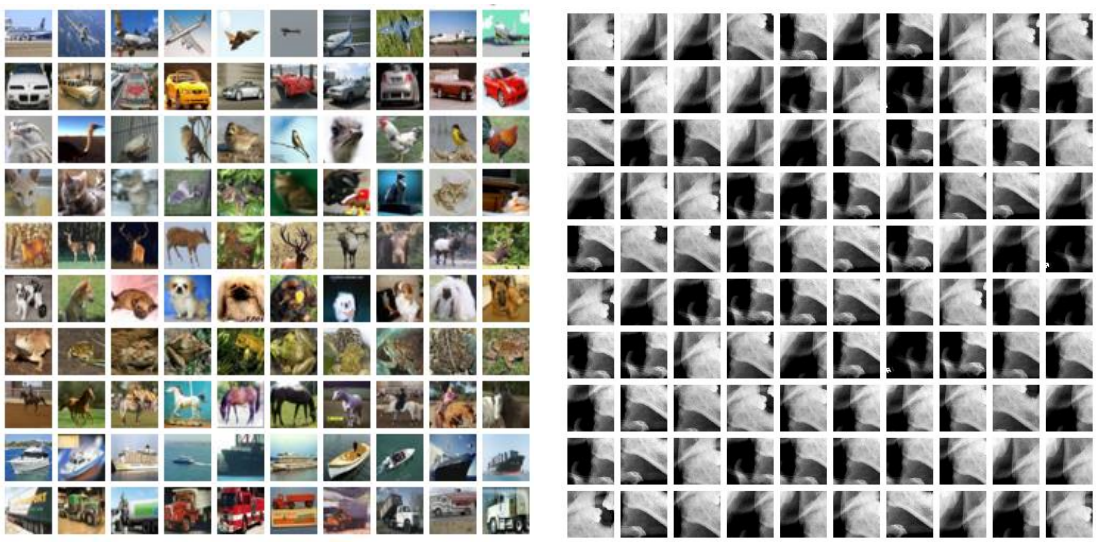
Transfer ImageNet for Medical Image Analyzing Using Unsupervised Learning

Ning Wang and Peng Chu

***Abstract*—Convolutional Neural Network (CNN) is powerful tool for image analyzing. It has demonstrated its success in multiple tasks. But requirement of large amount of annotated samples prohibits its widely using in medical image analyzing. This project explores using unsupervised convolutional auto-encoder (CAE) for fine-tuning CNN trained on ImageNet for medical images analyzing. The fine-tuned CNN then is test on a task for landmark estimation. Compared with CNN without unsupervised fine-tuning, CNN adjusted by CAE achieves 22% less average error on four landmarks.**

I. Introduction

Deep neural network recently becomes a popular way for learning knowledge from massive annotated database. It shows great success in multiple tasks such as image classification and playing the board game Go. Specifically, convolutional neural network (CNN) which is believed to have mimicked the mammal visual vortex is the most widely used framework and achieves stat-of-art performance on the ImageNet classification [1] and AlphaGo project [2].



(a) (b)

Figure 1: Image samples from (a) CIFAR10 and (b) Dental X-ray images.

The CNN is usually trained on gradient descent algorithm which is a supervised process. Thus annotated date is require for the training process. A popular CNN configuration for ImageNet classification named 'VGG-16' has 138 million free parameters for training [3]. Fortunately, ImageNet also has over 10 million annotated images [4]. But for ordinary task, there is usually no such huge amount of annotated database for training. A popular way to solve this problem is to transfer knowledge learned on ImageNet to other domain by using small amount of annotated date to fine-tuning the network learned on ImageNet [6]. It shows very promising results on a lot of domains [5]. But there are several problems for medical images. The first challenge is that the difference between ordinary images and medical images is huge. As an example, figure shows the image samples from CIFAR10 and image samples from dental X-ray images. We can see not only the color, but also objectiveness in the two domains are also very different. Thus the mount of images need to fine-tuning a CNN trained on ordinary image set to medical image is very large. The second problem is that it is very expensive to gather annotated medical. Since in most domains, annotation can be performed by un-trained ordinary people, while in medical, the accurate annotation must be achieved by well-trained doctors.

In this project, we propose an idea to accommodate unsupervised learning strategy to reduce the number of required training samples in fine-tuning a CNN learned on ImageNet to medical image domain.

