Capstone Project

Machine Learning Engineer Nanodegree

Liang Huiying
November 22, 2016

I. Definition

Project Overview

Cats and dogs are the most favorite pets in a human's life, therefore, we always have photos containing cats or dogs in our albums. Sometimes we try to classify these photos into categories. What if we can 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.

Many people have tried different ways to solve this problem and submitted their results on kaggle[1]. Most of them really did very good jobs while some are not so satisfied.

In this project, I built a classifier trained by the dataset downloaded from kaggle. The dataset contains a large amount of ordinary daily images including cats or dogs. The classifier runs on the Jupyter Notebook.

Project Statement

This is a supervised learning[2] problem because each example in training data is a pair consisting of an input image and a desired output value (*cat* or *dog*). The classifier implements a binary classification considering the goal is to tell you if it is cats or dogs in the image when you put an image into the classifier.

The project consists of the following parts:

1 Download dataset from kaggle.
2 Preprocess the data.
  2.1 Divide images in *train* folder into a training set and a validation set.
  2.2 Resize and rescale images in *train* folder.
3 Build the structure of the classifier.
  3.1 The basic structure of the classifier is ResNet-50.
  3.2 Transfer learning is used to make improvements of the accuracy.
4 Train the classifier.
5 Use the classifier to recognize cats and dogs.

[1] https://www.kaggle.com/c/dogs-vs-cats-redux-kernels-edition
[2] https://en.wikipedia.org/wiki/Supervised_learning

Metrics

Accuracy, time and log loss[3] are used to evaluate the performance of the classifier. Accuracy and log loss are typical metrics for a classifier, it will show how well the classifier performance.

$$accuracy = \frac{n}{N} * 100\%$$

- n: number of right predictions of dogs or cats
- N: total number of images in test dataset

$$LogLoss = -\frac{1}{n}\sum_{i=1}^{n}[y_i \log(\hat{y_i}) + (1 - y_i)\log(1 - y_i)]$$

- n: the number of images in the test set
- $\hat{y_i}$: the predicted probability of the image being a dog
- y_i : 1 if the image is a dog, 0 if cat

A higher accuracy and a smaller log loss is better.

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

$$time = \frac{T}{N}$$

- T: time used for predicting the test dataset
- N: total number of images in test dataset

II. Analysis

Data Exploration

The dataset downloaded from kaggle contains two files, *test* and *train*. The *train* folder contains 12500 images of dogs and 12500 images of 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. Images from *test* or *train* folders are ordinary daily images including cats or dogs with shapes and sizes.

I split *train* folder into 2 folders:

- *mytrain* ---- including two folders
  - *cat* ---- including about 11250 cat images
  - *dog* ---- including about 11250 dog images
- *myvalid* ---- including two folders
  - *cat* ---- including about 1250 cat images
  - *dog* ---- including about 1250 dog images

[3] https://www.kaggle.com/c/dogs-vs-cats-redux-kernels-edition/details/evaluation

Exploration Visualization

Figure.1. Images from dataset

Figure.2. Images from dataset

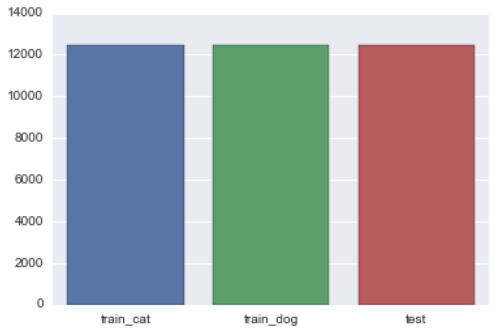

Figure.3. number of images from original dataset

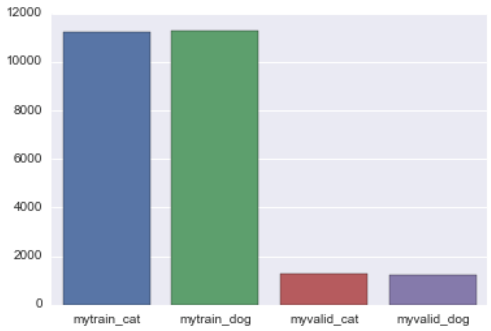

Figure.4. numbers of images after split in *mytrain*

Algorithms and Techniques

The basic structure of the classifier is ResNet-50[4]. The full name of ResNet is Deep Residual Networks. 50 means this model contains 50 layers. ResNet was invented by Kaiming He and his team for the reason of image recognition and won the 1st places in ImageNet classification. ImageNet is an image database organized according to the WordNet hierarchy, in which each node of the hierarchy is depicted by hundreds and thousands of images[5].

