Capstone Project

Machine Learning Engineer Nanodegree

Liang Huiying
November 22, 2016

I. Definition

Project Overview

Cats and dogs are the most favorite pets in human being's life, therefore, we always have photos or pictures containing cats or dogs in our albums. And sometimes we try to classify them from each other. What if we do these things automatically rather than by our own hands or eyes? With the technology of computer vision, the knowledge of machine learning or more precisely deep learning, these problems can be solved.

In order to do so, I build a classifier which is trained by the dataset downloaded from kaggle[1] and runs on the Jupyter Notebook in this project.

Project Statement

The goal is when you put an image into the classifier it will tell you it is cats or dogs in the image. The project consists of the following parts:

1. Download dataset from kaggle and preprocess the data.
2. Build the structure of the classifier.
3. Train the classifier.
4. Use the classifier to recognize cats and dogs.

Metrics

Validation accuracy and Prediction Time are used to evaluate the performance of the classifier. Validation accuracy is a typical metrics for a classifier, it will show how well the classifier performance.

$$validation\ accuracy = \frac{numbers\ of\ right\ predictions\ \ dogs\ or\ cats}{total\ number\ of\ images} * 100\%$$

Prediction Time is also a useful metrics for this classifier, it could tell how fast the classifier runs.

$$Prediction\ Time\ = \frac{time\ used\ for\ predicting\ the\ test\ dataset}{test\ dataset\ size}$$

[1] https://www.kaggle.com/c/dogs-vs-cats-redux-kernels-edition/data

II. Analysis

Data Exploration

The dataset downloaded from kaggle contains two files, *test* and *train*. The *train* folder contains 25,000 images of dogs and cats. Each image in this folder has the label as part of the filename. The *test* folder contains 12,500 images, named according to a numeric id without label. Each image from *test* or *train* folders have different shapes and sizes.

Exploration Visualization

Figure.1. Images from dataset

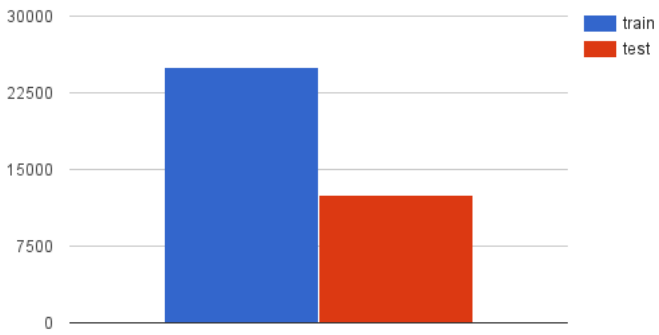

Figure.2. Size of ‘test’ and ‘train’ files

Algorithms and Techniques

The classifier is a ResNet-50[2] but I replace the top layer *fc1000* with one neuron.

[2] https://github.com/KaimingHe/deep-residual-networks

Figure.3. shows the structure of ResNet-50. The Resnet-50 consists of a 7x7 convolution layer, 4 convolution blocks, 12 identity blocks and a 1000 full connection layer.

Each identity block consists of 3 convolution layers and a shortcut without convolution layer. The structure of identity block shows in Figure.5.

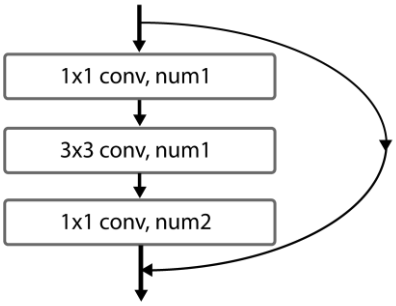

Figure.5. identity_block

Each convolution block consists of 3 convolution layers and a shortcut with 1 convolution layer. The structure of convolution block shows in Figure.6.

