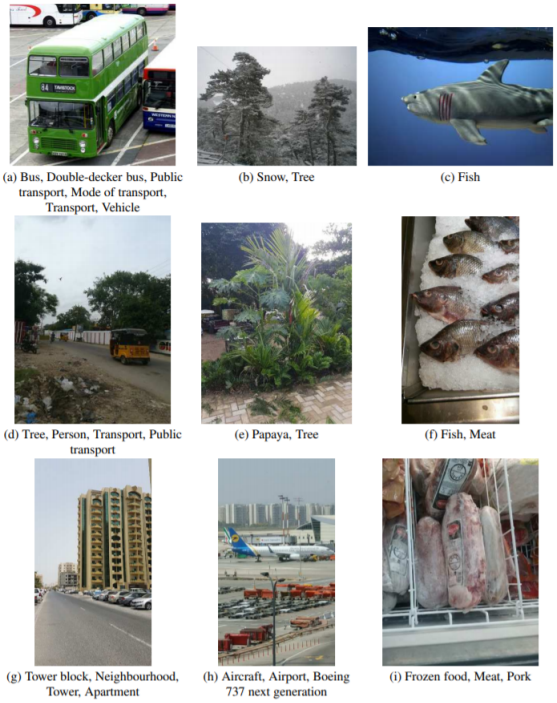
Machine Learning - Final Project

Liam Li*
*George Washington University, zli289@.gwu.edu

***Abstract*—Your abstract goes here.**

***Keywords*—Convolutional neural network, Dense network, Fine tuning, SEnet**



(a) Bus, Double-decker bus, Public transport, Mode of transport, Transport, Vehicle
(b) Snow, Tree
(c) Fish
(d) Tree, Person, Transport, Public transport
(e) Papaya, Tree
(f) Fish, Meat
(g) Tower block, Neighbourhood, Tower, Apartment
(h) Aircraft, Airport, Boeing 737 next generation
(i) Frozen food, Meat, Pork

I. Introduction

Inclusive image challenge was a competition held by google AI on Kaggle. The goal is to develop an image recognition system that are robust enough to blind the spots on training data, and that performs well on test images from different geographic distributions. This competition also was a part of the NIPS 2018 competition track.

II. Problem Statement

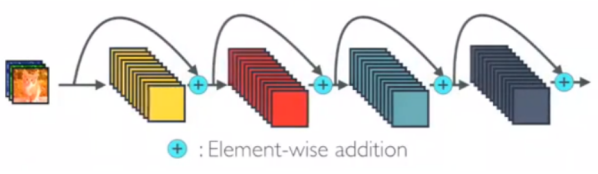
The convolutional neural network is a well-known deep learning algorithm in image recognition area. However, the performance of CNN largely depends on the diversity of the training dataset. Because of culture or geographic variances, images from different regions do not always have much similarities by same labels. For example, trees have different shape based on their location (pines and palms), and people who live in tropical and cold zone have different appearance due to their clothes. Moreover, the shape of a person on an image heavily shifts based on different postures and shooting angles.

Dealing with inclusive images is a common problem for CNN models, because it is difficult to find a dataset that covers all types of images all over the world. The motivation behind this work comes from the novel problem and the approach. A robust model is not only the benefit to all communities across the world, but also able to inspire other areas with similar problems. From my own experience, deep learning is also new area, I hope I will have a better understanding after this project.

III. Related Work

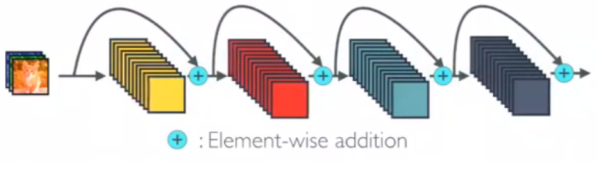
Some of the competitors have achieved great works through the competition, most of them built their models by combing multiple CNN architectures with fine tuning on last layers. [3] It could be a good direction for my own work by reviewing their approaches. The most common used architectures include deep residual networks, densely connected convolutional networks, and squeeze-and-excitation networks.

