

# EE\_628\_project

Jingyu Luo

Jiahao Zhang

Ying Lu

Classify whether images contain either a dog or a cat

## Ideas:

1. Data preparation, the training set is divided into train (before 2/3), validation (after 1/3)
2. Try simple cnn, accuracy 90%.
- 3.1. Try vgg16 fine tune 1 (using the vgg16 convolutional layer extraction feature, build a fully connected layer on its own training only top), training set 99%, accuracy of about 93%, overfitting
- 3.2 Try vgg16 fine tune 2 Use vgg16's convolutional layer extraction feature, construct a fully connected layer by yourself to train only the top, and then stitch the already weighted vgg16 convolution and the top of your own trained weight to lock the first few volumes of vgg16. Product block, training only the last convolution block and top, fine tune with a small learning rate, accuracy is about 93%
4. Use a number of different models (ResNet50, InceptionV3, VGG19) to extract features, stitch together the features of multiple different models, and construct a fully connected layer by themselves to train only the top. The accuracy is about 98%. (The three models are selected because Xception has a memory error in the extraction of features, and a memory error occurs when these three types of vgg16 are added.)

## **Relization:**

We decompose the problem into 6 parts, as in the folder:  
final\_dogs&cats\_withResult\_code

Part1: New folder & Copy images\_tested.ipynb

We create a new folder and copy the pictures from train to the corresponding folder.

Only need to modify two parameters, path\_folder is the location of the folder, and path\_img is the location of the training set. Modify the location parameters directly after run,