II. Method

There are multiple ways to achieve transfer learning. The most common way is to only exploit the knowledge shares in two domain, while re-train the task specific part. For example, the popular way to transfer CNN learned on ImageNet is to keep its convolutional part which is the learned filter bank working on all images, while re-train its full-connected part which is task specified.

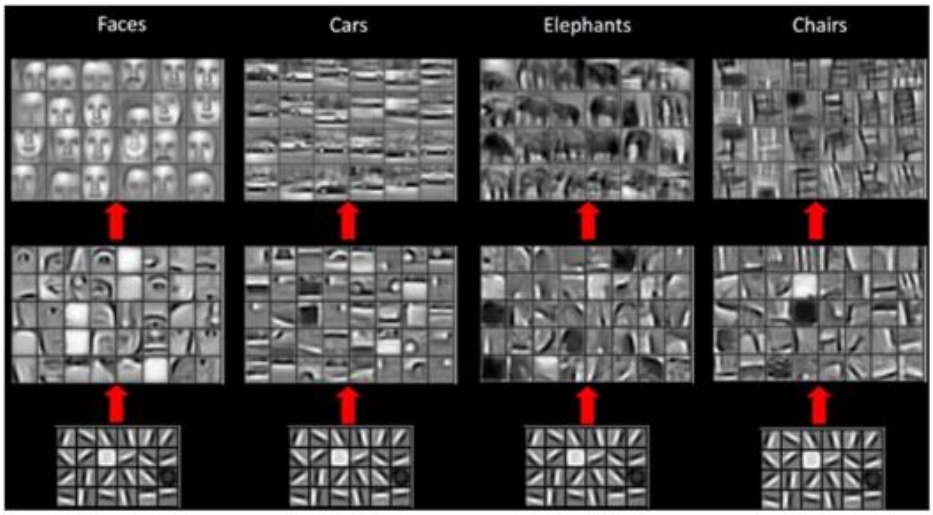


Figure 2: CNN shares low level feature filters.[8]

Instead of this directly transfer, there is also indirectly transfer method. The idea is that if we want to achieve a relative simple task, we can train the model on the same knowledge but a more difficult task. Then the model can be transferred to the target task through relative small adjustment, and more important usually achieving a better performance.

Unsupervised transfer is based on the similar idea. We augment the un-annotated data in multiple ways. Then annotate the data with its augment method. The first transfer task is to fine-tuning the CNN learned on ImageNet to classify the augment method using the augmented data. Then in the second transfer procedure, classification task specified part is removed or partly removed and replaced with new network. Then the new CNN is re-trained on the target task using doctor annotated data. In those process, the first transferring process transfer the ImageNet into medical image domain on a relative hard task. The second transferring process just use a little annotated data to transfer the CNN already trained medical image to a relative simple task.

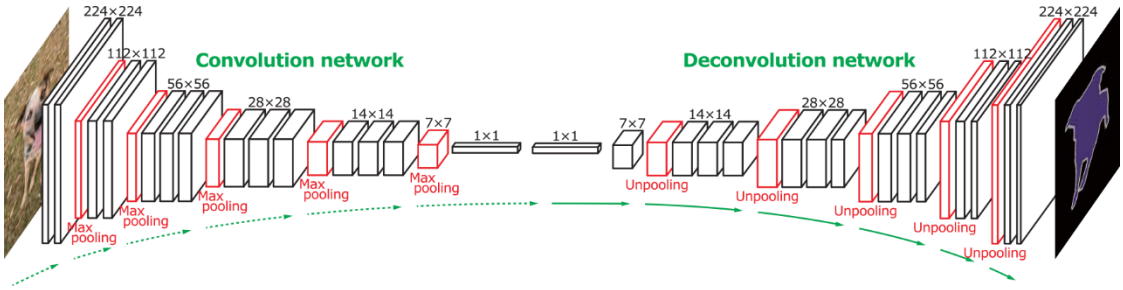


Figure 3: Illustration for convolutional auto encoder-decoder.[9]

Through this procedure, only in the second transfer step, we need to use the doctors annotated data, while in the first step, the data from the same domain but without annotation is enough for training. And since the target task usually is simpler than in the first step, a better performance of the second task is guaranteed.