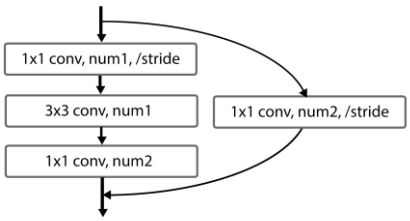

Figure.6. conv_block

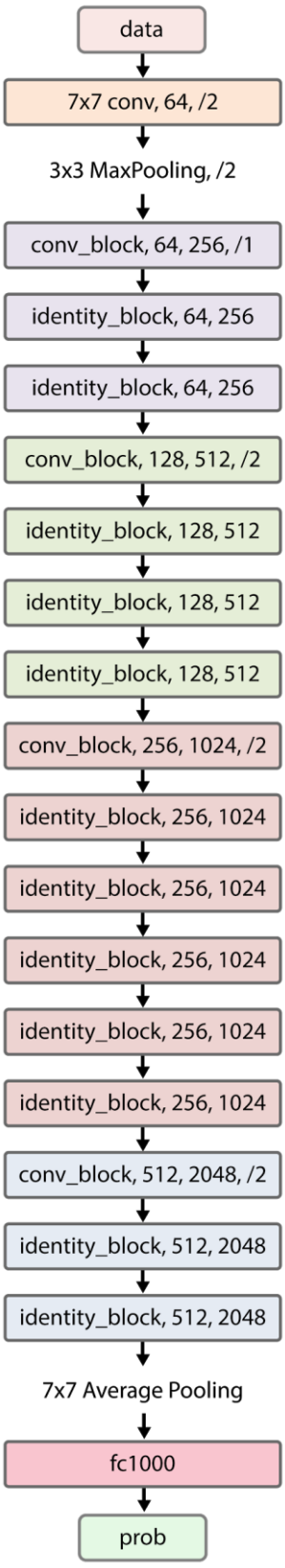

Figure.3. ResNet-50

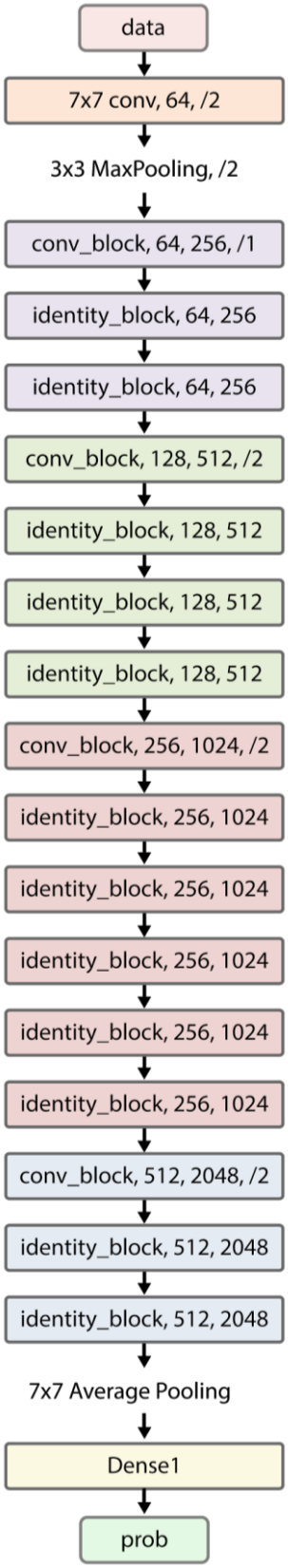

Figure.4. ResNet-50 for Cats.Vs.Dogs

ResNet-50 is good at image classification, detection and localization, so it is not a tough task for ResNet-50 to distinguish cats or dogs from each other. Normal deep convolutional neural networks also work when deal with such job. Usually, the deeper the convolutional neural network is, the better the accuracy shows. But when the network goes deeper again, the accuracy lowers because the gradient disappears. In ResNet-50, every identity block and convolution block has a shortcut. Therefore, ResNet-50 with 49 convolutional layers still does a good job and better than other networks.

