Pig Identification Based on MXNet

Xiatong Li
Department of Electrical Engineering
University of California, Santa Cruz
Santa Cruz, CA 95064
xli239@ucsc.edu

Yuchen Yuan
Department of Computer Engineering
University of California, Santa Cruz
Santa Cruz, CA 95064
yuyuan@ucsc.edu

Diyang Gong
Department of Electrical Engineering
University of California, Santa Cruz
Santa Cruz, CA 95064
digong@ucsc.edu

Yi Ru
Department of Electrical Engineering
University of California, Santa Cruz
Santa Cruz, CA 95064
yiru@ucsc.edu

Abstract

We tried the challenge of pig identification on MXNet, which is the new Amazon's deep learning framework. By using the feature-based transfer learning method with fine-tuning, the features of the pre-trained Convolutional Neural Network(CNN) model were stitched together with the random weight output layer, and then input Target data and update the new fine-tuning model with a lower learning rate by back-propagation strategy. The results show that this method can achieve high enough accuracy in less time and is more efficient than directly constructing a new CNN model. In addition, we also tried data-augmentation methods to scale up the original dataset to improve accuracy.

1 Introduction

Due to the development of the computer hardware, such as GPUs or large-scale distributed cluster [3], the Convolutional Neural Networks(CNN) have made a great success in large-scale image and video recognition [6] [12]. Since face recognition is a big challenge field for computer vision and machine learning, Labeled Faces in the Wild (LFW) [4] is the most famous Face Recognition Test Set today, which was created in 2007. The result shows that the classic face recognition algorithm Eigenface [11] has only a 60% recognition rate on this test set. In the non-depth learning algorithm, the best recognition rate is 96.33% [1]. At present, deep learning can achieve a recognition rate of 99.47% [10]. In this final project, We decide to implement different architecture model of CNN for a new data set of pig for face classification.

Under the traditional framework of machine learning, the task of learning is to learn a classification model based on given sufficient training data, and then use the learned model to classify and predict the test data. However, machine learning algorithms have a key problem in current Internet application research, that is, the large amount of training data in some emerging fields is scarce. With the rapid development of the Internet, the development of Web applications is very fast, a large number of new areas continue to emerge, from traditional news, to web pages, to pictures, to blogs and broadcasts. First of all, traditional machine learning needs to calibrate a large amount of training data in each field, which will consume a great deal of manpower and material resources. Without a large amount of annotation data, many studies and applications related to learning can not be carried out. Second, traditional machine learning assumes that training data and test data are subject to the same data distribution. However, in many cases, this same distributional assumption is not satisfied. What can usually happen if the training data expires. This often requires re-labeling a large amount of training

31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.

data to meet the training needs, but labeling new data is very expensive and requires a lot of manpower and material resources. On the other hand, if there is a large amount of training data under different distributions, it is very wasteful to completely discard these data. How to make rational use of these data is the main problem of transfer learning. Transfer learning of knowledge from existing data can be used to help future learning. The goal of Transfer Learning is to bring together the knowledge learned from one application scenario to help with the learning tasks in a new application scenario. Therefore, transfer learning does not have the same distributional assumptions as traditional machine learning. At present, the work on transfer learning can be divided into three parts: instance-based transfer learning in homogeneous space, feature-based transfer learning in homogeneous space, and transfer learning in heterogeneous space. The research indicates that instance-based Transfer Learning has stronger ability of knowledge transfer, feature-based transfer Learning has a broader knowledge transfer ability, and heterogeneous space transfer has extensive learning and expansion capabilities. These methods have their advantages and disadvantages.

In the new deep learning framework, MXNet, we try to combine the pig identification and transfer learning. The technology of "pig identification" is a kind of biological living body recognition technology and has a wide range of application prospects. It can not only identify pigs but also recognize cattle, sheep and other animals and even dogs and can not only help The farm tracks each pig for daily information management and whole-process traceability, and the technology can also realize the reform of rural financial farming insurance through "pig identification". The development of data agricultural loans enables farmers to use the finance Tools to better carry out breeding production; and in the future, "pig identification" is expected to achieve through the identification of animal facial features to determine the status of pig breeds, pig body posture and movement through the identification to determine the health of pigs.