III. Dataset and Setting

We use a batch of dental X-ray images to evaluate our proposed method. The task of the dataset is to estimate 8 landmarks points on each images. Due to their spatial symmetry, we mirror the right side of the images to left side to increase our dataset. So far we have 108 images, 54 of which has 8 landmarks manually annotated by proficient doctor, while the rests are not. Therefore, after mirror, we have 216 images and 108 have landmarks annotated.

The CNN configuration used in our project is based AlexNet. Its network configuration is shown below. It contains 5 convolutional layers and 3 fully connected layers. CNN framework is based on CAFFE.

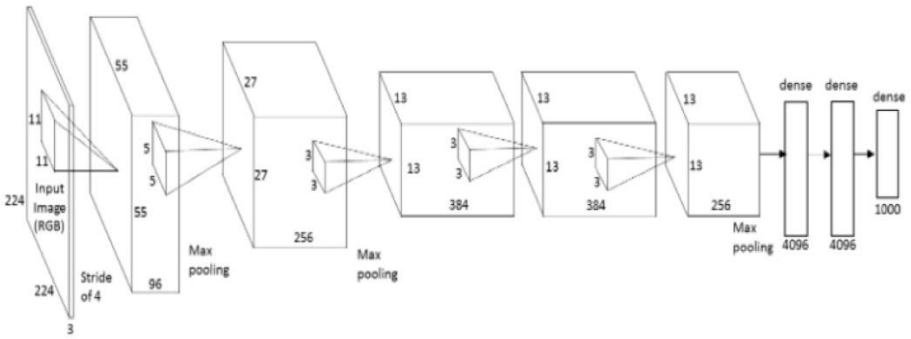
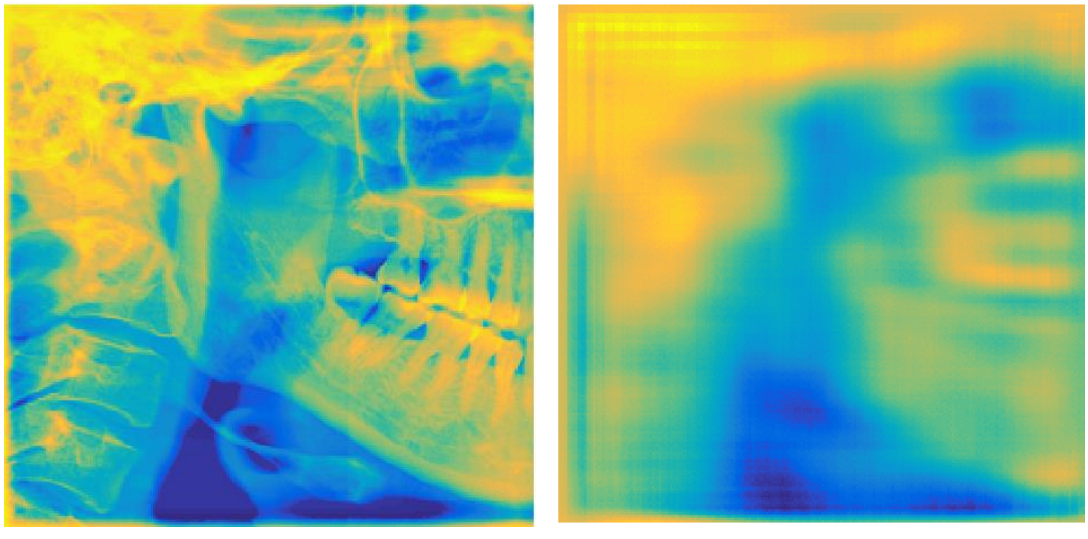


Figure 4: Network configuration for AlexNet. [1]

We first use convolutional auto encoder-decoder (CAE) to performance unsupervised fine-tuning on the AlexNet using all 216 dental images. The mechanism for CAE is that the input images will first be encoded into a low dimension vector, then the exactly mirror of the encoder network will present as decoder to decode the low dimension vector into the original images. Thus the input images itself can also act as the training label to create reference and be used to calculate loss. The training process of CAE is an unsupervised. In our work, the CAE network configuration is just a mirror the AlexNet, which also will be used as landmark detector, with both part sharing the 1×1000 dimension FC8 layer. In detail, left half side is cut from each training images and resize to 224×224. We 1/10 of all images as test samples, while the rests will be training samples. The first 5 convolutional layers in CAE are initialized using the weighted from AlexNet trained on ImageNet with top-1 error rate 37.5% on ILSVRC-2012 test set. The unsupervised fine-tuning results are shown below. For each image, it is first encoded into 1×1000 vector. The 1×1000 run into the decoder part to retrieve the original image.