The deep residual network is a CNN model with a residual framework supporting deeper layers than the original one.[4] The traditional CNN intend to be overfitting as increasing the layers. The authors solved this problem by introducing a deep residual learning block as an additional layer. The result show that the ensemble of residual nets has 3.57% error on the ImageNet test set with 152 layers, compared with the VGG net with 19 layers and higher error. This result won the 1st place on the ILSVRC 2015 classification task. The authors concluded the residual function with reference is easier to optimize than the unreferenced functions and it has higher accuracy on high depth.



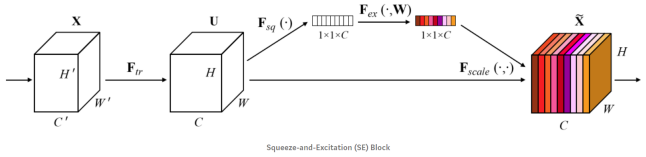
ResNet Concept

In contrast of residual network, the dense convolutional network combines features by concatenating instead of summation.[5] The dense block at each layer received feature maps from all previous layers, which makes the network denser and more efficient because it doesn't need to relearn the redundant feature maps. Another advantage of dense network is that gradients are accessible for each layer. It also deeper the depth of the architecture. After evaluating the model on four datasets (CIFAR-10, CIFAR-100, SVHN, and ImageNet), the authors concluded that dense network needs less parameters and high efficiency than current architectures with comparable accuracy.



ResNet Concept

Squeeze-and-excitation network uses a new architecture unit called squeeze-and-excitation block to improve channel interdependence without many computational expenses.[2] First, the block squeezes the feature maps at each layer into a vector, then the sample-specifics activations govern the excitation of each channel, and the vector are used to regenerate the weights into subsequent layers. One advantage of this new architecture is that it can add existing models easily to improve their performance. The result shows the combination of SE-blocks and Resnet-50 achieved 2.251% top 5 error on the test set, and the authors won the first place in the ILSVRC 2017 classification competition.



Squeeze-and-Excitation (SE) Block

IV. Dataset

The training data of the competition comes from the open images website. It contains more than 1.7 million images with human verified labels(truth ground) machine identified labels (confidence between 0 to 1) from one geographical distribution. The stage 1 test dataset includes 30 thousand images with ground truth labels from other local distribution. The competition also offered 1000 tuning labels from stage 1 for training. Followed by the rule of the competition, other datasets are not allowed to use.

The number of image labels offered by the competition for training is 7178, however, the number of unique labels from training set is more than 18 thousand. Each image has multiple labels. Figure 1 shows a few samples from the dataset. The labels have a hierarchical structure. The higher level labels are more general than the lower levels such as person, clothing, plant. In other word, the hierarchical structure makes higher level labels more robust on different geographical distribution. Consequently, the trained model with higher level labels would produce more general results instead of specific information. Figure 2 shows the most frequent label in training set.

In this project, I use 50 thousand of images from the training subset due to limited training time and computing power, and 10% of the training data are divided into validation set. 1000 images from stage 1 dataset are used to be the test set in order to keep it proportional with the original datasets. Also, 100 tuning labels are used to fine tune the trained model.

For the training classes, I first used 1000 labels from the training set based on the frequency. Since the test set and training set are from different geographical distributions, their labels are hardly to fully match. In order to avoid this problem, I merged the classes from training set and test set before training the model. It makes the final number of classes increase to 1545. However, if the test set is changed, this problem will still happen, or the model need to be retrained by new merged classes.

For the input images, I scaled the image size into 224*224, and the input classes were multi-hot encoded by the training classes.

