Evrişimli Yapay Sinir Ağları Kullanarak Leopar İçeren Fotoğrafların Bulunması
Detecting Photos With Leopards Using Convolutional Neural Networks

Semih Orhan, Yalın Baştanlar
Department of Computer Engineering, Izmir Institute of Technology, Izmir, Turkey
semihorhan@yandex.com, yalinbastanlar@iyte.edu.tr

***Özetçe* —Son yıllarda derin öğrenme yöntemi kullanılarak bilgisayarlı görü problemlerinin çözümü konusunda birçok başarı elde edilmiştir. Özellikle, evrişimli yapay sinir ağları kullanan ve resim sınıflandırılması konusunda yüksek başarılar elde edebilen birçok model oluşturulmuştur. Bu modellerden bazıları AlexNet, VGG19, GoogleNet ve ResNet'dir. Fotokapanlar, hayvan görüntüleme için doğaya yerleştirilen ve hareket algılandığında kayıt yapan kameralardır. Fotokapanlar araziden toplandıktan sonra, oluşan fotoğraf kümesi içerisinden belli türdeki hayvanların seçilme işlemi oldukça zaman almaktadır. Çalışmamızda, bu işlemi otomatik olarak yapabilmek için farklı sayıda katmana sahip bazı derin öğrenme modelleri kullanılmış ve fotokapan fotoğraflarından leopar resimleri ayırt edilmeye çalışılmıştır. Bu amaçla değişik eğitim ve test setleri hazırlanmış, seçilen modellerin performansları kaydedilmiştir. 20+ katmanlı ResNet modelleri ile çok yüksek bir başarı elde edildiği, ayrıca daha yüksek çözünürlüklü imgeler ile eğitilen modellerin daha başarılı olduğu gözlenmiştir.**

Anahtar Kelimeler—Derin öğrenme, fotokapan fotoğrafları, leopar resimlerinin sınıflandırılması.

***Abstract*—In recent years, Deep Learning has shown great performance in different problems of computer vision. Its popularity has increased year by year. A lot of models employing convolutional layers have been proposed which have good performance at image classification task. For instance; AlexNet, VGG19, GoogleNet and ResNet. Camera-traps are photo-cameras that are placed in the pathways of animals and used for wild life surveillance. Classification of camera-trap photos that are gathered from terrain takes so much time. To deal with this problem, different convolutional neural network models with varying number of layers were implemented to classify leopard images. Selected models were trained on the training set and their performance on the test set were compared. ResNet models which have more than 20 layers showed good performance on classification task and also, positive effect of increased input image size was observed.**

Keywords—Deep learning, camera-trap photos, classification of leopard images.

I. INTRODUCTION

Photo-trap cameras are placed to potential path-way of wild animals. These cameras take photos if they sense motion. Usage of photo-trap cameras has rapidly increased in the last decade. A single photo-trap camera may capture 1000 images for a month. Huge number of images may be stored in a few months. Examination of all images whether it contains animal or not, takes so much time and effort. To minimize time consumption, we propose to use a deep convolutional neural network considering its high performance on image classification task.

Artificial neural networks is not new research area. In 1943, McCulloch and Pitts built a model that demonstrate how neuron work in brain [1]. Computers became more sophisticated in 1950's, this improvement gave people to simulate theoretical neural networks. Marvin Minsky who was founder of MIT AI Lab and Seymour Papert wrote a book that is related to analysis on limitation of Perceptrons [21]. In this book, this approach of AI was thought to have a dead-end due to lack of trace on the system and its critical nature. This conclusion was caused to freeze funding and publication to AI. Most people believed that this paper was caused to the AI winter. Paul Werbos proposed that backpropagation can be used in neural networks [22]. He has solved how to train multilayer neural networks in this PhD thesis but due to the AI winter, it required a decade for researchers to work in this area. In 1986, this approach became popular [23]. First time in 1989, it was applied to a computer vision task which is handwritten digit classification [2]. It has demonstrated excellent performance on this task. Again it took more than a decade for computers to handle more complex tasks and to learn from huge amount of image data.