In this project, the network is named *ResNet-50 for Cats.Vs.Dogs*. using the structure of ResNet-50 and replacing the top layer from 1000 full connection to one neuron because the goal is to distinguish cats or dogs.

And I also keep the weights trained by ImageNet. This is a technique named t*ransfer learning*. Transfer learning is the improvement of learning in a new task through the transfer of knowledge from a related task that has already been learned. [3] By this way, the classifier becomes an experienced network. So when it starts to learn images of cats and dogs, it will generalize much more typical and representative features of cats and dogs.

Benchmark

The 1-crop validation error of ResNet-50 on ImageNet is 24.7%.[4] In other words, the accuracy is 75.3%, but the result is based on ImageNet which is much more challenging.

In this project, the goal of accuracy is above 90% and the prediction time less than 0.05s per image which is both effective and efficient.

III. Methodology

Data Preprocessing

The preprocessing consists of the following steps:

1. The images in *train* folder are divided into a training set and a validation set.
2. The images both in training set and validation set are separately divided into two folders -- *cat* and *dog* according to their lables.
3. The RGB color values of the images are rescaled to 0~1.
4. The size of the images are resized to 224*224.

[3] http://ftp.cs.wisc.edu/machine-learning/shavlik-group/torrey.handbook09.pdf

[4] https://github.com/KaimingHe/deep-residual-networks

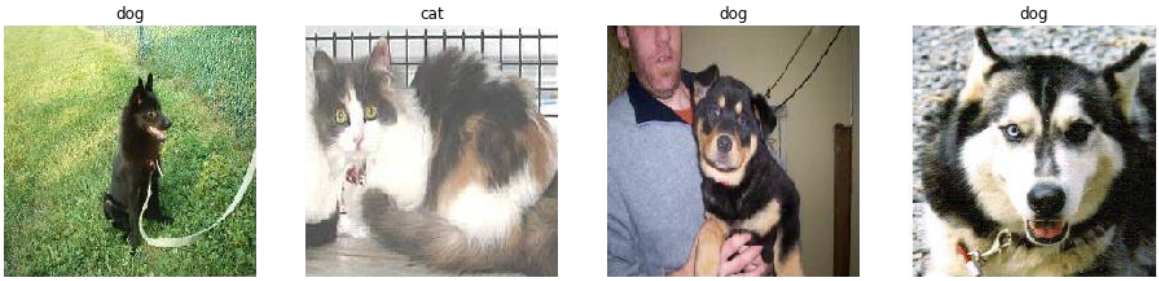

Figure.7. Images from the training set after being resized

Implementation

The implementation contains the following main steps:

- Build the structure of ResNet-50 for Cats.Vs.Dogs (The ResNet-50 for Cats.Vs.Dogs. consists of a 7x7 convolution layer, 4 convolution blocks, 12 identity blocks and a dense layer.)
  - Define identity block.
  - Define convolution block.
  - Build the structure of ResNet-50 without top layer.
  - Load weights
  - Add top layer to ResNet-50.
  - Setup training attribute.
  - Compile the model.
- Train ResNet-50 for Cats.Vs.Dogs.
- Save the best model.
- Use the best model to predict cats and dogs.

Refinement

An initial solution to distinguish cats and dogs from each other is also using ResNet-50. The ResNet-50 is trained by the dataset download from kaggle and without transfer learning, so this ResNet-50 did not load the weights trained by ImageNet.

In order to get the val_acc (validation accuracy) above 90%, I have tried to train 200 epochs with 2048 samples per epoch. The plot in Figure.8. shows that after training 200 epochs the validation accuracy reaches to 0.9035. It has not reached the upper limit, but I think it should perform better with training less epochs.

Therefore, I load weights trained by ImageNet, freeze the weights of the last 49 layer, training the top layer of the network by the dataset download from kaggle. I only tried 20 epochs with 2048 samples per epoch and the best validation accuracy reaches up to 0.9812. It is almost a perfect result. The validation accuracy and loss of ResNet-50 with transfer learning shows in Figure.9.