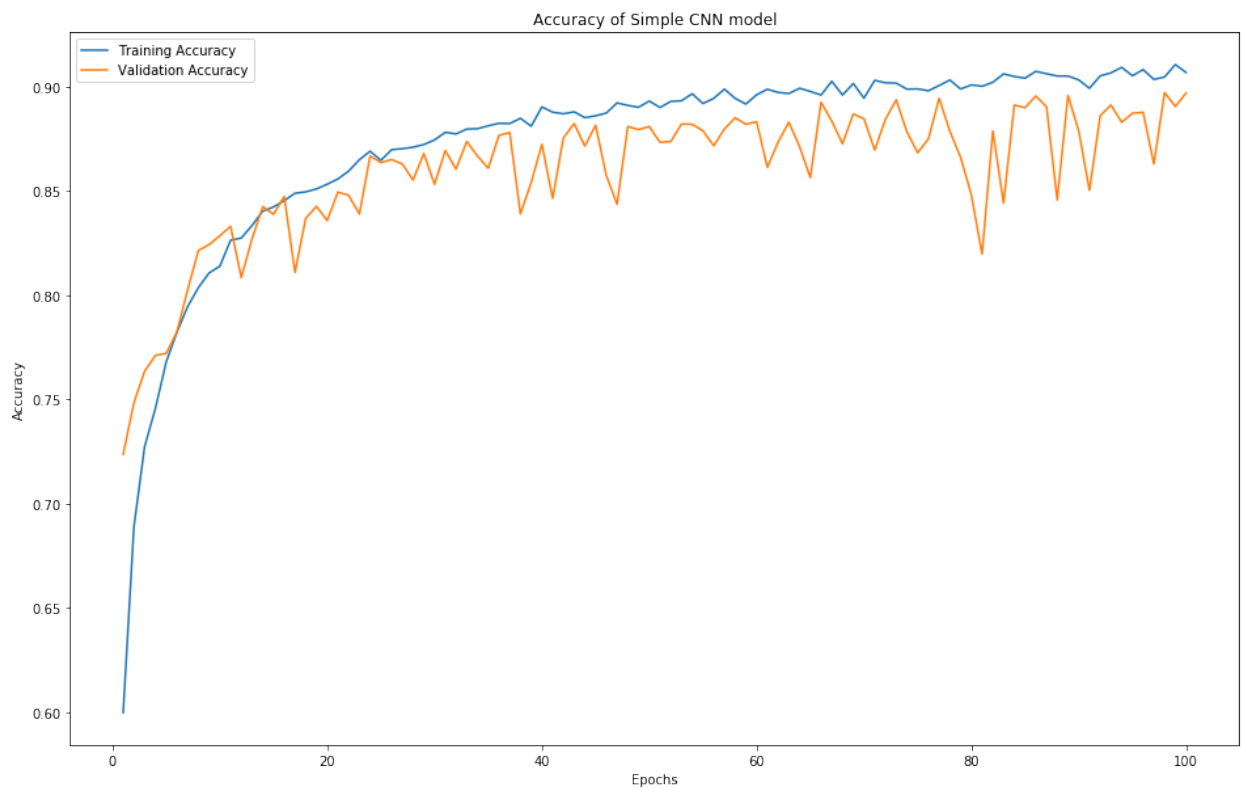
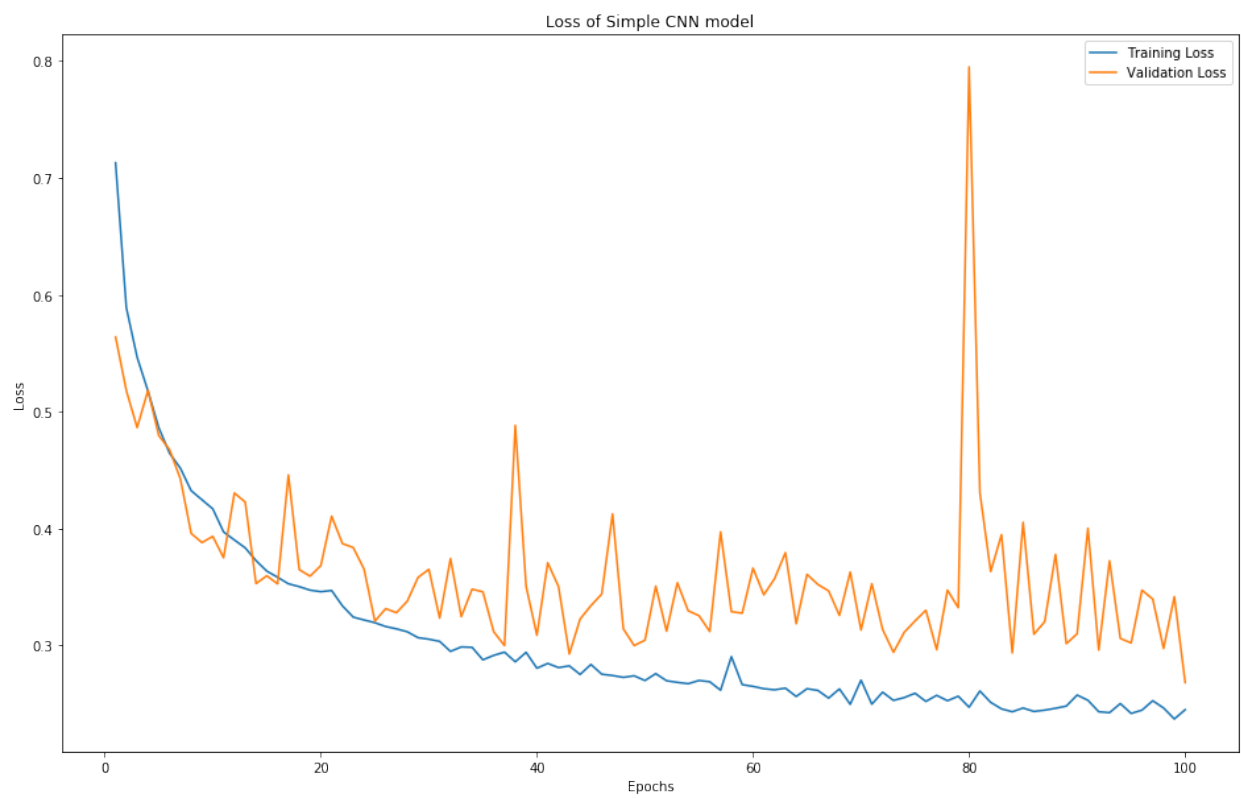
If there is a data folder in the folder where the folder is stored, the folder will already exist. You can rename or change the folder, or replace the "data" in the code.

Part2: Simple\_cnn\_.ipynb

Set the parameters and folder location. It is recommended to set epochs to 1 to test the model integrity. If you do not use gpu acceleration can reduce the batch\_size.

We use the CNN model to deal with it.