ResNet is one of the Convolutional Neural Networks(CNNs). They are made up of neurons that have learnable weights and biases. Each neuron receives some inputs, performs a dot product and optionally follows it with a non-linearity[6].

Deeper neural networks are more difficult to train(DNNs). A residual learning framework could ease the training of networks that are substantially deeper than those used previously. ResNet explicitly reformulates the layers as learning residual functions with reference to the layer inputs, instead of learning unreferenced functions.

In this project, the classifier is a ResNet-50 but I replace the top layer *fc1000* with one neuron.

[4] https://github.com/KaimingHe/deep-residual-networks
[5] http://image-net.org/
[6] http://cs231n.github.io/convolutional-networks/

I chose ResNet rather than other simple models like SVM. I have tried to train a SVM classifier but it took much more time than ResNet. SVM makes its decisions based on distance which is not good at dealing with so many images.

Figure.5. 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.7.

Each convolution block consists of 3 convolution layers and a shortcut with 1 convolution layer. The structure of convolution block shows in Figure.8.

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,

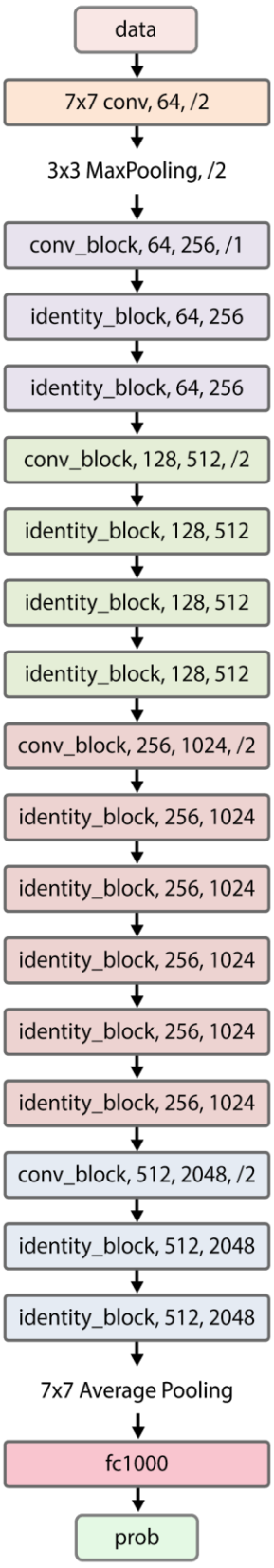

Figure.5. ResNet-50

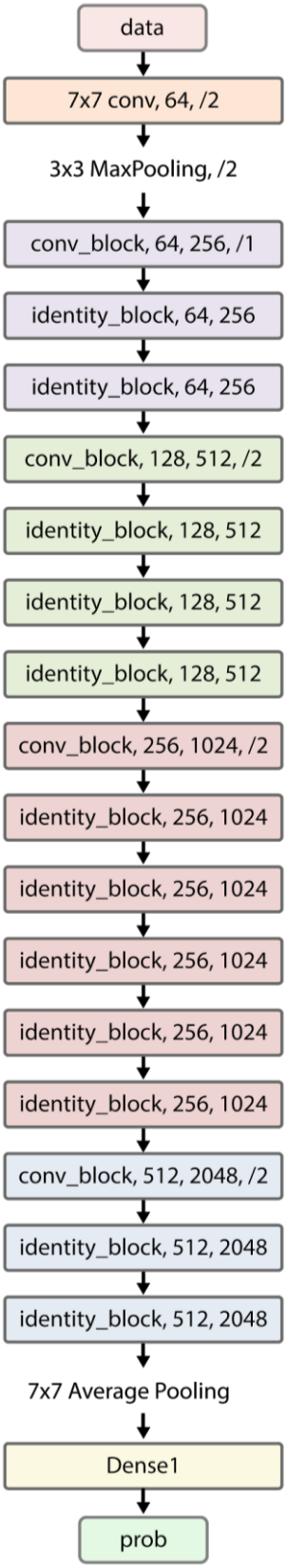

Figure.6. ResNet-50 for Cats.Vs.Dogs

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.