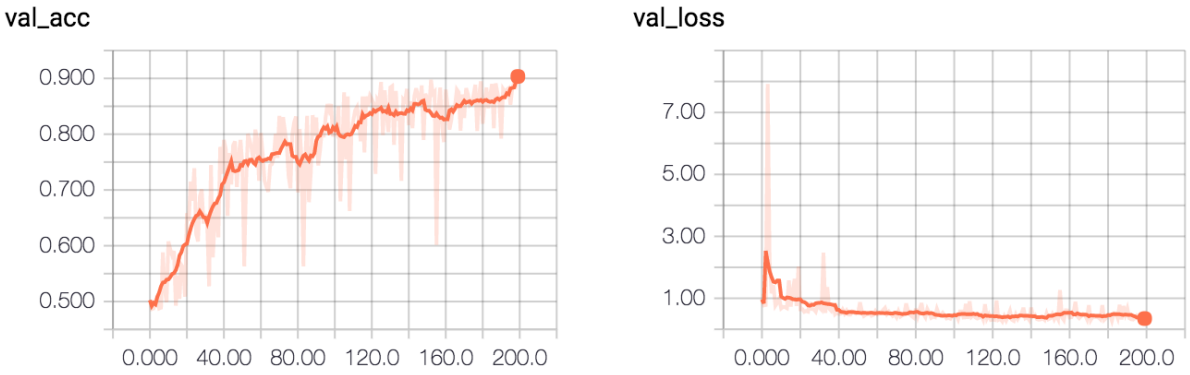

Figure.8. The validation accuracy and loss of ResNet-50 without transfer learning.

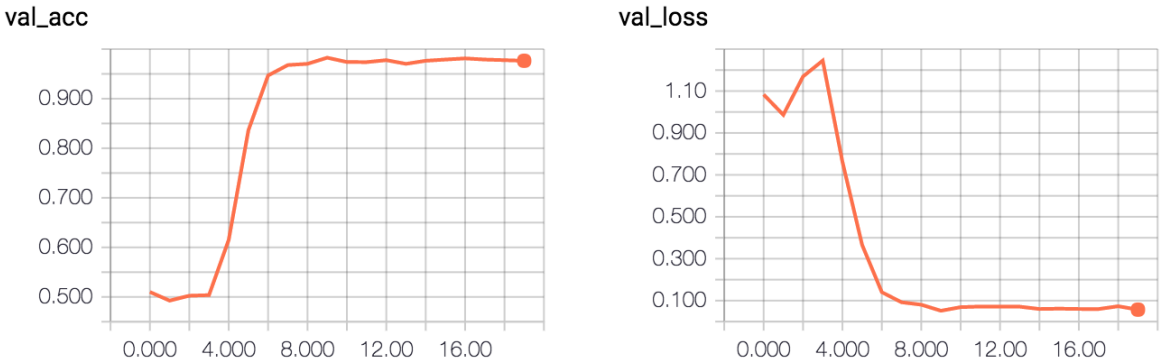

Figure.9. The validation accuracy and loss of ResNet-50 with transfer learning.

IV. Results

Model Evaluation and Validation

As mentioned in the Benchmark, the goal is accuracy is above 90% and the prediction time less than 0.05s per image which is both effective and efficient.

According to the comparison in *Refinement*, ResNet-50 with transfer learning is good enough for the problem in this project. The ResNet-50 for Cats.Vs.Dogs. consists of a 7x7 convolution layer, 4 convolution blocks, 12 identity blocks, a dense layer and loading weights trained by ImageNet.

The validation accuracy is above 95% after 6 epochs' training and the loss lowers than 0.1 after 8 epochs' training.

Justification

The plot in Figure.9. shows the best validation accuracy of ResNet-50 for Cats.Vs.Dogs is 98.12% which has achieved the goal.