To observe performance of neural networks on computer vision problems, several competitions are arranged all around the world. One of them is Large Scale Visual

Recognition Challenge(ILSVRC) [18]. This event contains several tasks which are object localization, object detection, object detection from video, image classification, and scene segmentation. In image classification task, all competitors train their model on ImageNet [18] dataset. ImageNet 2012 dataset contains 1.2 million images and 1000 classes. Images are classified according to two different evaluation criteria which are top1 and top5 score. In top1 classification, the label with the top score is chosen. If the predicted label is correct, image is counted as correctly classified.

An outstanding performance was observed in 2012. AlexNet [3] got the 1st place in ImageNet 2012 classification task with achieving 16.4% error rate. There is a huge difference between 1st place (16.4%) and 2nd place (26.1%) in ILSVRC 2012 classification task. Several factors are responsible for gaining this outstanding performance. (i) Training dataset reasonably extended, (ii) GPU computing has been used, (iii) better training method which employs 'dropout' [3] has been implemented. In ILSVRC 2014 image classification challenge, GoogleNet [5] took the 1st place achieving 6.67% error rate. Positive effect of network depth was observed. One year layer, ResNet took the 1st place achieving 5.7% error rate [15].

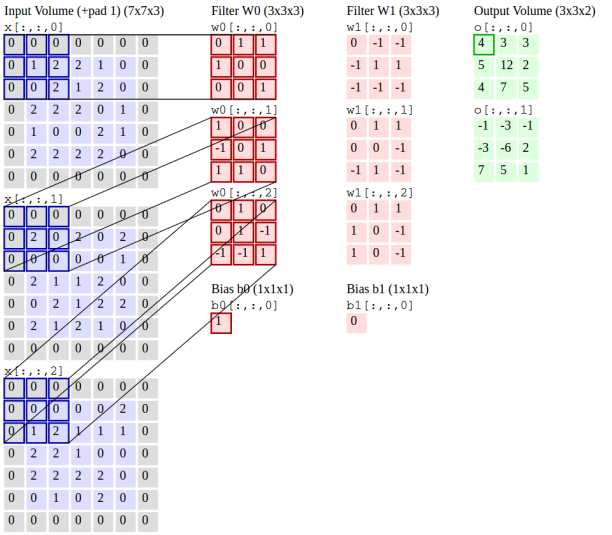
In this paper, we trained plain convolutional neural networks (CNNs) and deep residual NNs for our purpose. It is observed that deep residual networks outperform plain CNNs. We obtained great performance at classification of leopard images for 20+ layer deep residual networks. Best error rate on test set that belongs to plain network which has 5 convolutional layers was recorded as 4.5%. Best error rate coming from ResNet models is 0.0%.

The rest of the paper is organized as follows. In Section 2, we explain types of CNNs and discuss their advantages and disadvantages. Experiments are given in Section 3, conclusion part exists in Section 4.

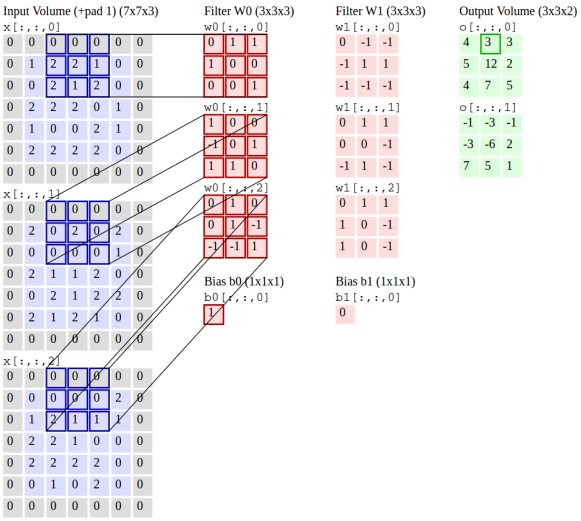
II. BACKGROUND

A. *Convolutional Layer*

Convolutional layer is core building of convolutional neural networks. It contains plenty of learn-able filters (or kernels). Each filter is convolved across width and height of input images. At the end of training process, filters of network are able to identify specific types of images which contain same types of shapes that are related to trained images, same type of spatial positions, etc. One mathematical example is given to illustrate how convolutional layers work. In this example, 5×5 RGB image is given to the network. It is convolved with two kernels that are $3 \times 3 \times 3$ (height, weight, and depth). Convolution is applied with stride size of 2. During the convolution, zero padding is added to enlarge image size. First convolution operation can be seen at Fig.1(a). After



