** even focus more on the main item in the picture. This can be seen easily in the 10<sup>th</sup> picture compared with the result of VGG16. Because the second and the third prediction rates in VGG16 are toyshop and throne (I think mostly decided by the background), but those in ResNet50 refers to Arabian camel and Indian elephant. I think this is because ResNet50 has more layers, even three times more than VGG16. These convolutional layers and linear layers help ResNet50 focus more on the foreground.

## Part Three: InceptionV3

In this part I use a pretrained InceptionV3 model to classify our test set. We show 5 classes with the maximum probability. The result is as follows:

				
'rapeseed', 0.999699	'peacock', 0.986	'yurt', 0.9999889	'hourglass', 0.99976	'water tower', 0.9867
'Appenzeller', 4.34e-06	'leaf beetle', 2.01e-04	'mountain tent', 1.16e-06	'grey fox', 2.75e-06	'armadillo', 1.04e-03
'pick', 4.26e-06	'drake', 1.49e-04	'Leonberg', 2.18e-07	'China cabinet', 2.50e-06	'American black bear', 6.19e-04
'tennis ball', 3.60e-06	'fountain', 1.46e-04	'moving van', 2.11e-07	'red wolf', 2.00e-06	'dung beetle', 2.50e-04
'school bus', 2.92e-06	'tiger beetle', 1.38e-04	'dome', 1.80e-07	'cougar', 1.88e-06	'white wolf', 1.05e-04

				
'zebra', 0.9939	'school bus', 0.9966	'pillow', 0.9953	'fireboat', 0.9524	'carousel', 0.9749
'gondola', 2.04e-04	'toucan', 3.88e-04	'radio', 1.07e-04	'bittern', 0.001	'toyshop', 3.51e-04
'ostrich', 1.51e-04	'hornbill', 6.98e-05	'jeep', 8.44e-05	'badger', 7.63e-04	'confectionary', 2.63e-04
'hartebeest', 1.17e-04	'box turtle', 3.05e-05	'pencil box', 7.05e-05	'fountain', 4.70e-04	'slot', 2.56e-04
'pickup', 5.24e-05	'remote control', 2.44e-05	'mailbag', 4.95e-05	'quill', 4.67e-04	'swing', 2.48e-04

All the input pictures are resized (299, 299) (**different from the above 2 models**). InceptionV3 is different from the above two models because it needs to use its own preprocess input function. This model uses small convolutional kernels, which makes the parameters less than the above two models. From the result above, we can see that Inception also gives the correct answers, but the probabilities are all lower than the above two models. Besides, from the 4<sup>th</sup> results we can find that InceptionV3 uses more of the color of a picture, because a red and grey hourglass can be predicted as grey fox or red wolf (2<sup>nd</sup> and 4<sup>th</sup> largest probability).

#### Part Four: DenseNet121

In this part I use a pretrained DenseNet121 model to classify our test set. We show 5 classes with the maximum probability. The result is as follows:

				
'rapeseed', 0.999845	'peacock', 0.9996	'yurt', 0.9999945	'hourglass', 1.0	'water tower', 0.99977
'stone wall', 3.40e-07	'fountain', 2.23e-04	'mountain tent', 1.31e-06	'scale', 2.92e-10	'beacon', 1.57e-04
'velvet', 9.77e-08	'mosquito net', 3.07e-05	'lampshade', 1.20e-06	'guillotine', 2.13e-10	'radio telescope', 8.38e-06
'snowplow', 8.71e-08	'pillow', 1.11e-05	'barn', 5.47e-07	'pedestal', 1.47e-10	'pier', 6.83e-06
'bison', 6.69e-08	'safety pin', 7.24e-06	'mosquito net', 4.26e-07	'China cabinet', 1.06e-10	'crutch', 3.67e-06











				
'zebra', 0.99993	'school bus', 0.99998	'pillow', 0.99999	'fireboat', 0.99999	'carousel', 0.99997
'tiger', 3.76e-05	'passenger car', 8.37e-07	'pencil box', 7.96e-08	'fountain', 2.09e-06	'altar', 7.70e-06
'tiger cat', 1.51e-05	'amphibian', 4.56e-07	'studio couch', 3.21e-08	'speedboat', 1.96e-07	'throne', 4.68e-06
'worm fence', 8.99e-06	'trolleybus', 2.32e-07	'quilt', 2.35e-08	'mosquito net', 1.29e-07	'vestment', 3.23e-06
'jaguar', 2.54e-06	'cab', 1.99e-07	'purse', 1.63e-08	'lifeboat', 8.70e-08	'maypole', 1.53e-06

All the input pictures are resized (224, 224). Besides, **we use the preprocess input function from the InceptionV3 package** (Otherwise can't get a satisfying result). From the result above,

we can see that DenseNet121 mainly focuses on the texture of the picture. It can be seen in the 2<sup>nd</sup> 3<sup>rd</sup>, 6<sup>th</sup> picture, for the second and third output of these pictures are predicted mainly by the lines of the pictures.