The total prediction time of 12500 images is 68s, so it cost 0.00544s per image. The classifier runs on the computer whose CPU is i7 6700k, GPU is GTX 980 Ti, Memory size is 32GB.

20 random images in the *test* folder predicted by the classifier shows in Figure.10. and each of them is correct.

V. Conclusion

Free-Form Visualization

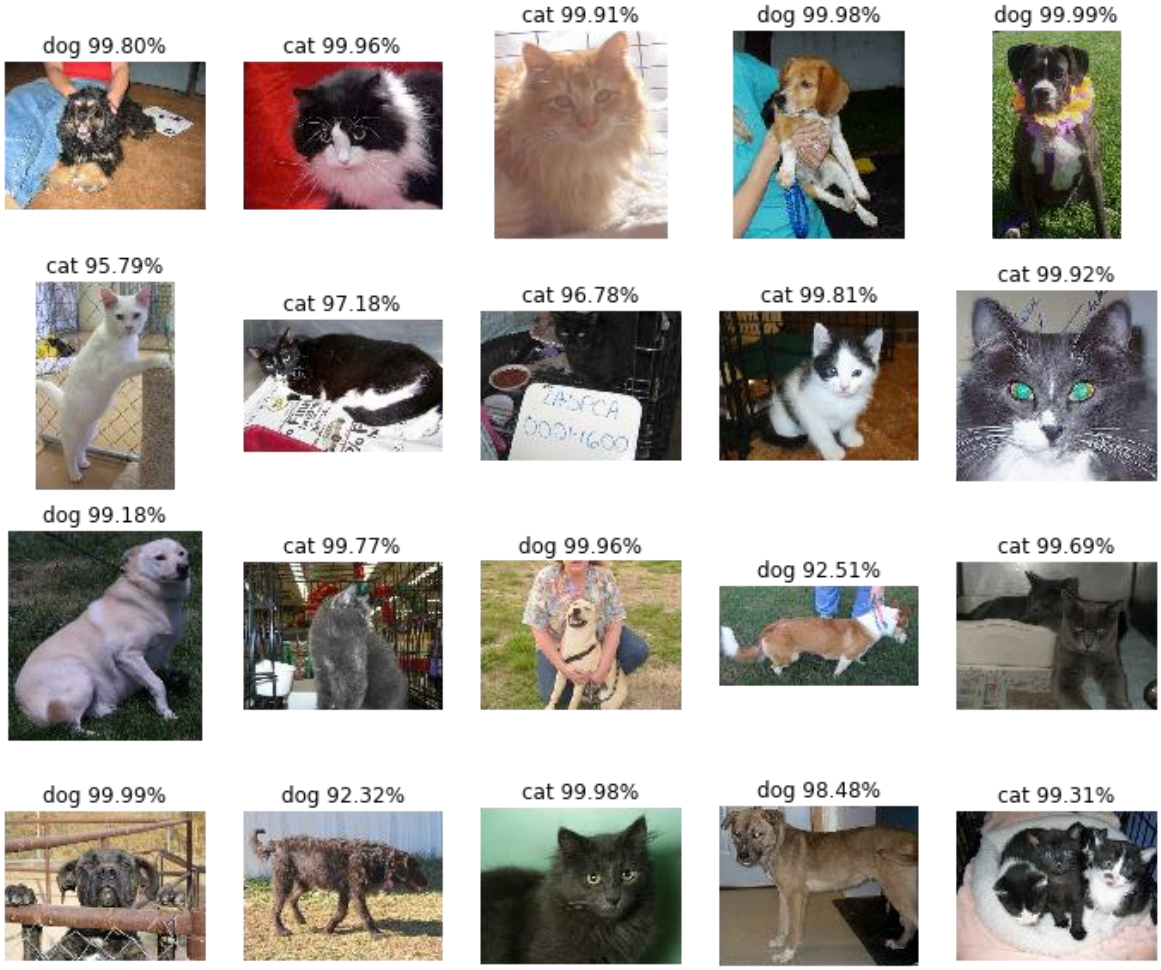

Figure.10. Random prediction result

20 random images in the *test* folder predicted by the classifier shows in Figure.10. and each of them is correct. Although the classifier could not give a 100% answer, it has done a good job at distinguishing cats or dogs from each other.