2 Related Work

Before we start programming and building models to identify pig face, we studied a lot of documents and theses for finding the most efficient methods. Since we have no high-performance calculation platform and limited dataset, we fix our attention on transfer learning and study about its related application and current and potential future direction of research.

The main research direction of transfer learning is a sort of special algorithm called Domain Adaptation and its application in computer vision [2]. It is an effective way to deal with the inconsistency between distribution of train-set and test-set. And it also matches our goal that use limited dataset to identify different types of pig faces. Moreover, there are diverse directions focus on combining Domain Adaptation and different learning methods or representation [7] [9]which are both optimize and improve the performance on deep learning through adapting the source and targeting tensor representations without vectorization as well as a new Domain Generalization(DG) techniques..

3 Method

3.1 Pre-process the Raw Data

Firstly, we need to capture the pig picture in each video. If we get photos frame by frame, it will cause bad redundancy of data. Hence, in this case we set 1 picture per second(which means we capture a picture every 49 frames), and finally we get 1800 pictures in total. To fit the training model, we resize the pictures into 224*224, and for particular model which is inception, we resize the pictures into 299*299.

Figure 1: The dataset is 30 videos of 30 different type of pigs, each video has 1 minute.

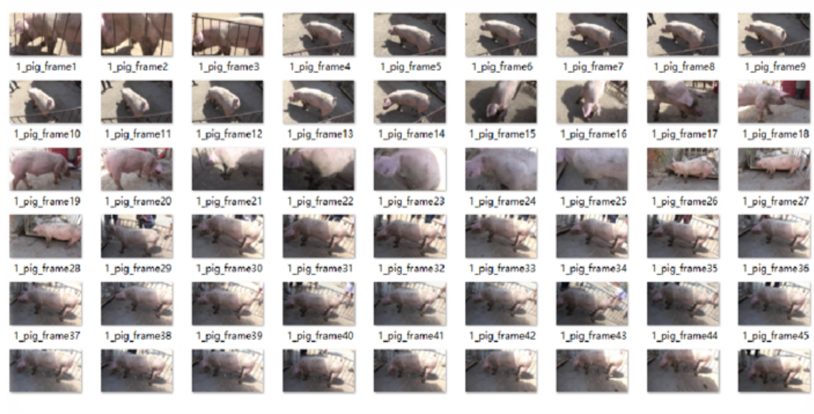


Figure 2: Extracted pictures – in total 1800 pictures

3.2 Data Augmentation

When we collect the data to prepare for the fine tuning model, we will often meet the case where the shortage of classified data occurs. And on the other hand, too small dataset may easily lead to over-fitting of the model. Thus, we need to implement data augmentation to solve this problem, which contains several different data augmentation methods.

(1) Color Jittering: This is an augmentation of the color, with respect to the image brightness, saturation and contrast.

(2) PCA Jittering: First take the mean and standard deviation of the image of all three of the RGB channel, respectively. Then calculate the covariance matrix on the whole training dataset, and do the characteristic decomposition to find the feature vector and feature value, in order to do the PCA Jittering.

(3) Random Scale: This is a scale transformation.

(4) Random Crop: This is to implement the method of random image difference, to crop and resize the image. It actually contains the method of Scale Jittering(which is used by VGG and ResNet model), and the scale and length width ratio enhancement transformation.

(5) Shift: This is a translation transformation.

(6) Rotation/Reflection: This is a rotation or affine transformation.

(7) Noise: This is to use Gaussian noise as well as blurring process.

(8) Label Shuffle: This is to broaden the data which is out of balance in types.

In our project, we use methods of Rotation/Reflection. And this makes our training dataset to be enlarged by 6 times, from 1800 images to 10800 images. By using data augmentation methods, we successfully decrease the loss by 6%.