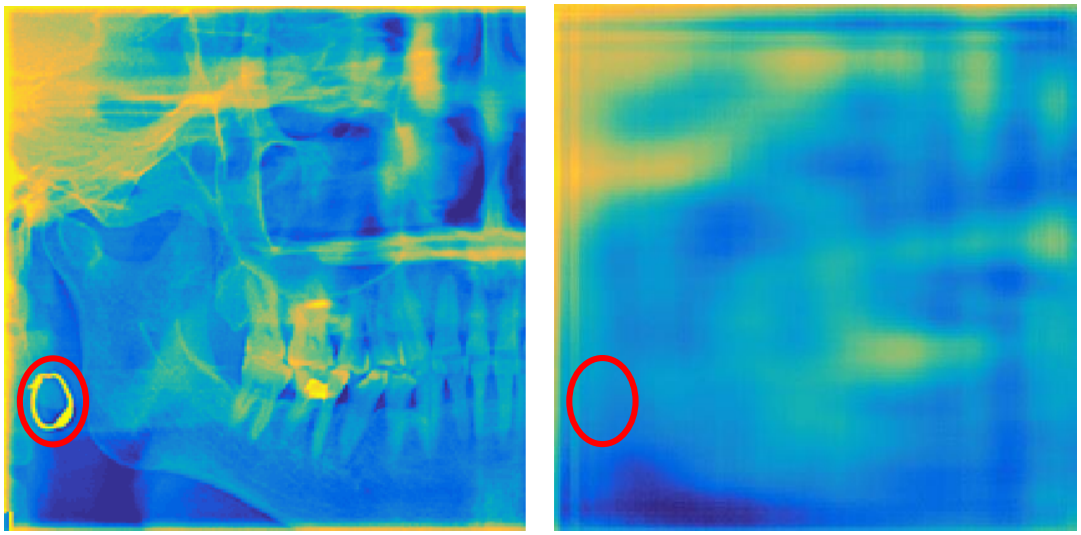


(a) (b)

Figure 5: CAE results (a) Input image and (b) Reconstructed image.

IV. Results

In this section, we first look at the results from CAE. Two input and reconstructed images pairs are shown in Fig. 5 and Fig. 6.Through the CAE, we can see, the detail information of each images may be lost. But the overall shape and key regions of image are conserved. Another benefit of CAE is that the non-relevant items in images are also filtered out. As shown in the images below. The ear ring in the image is non-relevant, so it will not be rebuilt in the decoding process.