Reflection

The process used for this project contains the following steps:

1. Download dataset from kaggle and preprocess the data.
2. Build the structure of ResNet-50 for Cats.Vs.Dogs
    a. Define identity block.
    b. Define convolution block.
    c. Build the structure of ResNet-50 without top layer.
    d. Load weights
    e. Add top layer to ResNet-50.
    f. Setup training attribute.
    g. Compile the model.
3. Train ResNet-50 for Cats.Vs.Dogs.
4. Save the best model.
5. Use the best model to predict cats and dogs.

The most difficult part in this project is also the most interesting part. At first, I just use the dataset to train ResNet-50 for Cats.Vs.Dogs without transfer leaning. Actually, Build the structure of ResNet-50 for Cats.Vs.Dogs is also a tough task. I tried and trained so hard but the improvement was slow, so I try to draw the feature heatmap to know how the classifier make a decision.

Figure.11. shows feature heatmap created by ResNet-50 without transfer learning and Figure.12. shows feature heatmap created by ResNet-50 with transfer learning. It is easy to know from Figure.12. that the classifier think it is a cat because it has a cat-like face or it is a dog because it has dogs' nose, dogs' eyes, dogs' ears, etc. But in Figure.11., things are different. From image (3,3) in Figure.11. the classifier think it is a dog because it is outside on the meadow rather than it has a dog-like face or at least dog-like body. Maybe, dogs more like outside than cats do, but the classifier is not satisfying. This is also why I use transfer learning to make an improvement.