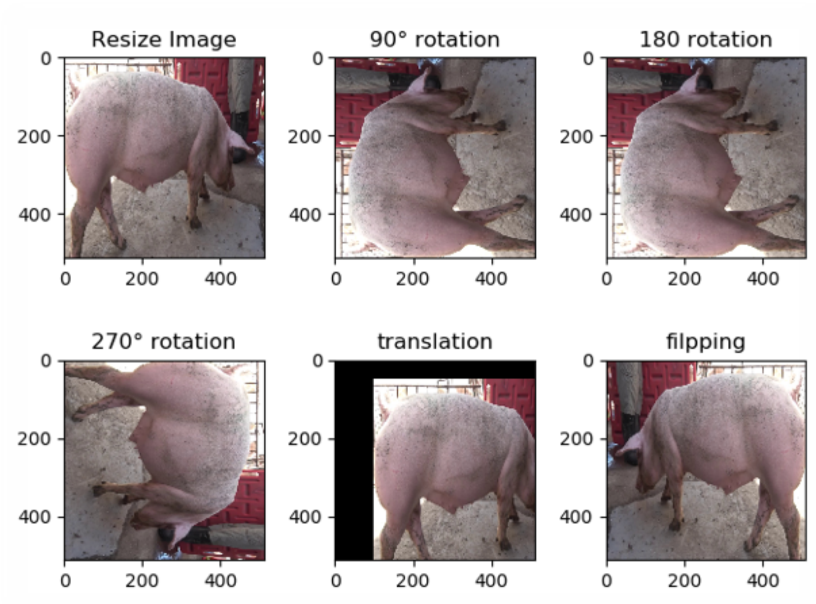


Figure 3: Output images of several data augmentation methods

3.3 Build Fine Tuning Model[8]

After we get the training dataset, the method we use is called fine-tuning, which is a kind of transfer learning based on features.

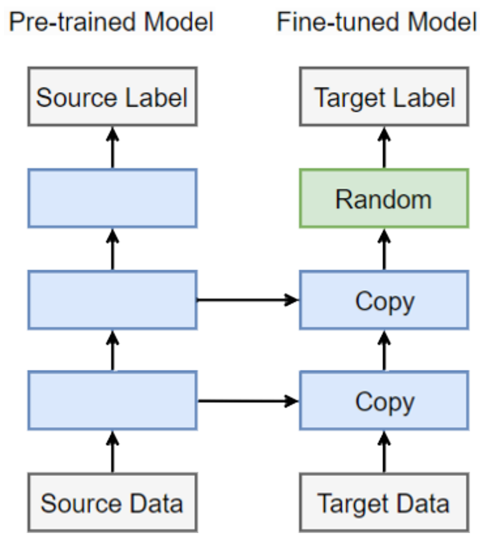
Transfer learning is a new idea in machine learning field. We know that those models trained with big dataset, like ImageNet, are actually wasteful for big trained data. As for those Small datasets, they cannot get good performance by training new model due to lack of data. For the new added data with no label, we need to re-label a large amount of training data to meet the training needs, but labeling new data is very expensive and requires a lot of manpower and material resources.

On the other hand, traditional machine learning assumes that training data and test data are subject to the same data distribution. However, in many cases, this same distributional assumption is not satisfied. So when the training data expires, there's surely nothing to do about it.

And plus, We can spend less time on this model and get better performance. Thus, we use the fine-tuning method as follows.

What we do exactly is that We download the different CNN pre-trained models from MXNet model zoo, which is shown in figure 4(b), then we get the model features, and connect them with a output layer with random weights. Last, we train the new models and choose two kinds of models with best performance(In our case, the Resnet and VGG model with least loss). The whole procedure of the fine-tuning model is shown in figure 4(a).