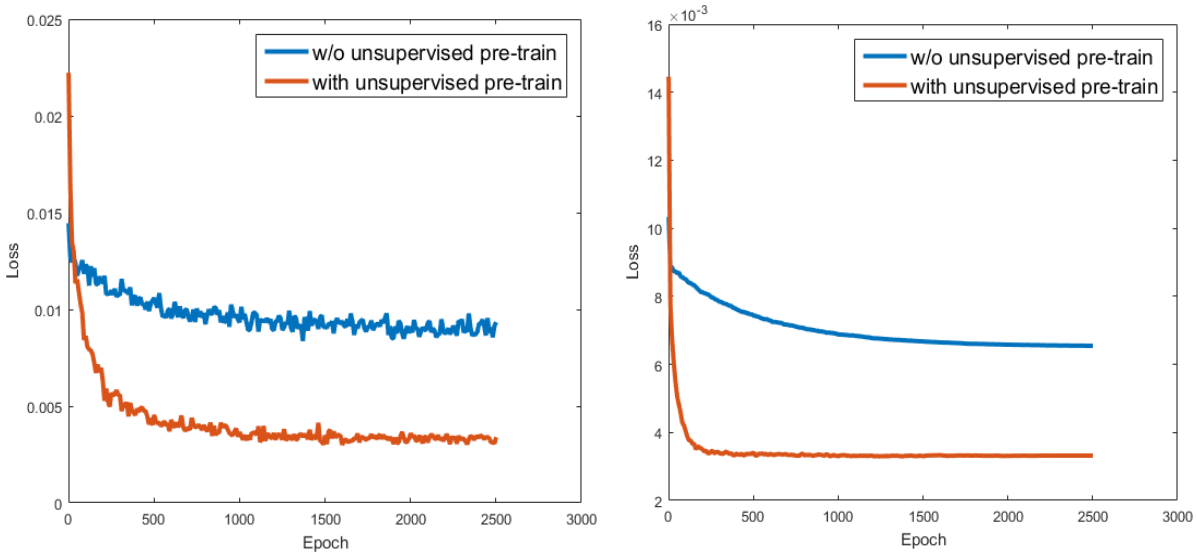


(a) (b)

Figure 6: CAE results (a) Input image and (b) Reconstructed image.

In the next steps, the weights of first 5 convolutional layers and weights of the first fully connected layers are stored for the detection part.

We use a regression CNN to handle the 4 landmarks detection problem. We input the images at bottom of the network, the regression CNN will output the 8 coordinates of the 4 landmarks points ($4 \times (x, y)$). We still use the AlexNet with replacing the softmaxloss layer at top by a Euclidean loss layer. Accordingly, the output of last fully convolutional layers is also changed to 8 instead of 1000. The 1/5 of all annotated images are taken as test samples, while the rests are used as training samples. The weights of first 5 convolutional layers is initialized using the weights directly from the trained AlexNet on ImageNet or the weights from our CAE which represents first using unsupervised fine-tuning. The learning curves are shown below.



(a) (b)

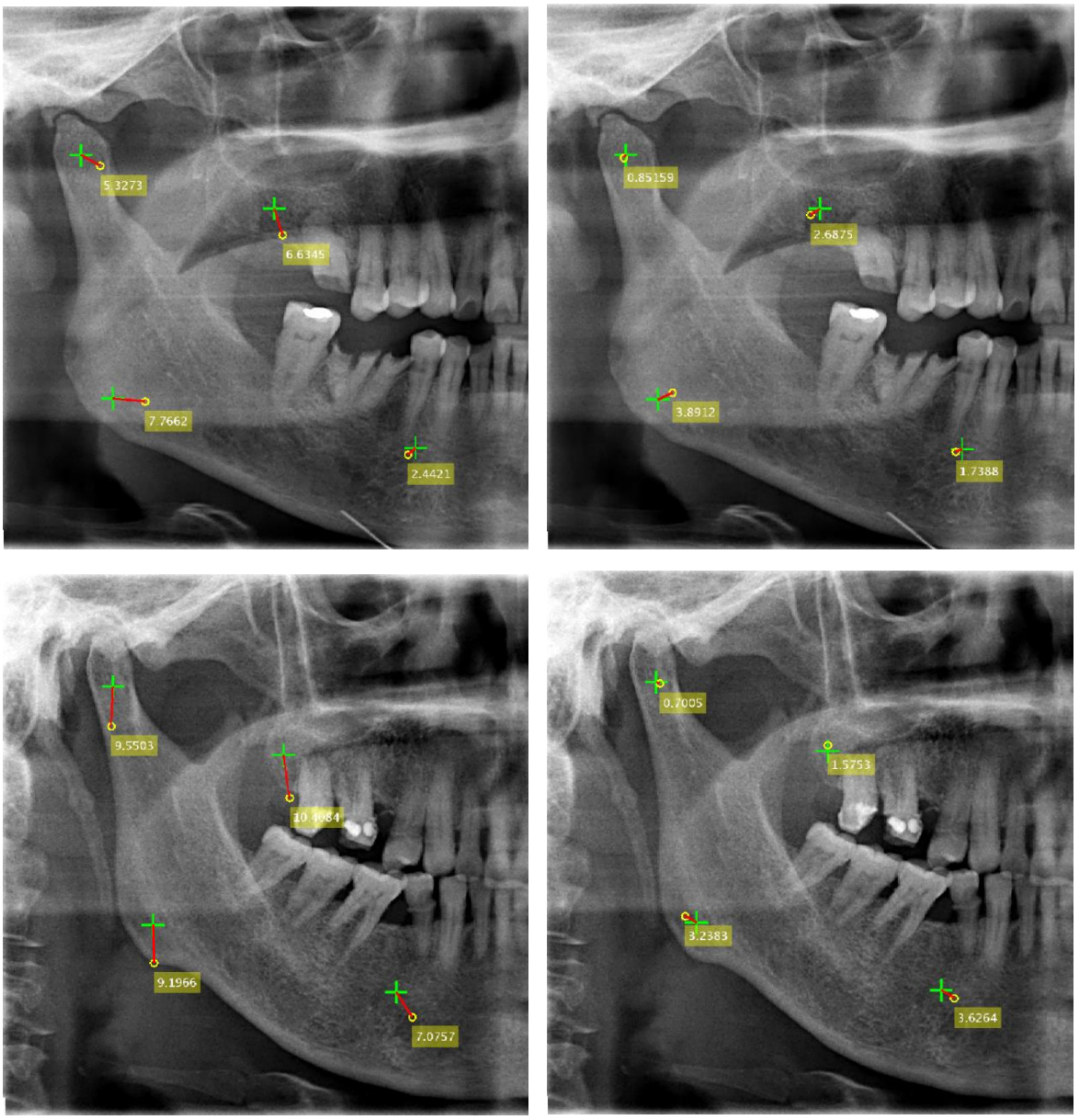
Figure 7: Learning curves (a) Train and (b) Validation.