For the results:

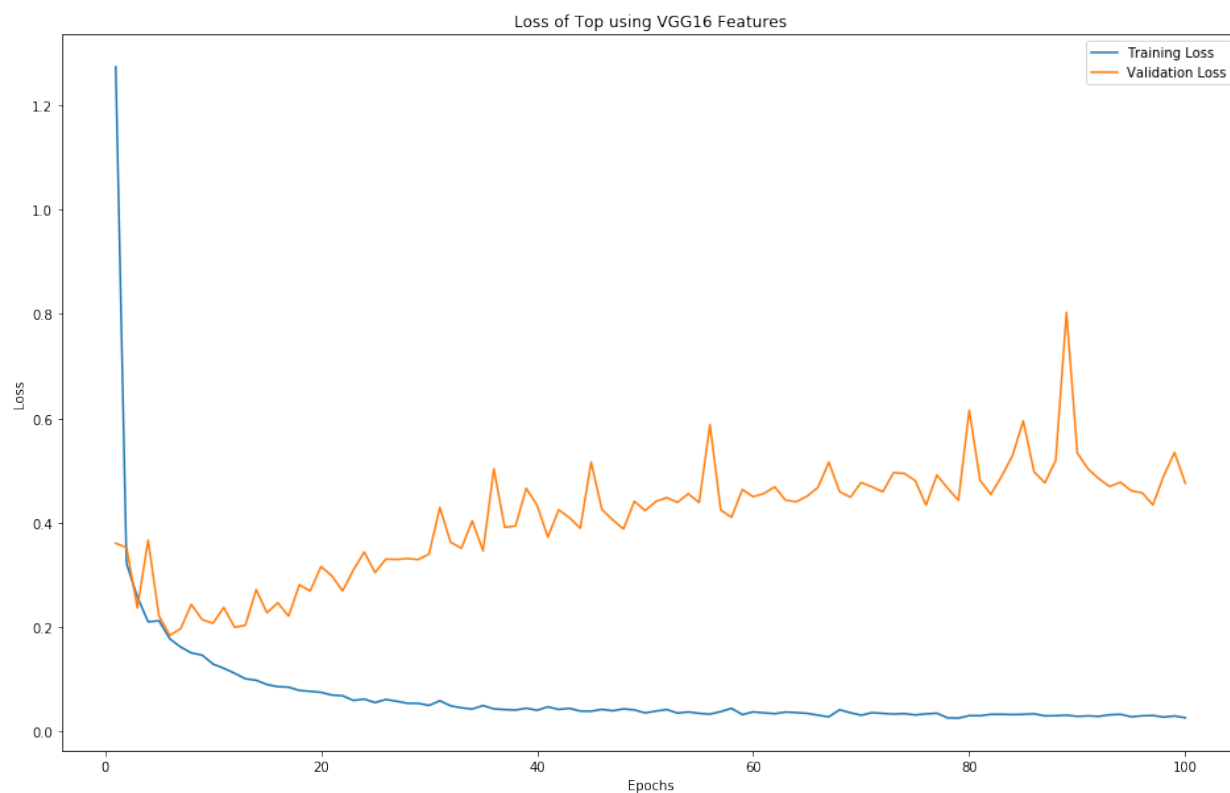


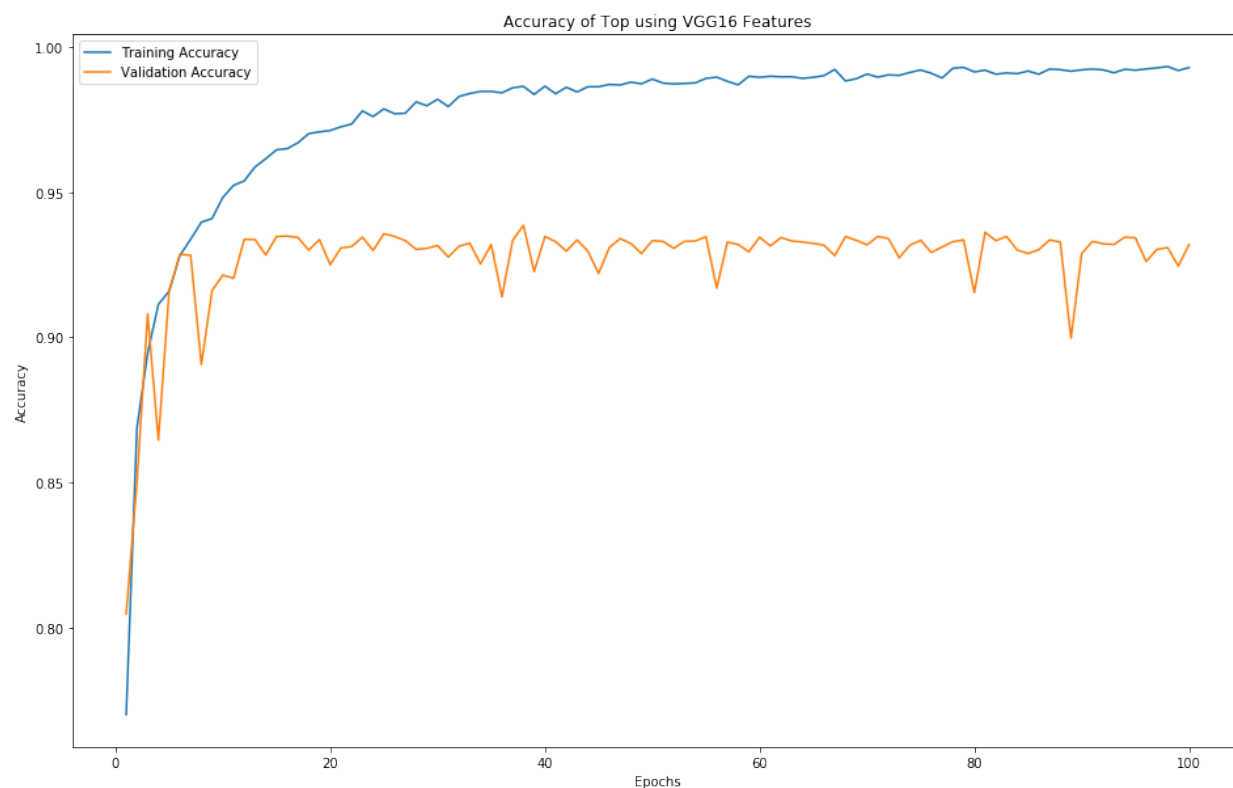
Part3: Vgg16\_top.ipynb

We use the vgg16 features to assess the model.

Layer (type)	Output Shape	Param #
flatten_2 (Flatten)	(None, 25088)	0