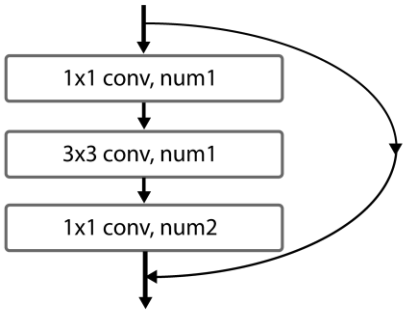

Figure.7. convolution block

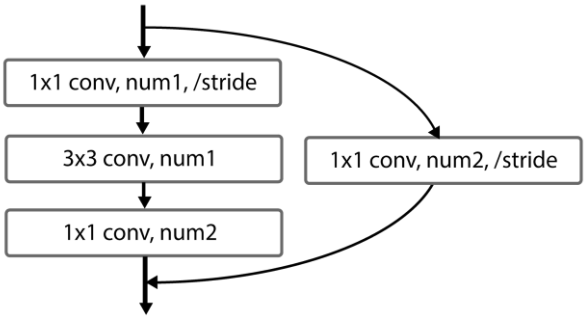

Figure.8. identity block

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[7]. 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%.[8] 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 is to keep accuracy above 90% and the prediction time less than 0.05s per image which is both effective and efficient. The log loss less than 0.69315 which is the all 0.5 benchmark on kaggle[9].

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 labels.

7 http://ftp.cs.wisc.edu/machine-learning/shavlik-group/torrey.handbook09.pdf
8 https://github.com/KaimingHe/deep-residual-networks
9 https://www.kaggle.com/c/dogs-vs-cats-redux-kernels-edition/leaderboard

3. The RGB color values of the images are rescaled to 0~1.
4. The sizes of the images are resized to 224*224.

Figure.9. 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 images.

I use keras to build the structure of ResNet-50 for Cats.Vs.Dogs. Keras is a high-level neural networks library, written in Python and capable of running on top of either TensorFlow or Theano[10]. Keras make it more convenient to design convolutional networks.

The weights of ResNet-50 for Cats.Vs.Dogs. without top layer were downloaded from keras and trained by ImageNet. The weights of top layer were trained by the dataset downloaded from kaggle.

The loss of the model is binary cross-entropy, the training method of the model is adadelta which dynamically adapts over time using only first order information and has minimal computational overhead beyond vanilla stochastic gradient descent[11].

[10] https://keras.io/
[11] http://www.matthewzeiler.com/pubs/googleTR2012/googleTR2012.pdf

I use dropout to prevent ResNet-50 for Cats.Vs.Dogs. from overfitting. Dropout will randomly cut up a certain percentage connection of the full connection of top layer.

Refinement

The refinement made in this project was that I was not just build the structure of ResNet-50 and trained the classifier directly by the dataset downloaded from kaggle, I use transfer learning to improve the accuracy.

An initial solution to distinguish cats and dogs from each other is using a ResNet-50 with random weights. The ResNet-50 is trained by the dataset downloaded from kaggle and without transfer learning.

In order to obtain the val_acc (validation accuracy) above 90%, I have tried to train 200 epochs with 2048 samples per epoch. The plot in Figure.12. shows that after training 200 epochs the validation accuracy reaches to 0.89 and the training accuracy is above 0.93. It has not reached the upper limit, but the classifier is overfitting. So, I think there are better solutions which will achieve higher accuracy and cost less training time.

In order to make a refinement, I load weights trained by ImageNet, freeze the weights except the top layer, training the top layer of the network by the dataset download from kaggle. I have tried 40 epochs with 2048 samples per epoch to find the best model and the best validation accuracy reaches up to 0.98. It is almost a perfect result. The validation accuracy and loss of ResNet-50 with transfer learning shows in Figure.13. The training accuracy is less than 0.95 which means it is not overfitting.