(a) First convolution operation applied with filter W0. Computation gives us the top-left member of next layer



(b) Second convolution operation. Again applied with filter W0. Stride is equal to 2

Figure 1: Convolution Process [25]

applying stride size of 2 to filters, second convolution operation can be seen at Fig.1(b).

Element wise addition is applied in each convolution phase. For example in the first convolution operation which can be seen at Fig.1(a), $[(0 \times 0 + 1 \times 0 + 1 \times 0 + 1 \times 0 + 0 \times 1 + 0 \times 2 + 0 \times 0 + 0 \times 0 + 1 \times 2) + (1 \times 0 + 0 \times 0 + 0 \times 0 + -1 \times 0 + 0 \times 2 + 1 \times 0 + 1 \times 0 + 1 \times 0 + 0 \times 0) + (0 \times 0 + 1 \times 0 + 0 \times 0 + 0 \times 0 + 1 \times 0 + (-1) \times 0 + (-1) \times 0 + (-1) \times 1 + 1 \times 2) + 1(bias) = 4]$ which is expected.

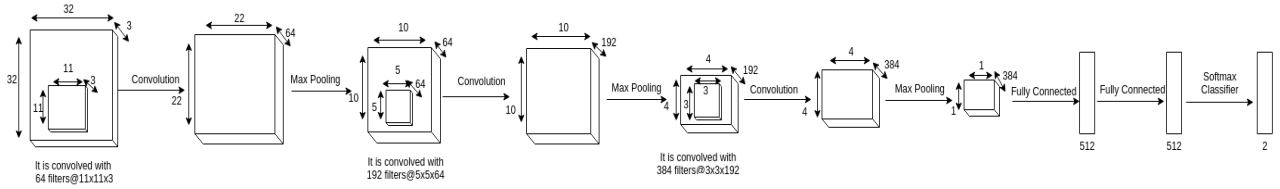


Figure 2: Plain convolutional neural network example with 3 convolutional and 2 fully connected layers

B. Plain Convolutional Neural Networks

Plain CNNs generally have a few convolution layers and fully connected layers. One example of plain network can be seen in Fig.2, where input image (size 32x32) is convolved with 64 filters of size 11x11x3. Depth of the next layer becomes 64 due to the convolution with 64 filters. After the first convolution layer, max-pooling operation is applied and dimensions reduce to 10x10. Feature maps in this layer are convolved with 192 filters of size 5x5x64. Depth of the next layer becomes 192. After each convolution layer, max-pooling operation is applied. Same steps are applied to the last convolution layer. Output of the last max-pooling operation is connected to a fully connected layer.

C. Vanishing Gradient Problem

After observing huge success with CNNs [3], layer number of neural networks has increased year by year [4] [5]. Is it meaningful to increase the number of layers in neural networks as we want? Answer to this question, accuracy of neural networks which have more than tens of layers is saturated after few iterations. This early converging problem isn't caused by overfitting. Adding more layers cause to higher training error, mentioned in [13], [14]. This problem named as vanishing gradient. How depth affects the neural networks is examined in an experiment [19]. Example code can be found at the following link: https://github.com/mnielsen/neural-networks-and-deep-learning.git. Several models with different number of layers have been trained MNIST handwritten digit dataset [20]. Classification accuracy can be seen at Table I [19].