dense_3 (Dense)	(None, 256)	6422784
dropout_2 (Dropout)	(None, 256)	0
dense_4 (Dense)	(None, 1)	257

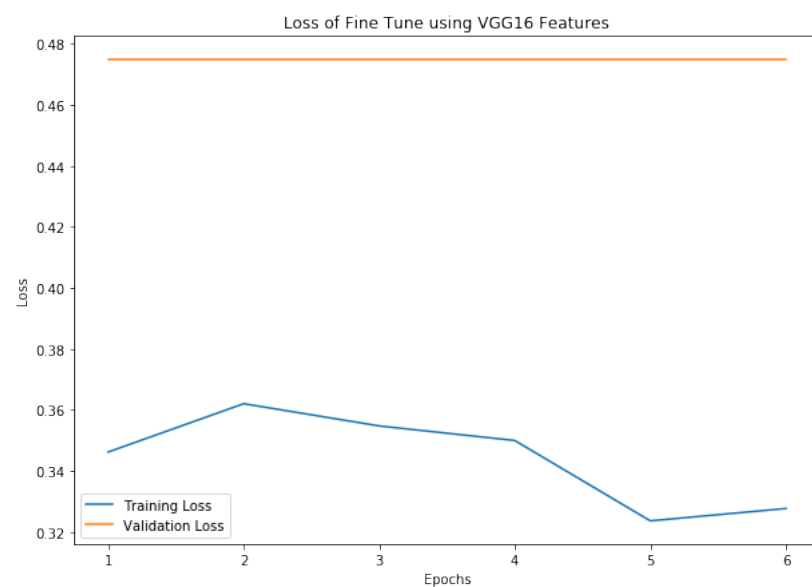
=====  
Total params: 6,423,041  
Trainable params: 6,423,041  
Non-trainable params: 0  
=====