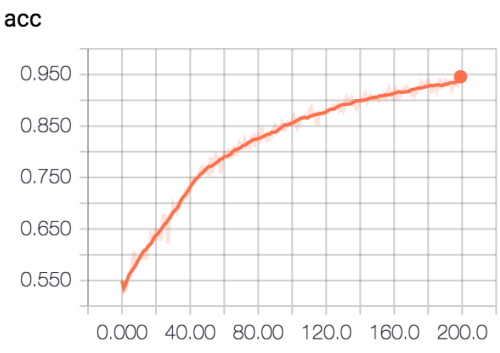

Figure.10. the training accuracy of ResNet-50 without transfer learning

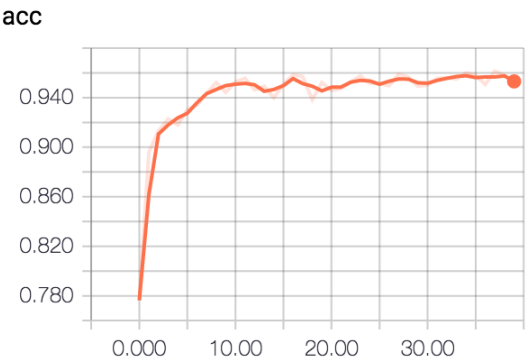

Figure.11. the training accuracy of ResNet-50 with transfer learning

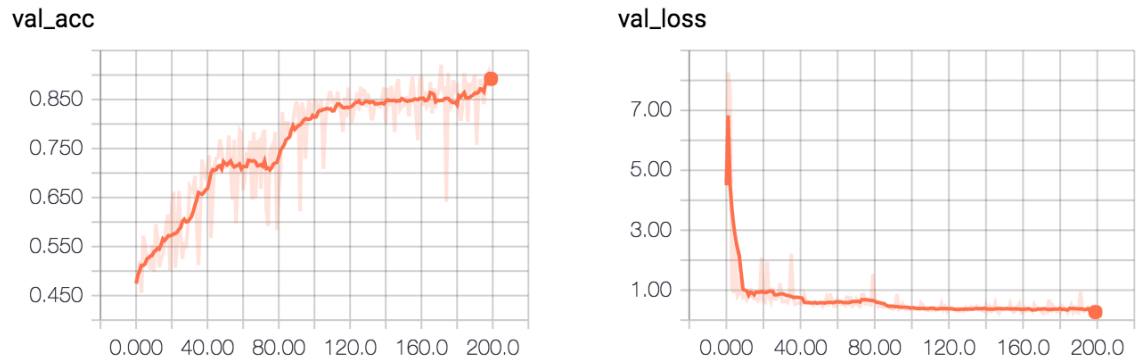

Figure.12. The validation accuracy and loss of ResNet-50 without transfer learning.

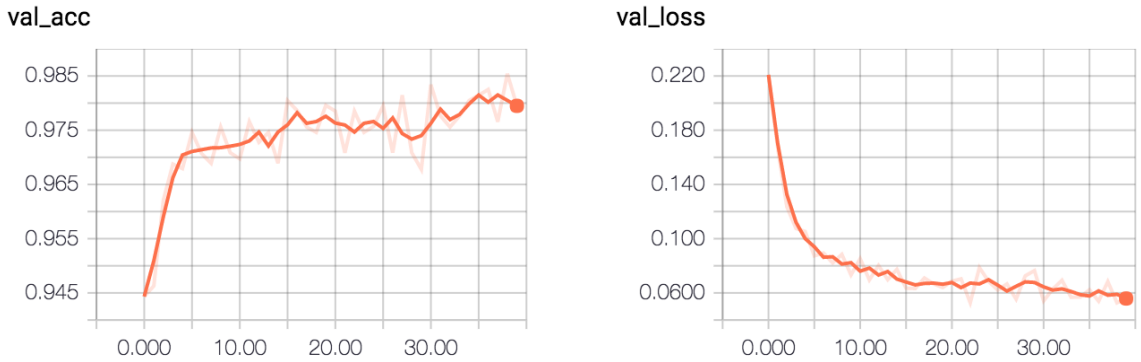

Figure.13. The validation accuracy and loss of ResNet-50 with transfer learning.

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.

After 5 epochs' training, the validation accuracy is above 95% and the loss lowers than 0.1. After 40 epochs' training, the training accuracy shows in Figure.11. is less than 0.95 and the validation accuracy is above 0.98 shows in Figure.13. The classifier is not overfitting.