Classification accuracy	
Model that has 1-hidden layer	96.48
Model that has 2-hidden layers	96.90
Model that has 3-hidden layers	96.57
Model that has 4-hidden layers	96.53

Table I: Classification accuracy of models that are trained with MNIST dataset

It is observed that, more than few layers does not have positive effect on the accuracy. Accuracy of model may even decrease with respect to its depth.

We will try to explain the vanishing gradient problem here. A simple neural network is shown at Fig.3 [19], where $w_1, w_2, ...$ are weights, $b_1, b_2, ...$ are biases, and C is the cost function. a_j is $\sigma(z_j)$, where σ denotes activation function which can be sigmoid, rectified linear unit etc., and $z_j = w_j * a_{j-1} + b_j$ is weighted input to the neuron. If predicted output which is a_4 in this example is close to actual output, then cost will come near zero. Otherwise, it will be high.

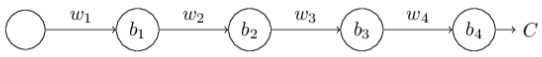


Figure 3: Simple Neural Network

Gradient equation which is related to the first hidden neuron is shown below:

$$\frac{\partial C}{\partial b_1} = \sigma'(z_1) \cdot \overbrace{w_2 \cdot \sigma'(z_2)}^{1/4} \cdot \overbrace{w_3 \cdot \sigma'(z_3)}^{1/4} \cdot w_4 \cdot \sigma'(z_4) \cdot \frac{\partial C}{\partial a_4} \quad (1)$$

Derivative of sigmoid function equals to maximum 25% of its previous value (Fig.4 [19]). Thus, weights usually satisfy $|w_j \times \sigma'(z_j)| < 1/4$. As a result of that, products decrease exponentially.

As shown in Equation 1, magnitude of weights decreases to 25% of its previous value at end of each layer. So, gradient of $\partial C/\partial b_1$ usually become 16 times smaller than the gradient of $\partial C/\partial b_3$. As a result, vanishing gradient phenomenon occurs and layers at the beginning learn very slow.

D. Deep Residual Learning

To deal with the vanishing gradient problem, deep residual learning was proposed [15]. Deep residual network consists of many "Residual Units". Residual block can be applied at regular intervals. General equation is shown below [16]:

$$y_l = F(x_l, W_l) + x_l \quad (2)$$

where x_l is the input and y is the output vector. In a standard convolutional layer, $x_{l+1} = f(y_l)$ where f is a

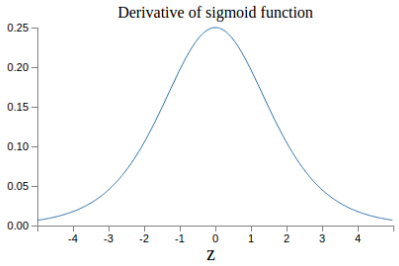


Figure 4: Derivative of Sigmoid Function

Relu function. $F(x_l, W_l)$ symbolizes the residual block which includes one or more layers of convolution. Result of the residual block is added to x_l before applying the activation function (Fig.5 [15]). Operation of $F(x_l, W_l) + x_l$ is an element-wise addition.

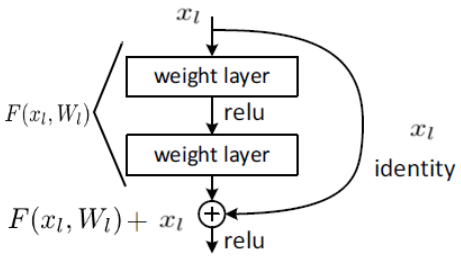


Figure 5: Residual Learning block

Residual block does not increase the model complexity. It is the major advantage of residual learning. Dimensions of x_l and $F(x_l, W_l)$ should be equal to apply an element-wise addition. If dimension of x_l and $F(x_l, W_l)$ are not equal, lower dimension can be increased with this equation [15]:

$$y_l = F(x_l, W_l) + W_s x_l \tag{3}$$

$F(x_l, W_l)$, the residual block, can be made by any number of layers. Generally, it is used with two or three layers. If $F(x_l, W_l)$ has just one layer, no advantage of residual learning has been observed.