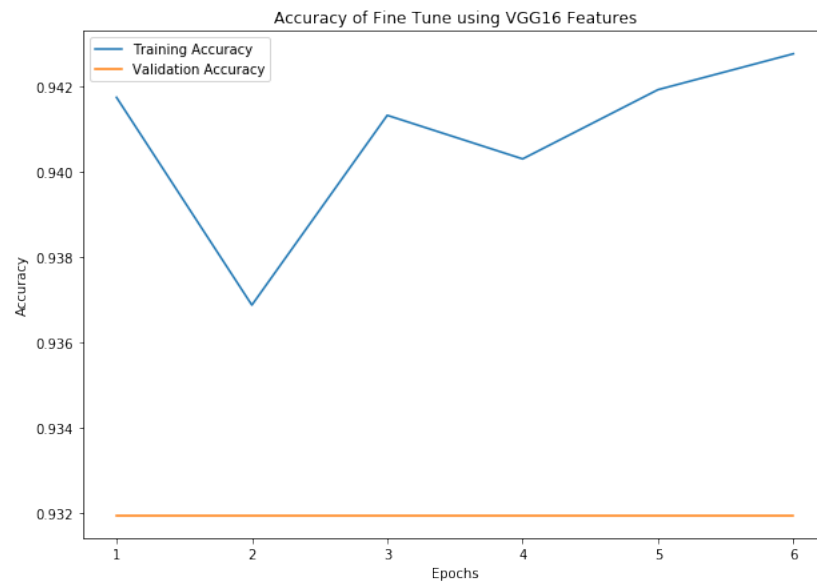




#### Part4: Vgg16\_ finetune.ipynb

We improve the method by finetune using Vgg16



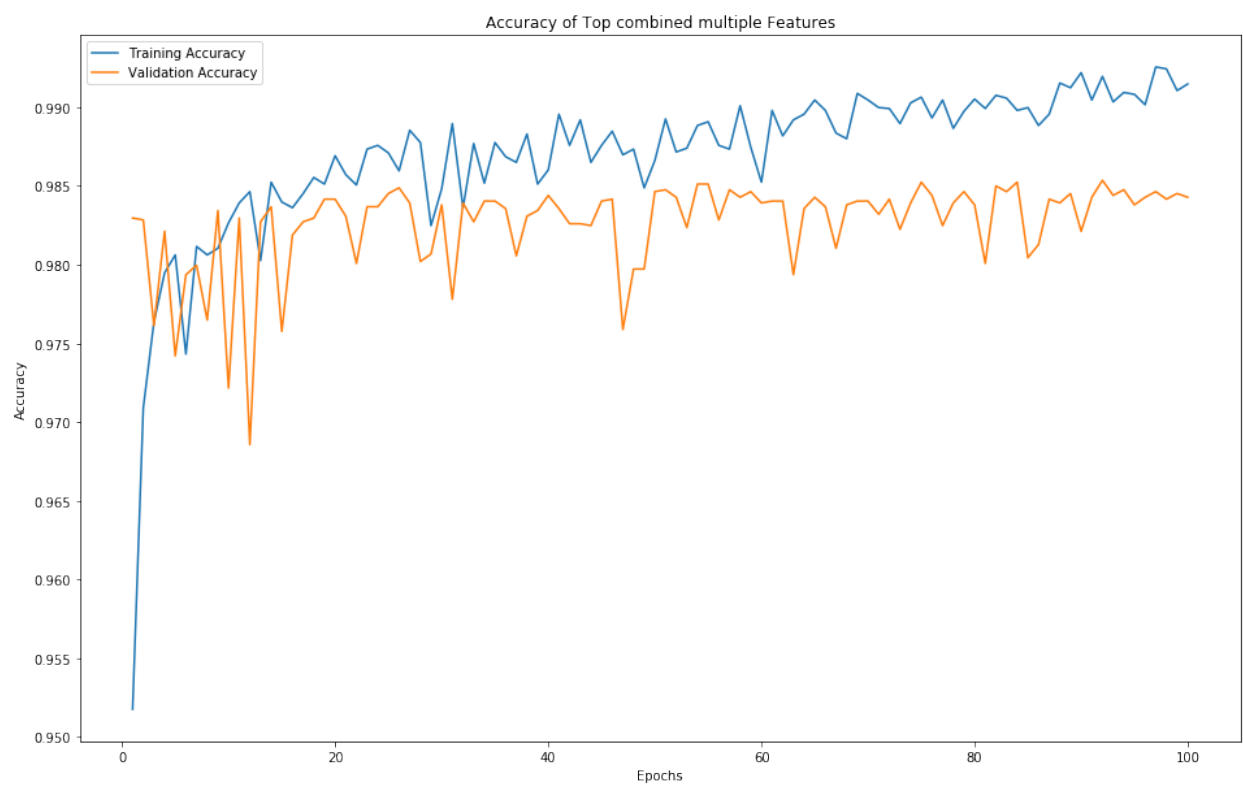
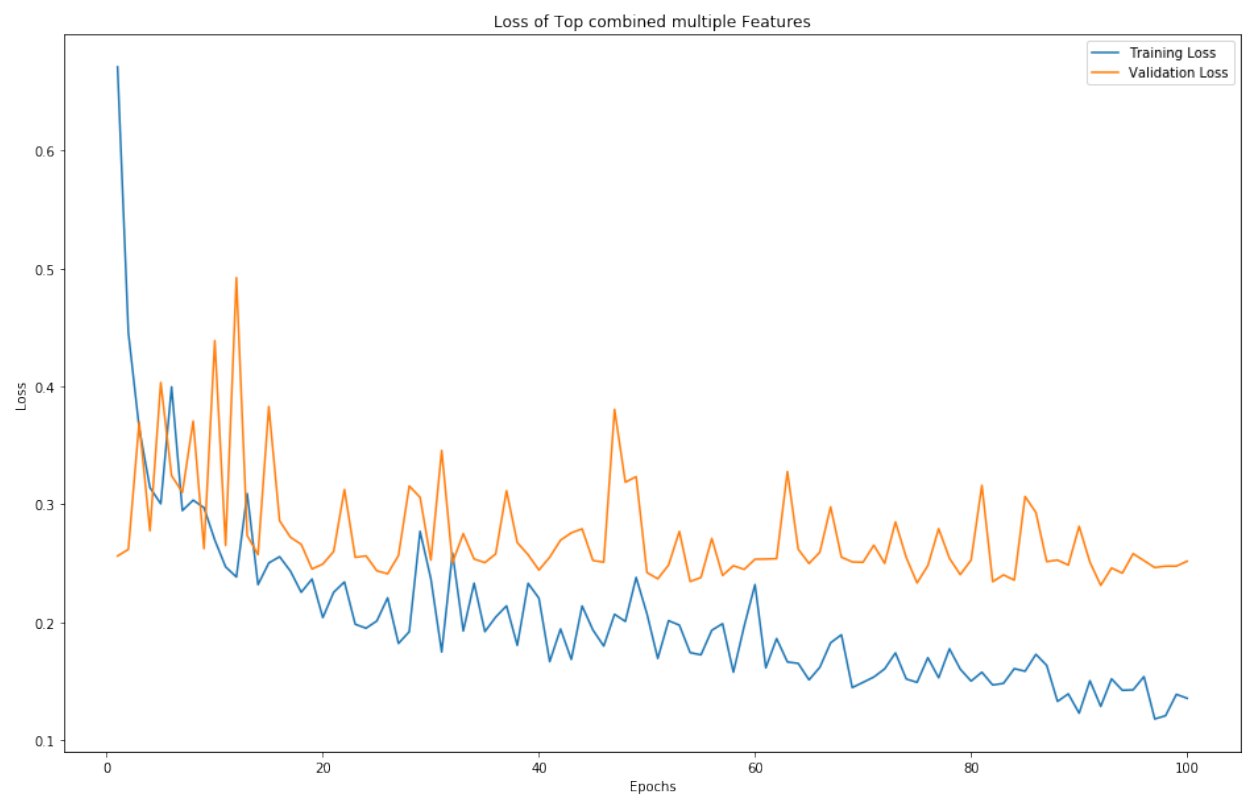


## Part5: Feature\_extraction

The result:

Found 16666 images belonging to 2 classes.  
Found 8334 images belonging to 2 classes.  
Found 12500 images belonging to 1 classes.

## Part6: Combined\_top.ipynb



Extended ideas:

1. Change the training set ratio (1/4)
2. Try 2,3 with other models
3. Try to solve the memory error problem, complete the feature extraction of Xception, and use a combination of five features to train the full connection layer.
4. Deal with the memory problems.