It is generally believed that the front-end of CNN (near the input picture) extracts basic features such as texture and color. The closer the extracted feature is to the back end, the more advanced features are abstracted and the specific tasks are oriented. So the more common method of fine-tuning is to fix the other parameters, replace the last few layers of the pre-training network, retrain the last few layers of parameters based on the new dataset (the previous layer parameters remain unchanged as feature extractors) With a smaller learning rate of the network as a whole training.



(a) Demonstration of Algorithm

```
resnet34_v2 | 0.4672200890382131
resnet18_v2 | 0.5078153510888418
resnet152_v1 | 0.6772272189458212
resnet50_v1 | 0.7524495720863342
resnet34_v1 | 0.7814714312553406
vgg16_bn | 0.788814187049865 7
vgg19_bn | 0.7992375890413921
vgg11_bn | 0.8299247622489929
resnet18_v1 | 0.8435645103454591
resnet101_v1 | 0.8530370990435282
vgg13_bn | 0.8742475112279257
alexnet | 0.969298283259074
squeezenet1.0 | 0.980705757935842
squeezenet1.1 | 1.0228662292162578
inceptionv3 | 1.0326079527537029
vgg19 | 1.091732641061147
densenet201 | 1.1217530965805054
vgg16 | 1.1551291545232136
densenet169 | 1.1591972510019941
densenet161 | 1.218744158744812
vgg11 | 1.2438867886861165
densenet121 | 1.3429047664006548
vgg13 | 1.4020016590754192
```

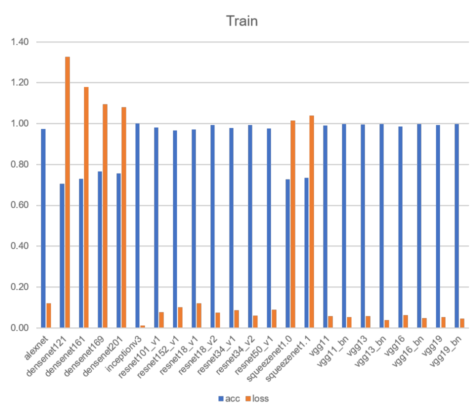
(b) Model Zoo

Figure 4: Demonstration of Algorithm and Model Zoo

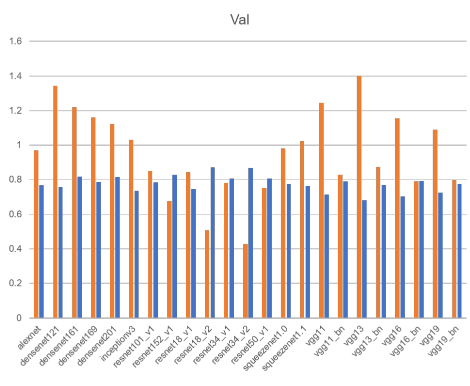
4 Performance and Analysis

After 50 epochs, we get the accuracy and loss of 23 different models, and visualize them as figure 5, with (a) being the training dataset, and (b) the validation dataset.

We can see that, the highest accuracy of validation set is about 90%, and the lowest loss is about 0.46.



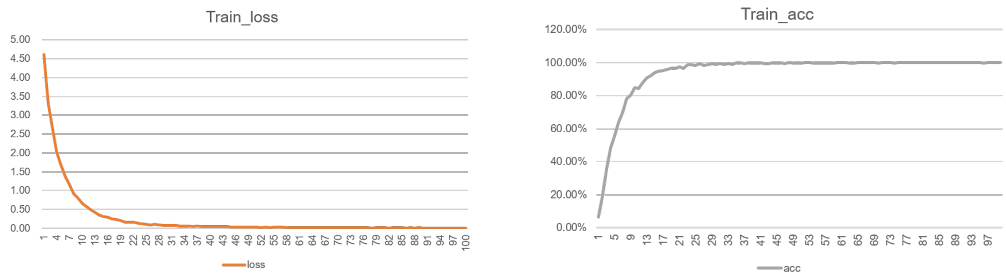
(a) Accuracy and Loss of the training dataset