## Part Five: EfficientNetB2

In this part I use a pretrained EfficientNetB2 model to classify our test set. We show 5 classes with the maximum probability. The result is as follows:

				
'rapeseed', 0.8951	'peacock', 0.8475	'yurt', 0.7547	'hourglass', 0.9430	'water tower', 0.8589
'hay', 9.59e-04	'tiger beetle', 0.0123	'mountain tent', 0.0236	'China cabinet', 3.24e-04	'beacon', 7.21e-04
'barn', 6.24e-04	'quill', 8.89e-04	'dome', 0.0019	'hartebeest', 2.22e-04	'jellyfish', 6.61e-04
'harvester', 5.92e-04	'fly', 7.68e-04	'hoopskirt', 0.0018	'Windsor tie', 20.7e-04	'planetarium', 6.57e-04
'maze', 5.73e-04	'humming bird', 6.74e-04	'solar dish', 0.0015	'cabbage butterfly', 1.97e-04	'radio telescope', 6.48e-04
				
'zebra', 0.8916	'school bus', 0.9036	'pillow', 0.9519	'fireboat', 0.84949	'carousel', 0.8844
'ostrich', 9.07e-04	'passenger car', 9.99e-04	'quilt', 6.01e-04	'fountain', 0.0014	'pole', 4.72e-04
'hartebeest', 8.98e-04	'titi', 6.14e-04	'studio couch', 5.62e-04	'geyser', 0.0012	'streetcar', 4.41e-04
'impala', 8.80e-04	'squirrel monkey', 4.70e-04	'velvet', 5.32e-04	'liner', 9.77e-04	'cowboy boot', 4.22e-04
'warthog', 7.95e-04	'capuchin', 3.63e-04	'pencil box', 4.10e-04	'fire engine', 9.24e-04	'mask', 3.92e-04

All the input pictures are resized (260, 260). EfficientNetB2 has a smaller amount of parameters and can get a rather good result. **Note that we don't need to add a preprocessing for the input image, because EffectiveNet has a Rescaling layer for that. EffectiveNet uses a more clear image (with more pixels input) to maintain its accuracy even in a smaller amount of parameters. From the result we can find that the accuracy decreases to some extent, but for a classification problem, the main task is to classify correctly. While other classes still hold relatively small probabilities, I think this model works well.**



## Part Six: MobileNetV2

In this part I use a pretrained ResNet50 model to classify our test set. We show 5 classes with the maximum probability. The result is as follows:



'rapeseed', 0.9971

'hay', 8.80e-05

'picket fence', 4.51e-05

'worm fence', 4.05e-05

'harvester', 3.46e-05



'peacock', 0.9806

'crane', 7.11e-04

'zucchini', 3.96e-04

'fountain', 3.15e-04

'drake', 1.93e-04



'yurt', 0.9536

'mountain tent', 9.64e-04

'ashcan', 9.17e-14

'sandbar', 8.95e-04

'baseball', 4.28e-04



'hourglass', 0.9455

'abacus', 5.08e-04

'wall clock', 4.83e-04

'guillotine', 4.54e-04

'greenhouse', 3.97e-04



'water tower', 0.9624

'pier', 6.07e-04

'parachute', 5.77e-04

'beacon', 4.69e-04

'airship', 4.40e-04



'zebra', 0.9479

'ostrich', 6.59e-04

'maze', 4.32e-04

'impala', 3.00e-04

'lionfish', 2.96e-04



'school bus', 0.9898

'trolleybus', 1.90e-04

'rapeseed', 1.87e-04

'passenger car', 1.53e-04

'fire engine', 1.50e-04



'pillow', 0.9999

'purse', 6.98e-06

'quilt', 6.40e-06

'velvet', 5.35e-06

'studio couch', 5.09e-06



'fireboat', 0.8485

'yawl', 0.0031

'schooner', 0.0020

'fire engine', 0.0015

'paddlewheel', 0.0013



'carousel', 0.9917

'harp', 0.0011

'panpipe', 0.0007

'cowboy boot', 6.27e-04

'toyshop', 4.75e-04

All the input pictures are resized (224, 224). Besides, **we use the preprocess input function from the MobileNet package** (Otherwise can't get a satisfying result). MobileNet is a relatively light network because it is mainly used on mobile devices. From the result above, we can see that **MobileNetV2 mainly learns texture (lines) and color to classify the picture. The most salient example is the 2<sup>nd</sup> and the 6<sup>th</sup> picture (lines), and the 7<sup>th</sup> picture (color).**

## Conclusion

In this experiment I use some pretrained models to classify existing pictures. These models have their own cons and pros. And I also try to analyze the result that model gives.