From the curves, we can see, using unsupervised fine-tuning pre-train, the final loss is as low as 0.003593 while directly fine-tuning only achieved a loss of 0.006545. Using the trained networks for the detection task, it gives the error rate as follows. The error is define as mean distance between the detected landmarks and the annotated landmarks in millimeter. We also calculate the standard deviation of each results.

Table I: Detection error (*mm*)

Landmarks	1	2	3	4
without unsupervised fine-tuning	6.6244 ± 2.9509	8.0094 ± 5.4466	7.6486 ± 4.6781	6.9465 ± 3.5296
with unsupervised fine-tuning	**4.2447 ± 3.1350**	**6.7628 ± 5.5224**	**6.6757 ± 4.9623**	**3.9234 ± 1.9063**

The error rate of using unsupervised fine-tuning pre-train is much lower than without using. For landmark 4, the former is almost 50% lower the latter. For illustration, we draw the two results from each of the two approach.

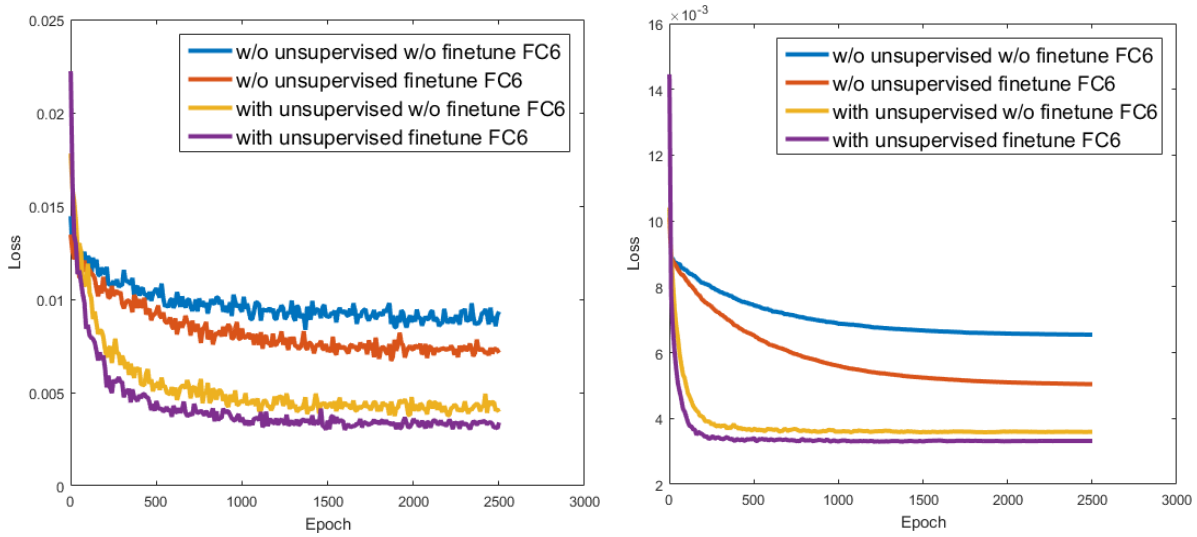


(a) (b)

Figure 8: Detection results (a) without unsupervised fine-tuning (b) with unsupervised fine-tuning.