(b) Accuracy and Loss of the validation dataset

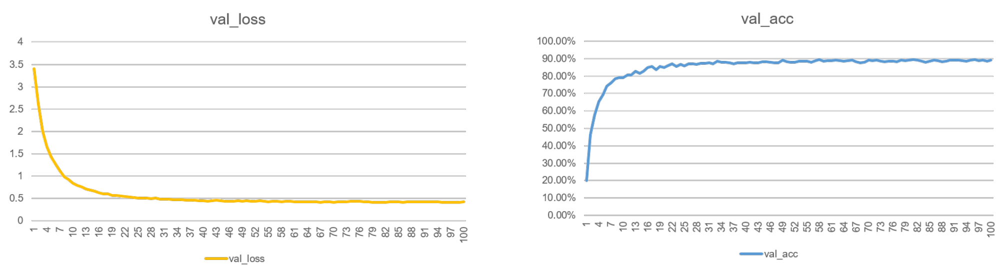
Figure 5: Accuracy and Loss of 23 models

Then we choose two models with lowest loss and highest accuracy, and combine them through transfer learning method. And we visualize the training process with respect to the increase of epochs as figure 6 and 7.



(a) Training dataset loss (b) Training accuracy

Figure 6: Training dataset loss and accuracy



(a) Validation dataset loss (b) Validation dataset accuracy

Figure 7: Validation dataset loss and accuracy

5 Conclusion and Future Work

In this course, we tried to use a new depth learning framework different from tensorflow, MXNet, and used a transfer learning method to identify pigs in this framework. By using existing data and pre-trained model stitching training, the results shows the superiority of this method. In the future work, we can further deal with the picture, for example, through the bounding box, as shown in Figure.

to determine whether there are pigs in our extracted pictures, if there is no pig, you can remove these useless pictures, This will help us to further improve the accuracy. Or we might add some noise to the picture to further expand the data set, making the training data set big enough.

References

[1] Dong Chen, Xudong Cao, Fang Wen, and Jian Sun. Blessing of dimensionality: High-dimensional feature and its efficient compression for face verification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3025–3032, 2013.

[2] G. Csurka. *Domain Adaptation in Computer Vision Applications*. Advances in Computer Vision and Pattern Recognition. Springer International Publishing, 2017.

[3] Jeffrey Dean, Greg S. Corrado, Rajat Monga, Kai Chen, Matthieu Devin, Quoc V. Le, Mark Z. Mao, Marc'Aurelio Ranzato, Andrew Senior, Paul Tucker, Ke Yang, and Andrew Y. Ng. Large scale distributed deep networks. NIPS'12, pages 1223–1231, USA, 2012. Curran Associates Inc.

[4] Gary B Huang, Manu Ramesh, Tamara Berg, and Erik Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical report, Technical Report 07-49, University of Massachusetts, Amherst, 2007.

[5] JD.com Inc. Jdd discovery competition.

[6] Alex Krizhevsky, Sutskever, Ilya, Hinton, and Geoffrey E. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[7] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Learning to generalize: Meta-learning for domain generalization. *arXiv preprint arXiv:1710.03463*, 2017.

[8] Zack Chase Lipton. Deep learning - the straight dope.

[9] Hao Lu, Lei Zhang, Zhiguo Cao, Wei Wei, Ke Xian, Chunhua Shen, and Anton van den Hengel. When unsupervised domain adaptation meets tensor representations. *arXiv preprint arXiv:1707.05956*, 2017.

[10] Yi Sun, Xiaogang Wang, and Xiaoou Tang. Deeply learned face representations are sparse, selective, and robust. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2892–2900, 2015.

[11] Matthew Turk and Alex Pentland. Eigenfaces for recognition. *Journal of cognitive neuroscience*, 3(1):71–86, 1991.

[12] Matthew D. Zeiler and Rob Fergus. *Visualizing and Understanding Convolutional Networks*, pages 818–833. Springer International Publishing, Cham, 2014.