I submitted the result.csv which contains the prediction of all images in test dataset and the score is 0. 0.08708.

Justification

The plot in Figure.13. shows the best validation accuracy of ResNet-50 for Cats.Vs.Dogs is 98.54% which has achieved the goal.

The total prediction time of 12500 images is 68s, so it costs 0.00544s per image. The classifier ran on a computer with the following components:

1. CPU is i7 6700k
2. GPU is GTX 980 Ti
3. Memory size is 32GB.

20 random images in the *test* folder predicted by the classifier are shown in Figure.14. and each of them was predicted correctly.

V. Conclusion

Free-Form Visualization

Figure.14. Random prediction result

20 random images in the *test* folder predicted by the classifier are shown in Figure.10. and each of them was predicted correctly. 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 used the dataset to train ResNet-50 for Cats.Vs.Dogs without transfer leaning. Building the structure of ResNet-50 for Cats.Vs.Dogs was also a tough task. I tried and trained very hard but the improvement was slow, so I tried to draw the feature heatmap to understand how the classifier made its decisions.

The shape of the output of the base model is (7, 7, 2048). The shape of the weights of full connection is (2048, 1). In order to draw the heatmap, I calculated the Class Activation Mapping[12] of the output of the network then used OpenCV to visualize the result.

$$cam = (P - 0.5) * output * w$$

- cam: class activation mapping
- P: the probability of cats or dogs
- output: the output of base model
- w: the weights of the full connection

Figure.15. shows the feature heatmap created by ResNet-50 without transfer learning and Figure.16. shows the feature heatmap created by ResNet-50 with transfer learning. It is easy to recognize from Figure.16. that the classifier recognizes a cat because the image has a cat-like face or it is a dog because it has dogs' nose, dogs' eyes, dogs' ears, etc. Figure.15., shows a different perspective. From image (3,3) in Figure.15. the classifier recognized a dog because it is outside on a meadow field rather than a dog-like face or at least dog-like body. Maybe, dogs are more likely to be outside than cats, but the classifier is not satisfied. This is also why I use transfer learning to make improvements.