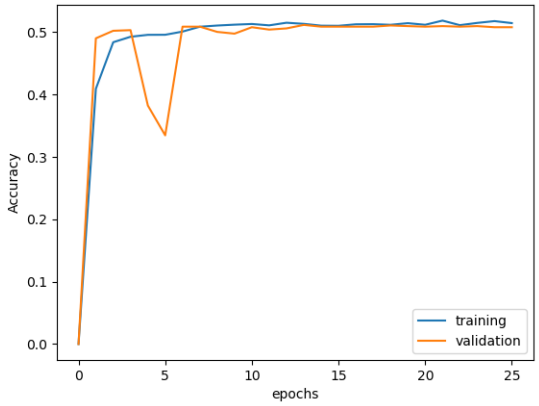
V. Approach

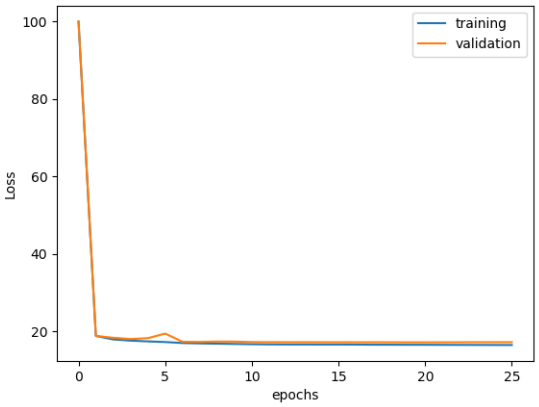
The approach I used is a standard dense network with fine tuning on last layer. During the training, I used the Adam as the optimizer with 0.001 learning rate for 25 epochs, and reduced it when the performance does not improve after each epoch. Moreover, I used checkpoint to store trained weights at improved epoch. The sigma function on last layer were used to produce the probabilities of all classes.

After training model, I use the 100 tuning labels to adapt last layer on the trained CNN model. The purpose of this step is to tune the weights of classes with specific information. The CNN models are able to automatically identify the hierarchical structure of labels, so the lower level classes would be the end of the layers. Fine tuning these classes by frozen most layers except the last one to achieve better performances on a different distribution dataset.

VI. Experiments

The figures below show the accuracy and loss from training set and validation set during 25 epochs.





VII. Results

The final F2 score generated after fine tuning was 0.25.

VIII. Discussion

The accuracy and lost might not be reliable because the general classes are easily to predict such as person, animal. As improvements, there are more than one way to achieve better performances. One approach is data augmentation. It is similar with bagging. It is a strategy that significantly increases the diversity of training data without adding new data. It can be achieved by random crops and random horizontal. Another way is to weight rare samples and labels for the unbalanced training data.[1]

An image of the number "3" in original form and with basic augmentations applied.

IX. Conclusion

Given the constraints of the size of training set and limited computing power, I used a single CNN model with fine tuning on last layer to solve the inclusive images challenge. The final model achieved the F2 score is 0.255. As a comparison, the top score on Kaggle was 0.39 by combing multiple models and weeks for training. It is a motivation for me to continue working on this problem by more researches and multiple methods. Except data augmentation and weight sampling discussed above. Another straightforward approach is combining different models by averaging weights or most vote. It could be achieved by sufficient time or powerful GPUs.

References

[1] M. R. . . W. Z. . . B. Y. . . R. U. . 2, "Learning to reweight examples for robust deep learning," *Science*, vol. 356, no. 6334, pp. 183–186, 2017.

[2] S. A. G. S. E. W. Jie Hu, Li Shen, "Squeeze-and-excitation networks," *Science*, vol. 356, no. 6334, pp. 183–186, 2017.

[3] P. Ostyakov1 and . S. A. C. M. R. S. I. o. M. a. S. P. R. N. O. T. E. Sergey I. Nikolenko1, 2, "Adapting convolutional neural networks for geographical domain shift," *Science*, vol. 356, no. 6334, pp. 183–186, 2017.

[4] K. H. X. Z. S. R. J. Sun, "Deep residual learning for image recognition," *Science*, vol. 356, no. 6334, pp. 183–186, 2015.

[5] G. H. C. U. Z. L. T. U. L. van der Maaten Facebook AI Research Kilian Q. Weinberger Cornell University, "Densely connected convolutional networks," *Science*, vol. 356, no. 6334, pp. 183–186, 2017.