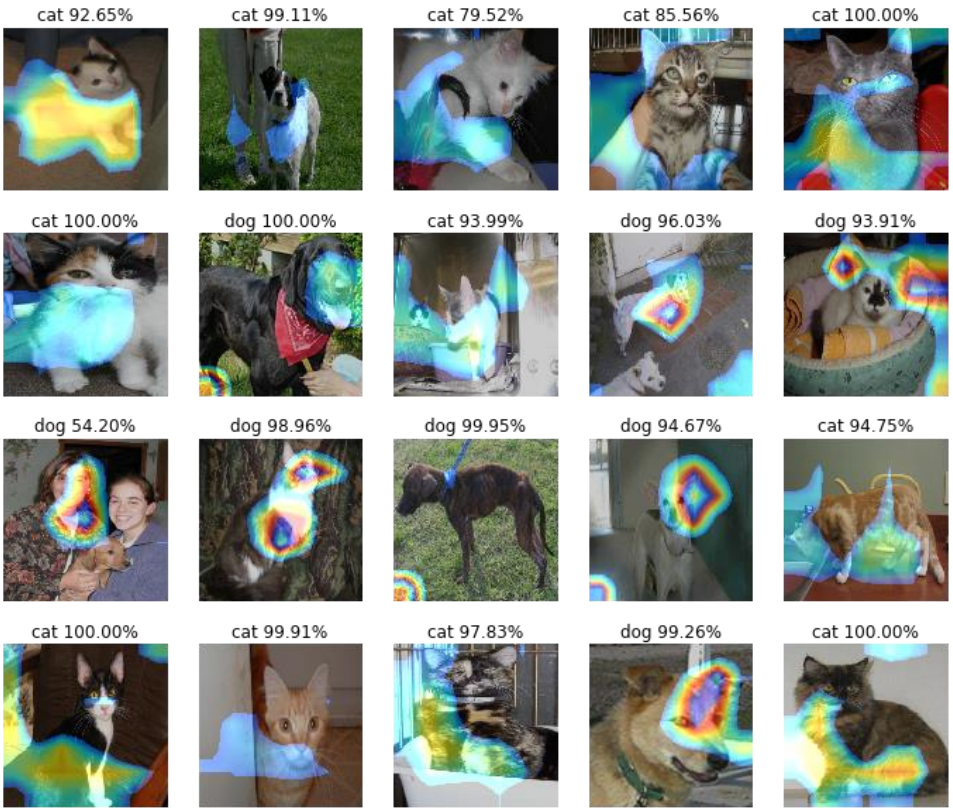

Figure.11. Feature heatmap created by ResNet-50 without transfer learning

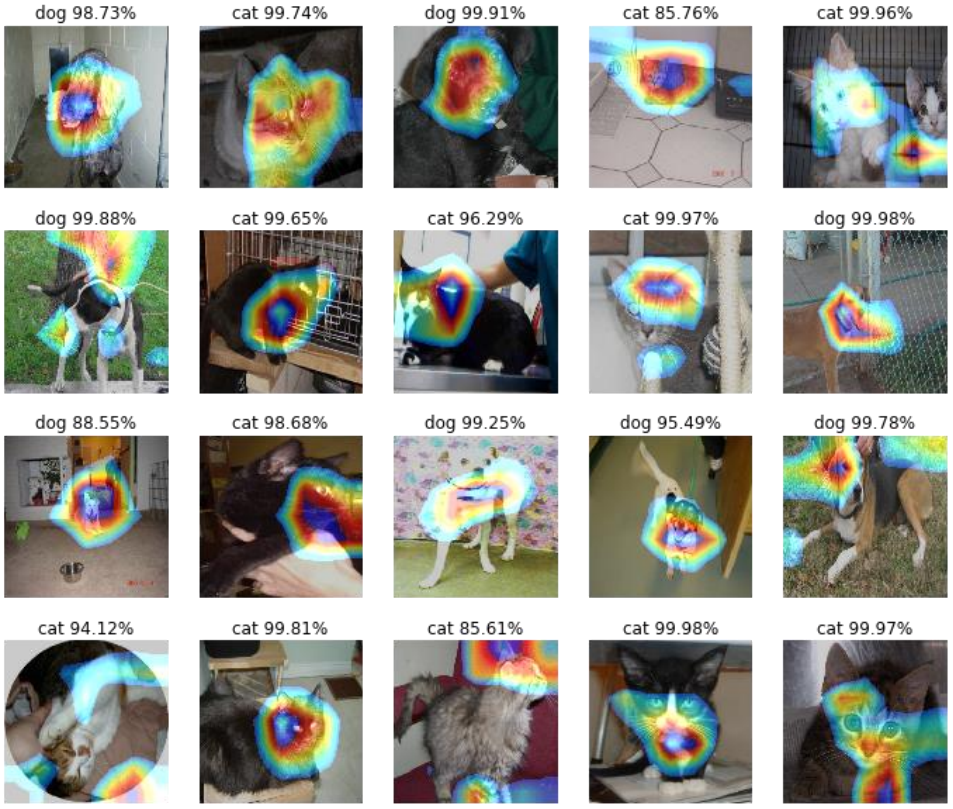

Figure.12. Feature heatmap created by ResNet-50 with transfer learning

Improvement

To achieve higher accuracy, the following improvement is worth trying.

- Use ResNet-101, ResNet-152 even 1K-layer ResNets. The 1-crop validation error on ImageNet of ResNet-101 is 23.6% while ResNet-50 is 24.7%. There is no doubt that ResNet-101 will perform better in this project.
- More kinds of images in dataset is also helpful. Cats or dogs are taken photo of both indoors and outdoors. Images could be zoomed in or zoomed out or rotated to some angles.
- Dropout 7x7 Average Pooling output.
- Fine-tune[5] one or more convolution blocks.

[5] https://blog.keras.io/building-powerful-image-classification-models-using-very-little-data.html