[12] http://cnnlocalization.csail.mit.edu/

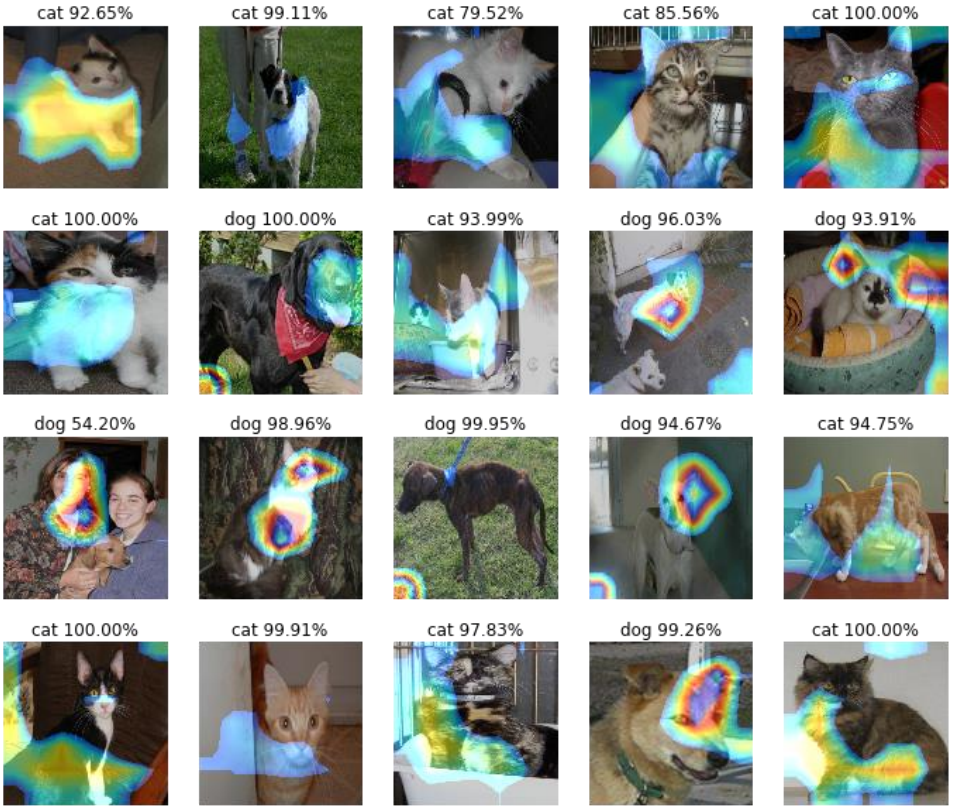

Figure.15. Feature heatmap created by ResNet-50 without transfer learning

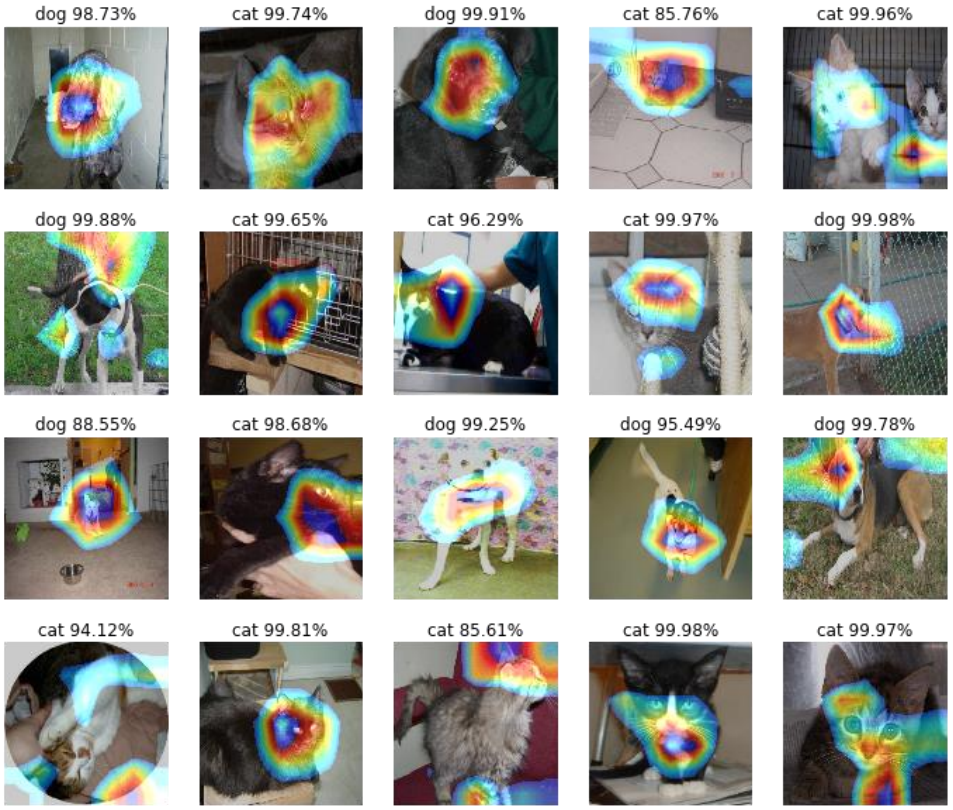

Figure.16. Feature heatmap created by ResNet-50 with transfer learning

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 a large possibility that ResNet-101 will perform better in this project.
- More diversity 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.
- Fine-tune[13] one or more convolution blocks.

VI. Reference

1 https://www.kaggle.com/c/dogs-vs-cats-redux-kernels-edition
2 https://en.wikipedia.org/wiki/Supervised_learning
3 https://www.kaggle.com/c/dogs-vs-cats-redux-kernels-edition/details/evaluation
4 https://github.com/KaimingHe/deep-residual-networks
5 http://image-net.org/
6 http://cs231n.github.io/convolutional-networks/
7 http://ftp.cs.wisc.edu/machine-learning/shavlik-group/torrey.handbook09.pdf
8 https://github.com/KaimingHe/deep-residual-networks
9 https://www.kaggle.com/c/dogs-vs-cats-redux-kernels-edition/leaderboard
10 https://keras.io/
11 http://www.matthewzeiler.com/pubs/googleTR2012/googleTR2012.pdf
12 http://cnnlocalization.csail.mit.edu/
13 https://blog.keras.io/building-powerful-image-classification-models-using-very-little-data.html

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