In this part, we want to explore the necessity to fine-tuning the first fully connected layer. We have known that the first FC layer is the largest layer which contains the most weights in the whole network. For Alexnet, the first FC layer (FC6) contains 6×6×256×4096 = 38M weights which is more than 60% of the weights in the whole network (61M). After our unsupervised fine-tuning, the first FC layer is already trained on the medical related task. We want to know whether the pre-train first FC is compatible with the afterwards landmark detection task, since the landmark detection also bases on the same set of images. For the approaches with or without

using unsupervised fine-tuning pre-train, we add another parameter which indicates whether we fine-tuning the first FC layer. The learning curves and detection results are shown below.



(a) (b)

Figure 9: Learning curves (a) Train and (b) Validation.

Table II: Detection error (*mm*)

Landmarks	1	2	3	4
w/o unsupervised fine-tuning w/o fine-tuning first FC	7.3520 ± 3.4258	8.6705 ± 5.3689	8.3074 ± 4.3578	7.4930 ± 4.0606
w/o unsupervised fine-tuning but fine-tuning first FC	6.6244 ± 2.9509	8.0094 ± 5.4466	7.6486 ± 4.6781	6.9465 ± 3.5296
with unsupervised fine-tuning w/o fine-tuning first FC	5.0618 ± 2.8706	6.9777 ± 5.8913	6.6981 ± 4.9015	5.0999 ± 2.3172
with unsupervised fine-tuning and fine-tuning first FC	**4.2447 ± 3.1350**	**6.7628 ± 5.5224**	**6.6757 ± 4.9623**	**3.9234 ± 1.9063**

Results with FC6 fine-tuning are all better than their control pair where FC6 is not fine-tuned. That rule also works for the methods using unsupervised fine-tuning pre-train. Thus we can conclude that the first fully connected layer is very task specified. Even using the same dataset, if the task is different, it is necessary to re-train or fine-tuning it on the target task.

V. Conclusion

In this project, we use CAE for transferring knowledge learned on ImageNet for medical image analyzing. The CNN after unsupervised fine-tuning then is tested for training a landmark

detector on dental X-ray. Our method gains average 5.374 *mm* error on four landmarks, which is 22% less than the results using CNN directly using knowledge learned on ImageNet.

VI. Reference

[1]. Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." Advances in neural information processing systems. 2012.

[2]. Silver, David, et al. "Mastering the game of Go with deep neural networks and tree search." Nature 529.7587 (2016): 484-489.

[3]. Simonyan, Karen, and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition." arXiv preprint arXiv:1409.1556 (2014).

[4]. ImageNet, http://www.image-net.org/.

[5]. Razavian, Ali, et al. "CNN features off-the-shelf: an astounding baseline for recognition." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. 2014.

[6]. Maxime Oquab, Leon Bottou, Ivan Laptev, and Josef Sivic, "Learning and Transferring Mid-Level Image Representations using Convolutional Neural Networks", CVPR 2014.

[7]. Alexey Dosovitskiy, Jost Tobias Springenberg, Martin Riedmiller, and Thomas Brox, "Discriminative Unsupervised Feature Learning with Convolutional Neural Networks", NIPS 2014.

[8]. Updated with Google's TensorFlow: Artificial Intelligence, Neural Networks, and Deep Learning, https://kimschmidtsbrain.com/2015/10/29/artificial-intelligence-neural-networks-and-deep-learning/.

[9]. Noh, Hyeonwoo, Seunghoon Hong, and Bohyung Han. "Learning deconvolution network for semantic segmentation." Proceedings of the IEEE International Conference on Computer Vision. 2015.

[10]. Masci, Jonathan, et al. "Stacked convolutional auto-encoders for hierarchical feature extraction." Artificial Neural Networks and Machine Learning–ICANN 2011. Springer Berlin Heidelberg, 2011. 52-59.

[11].

[12]. Nguyen Anh, Jason Yosinski, and Jeff Clune. "Deep neural networks are easily fooled: High confidence predictions for unrecognizable images", CVPR 2015.

[13]. Alex, K., Stskever, I., & Hinton, G. (2012). ImageNet Classification with Deep Convolutional Neural Networks. Advances in Neural Information Processing Systems 25, 1097-1105.