We trained several deep residual networks to observe their performance at classification of leopard images. 20-layer ResNet model is shown at Fig.6.

III. EXPERIMENTS

A. Dataset

To train models, two classes are identified that are leopard and background classes. Two different datasets which consist of same images with two different sizes have been prepared. Images are taken from ImageNet dataset [18]. In first dataset, each image is resized to

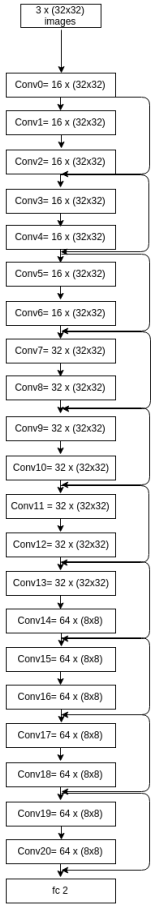


Figure 6: Architecture of Deep Residual Network

136x136 pixels. Then, 128x128 pixels are randomly cropped from resized image. In second dataset, each image is resized to 40x40 pixels and 32x32 pixels are randomly cropped from image. We have used 700 images for each class to train the models and 120 images of each class to test. We selected the leopard images in the test set so that they resemble actual photo-trap photos. Diversification of the dataset is increased by horizontally flipping some of the images. If horizontally flip operation is applied, then image is randomly cropped. If horizontally flip operation is not applied, central of image is cropped. Flowchart of applied operation can be seen at Fig.7.

B. Training Methodology

We have trained the NN models with ILSVRC-2012 dataset [18]. Plain-Network-A consists of 3 convolution layers, 2 fully connected layers and a softmax layer. Plain-Network-B consists of 5 convolution, 2 fully connected layers and a softmax layer. ResNet-20 consists of 20 convolution layers and a softmax layer, ResNet-32 consists of 32 convolution layers and a softmax layer, ResNet-

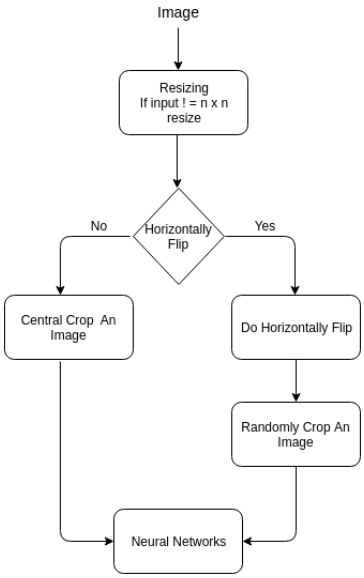


Figure 7: Flowchart of image preparation to train models.

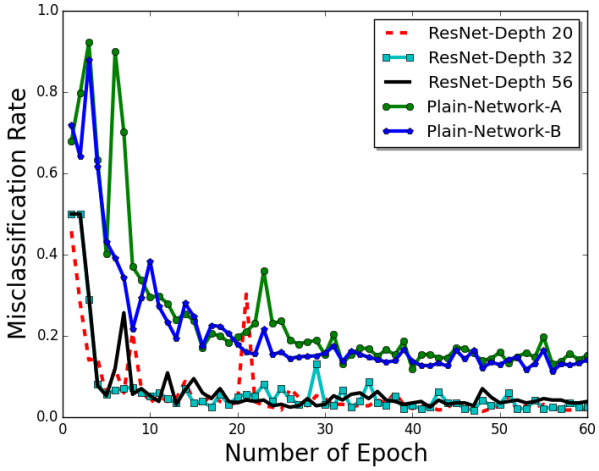


Figure 8: Misclassification Rate of Models

56 consists of 56 convolution layers and a softmax layer. Convergence of each model took 50 epochs. Fig.8 shows the convergence of all five trained models. Misclassification rate (misclassification of training samples) decreases as the number of epochs on the training set increases. Trained filters which belong to the first convolution layer of Plain Network-B shown at Fig.9(a), input image can be seen at Fig.9(b) and output of first convolution layer is shown at Fig.9(c). Small details which can be the speckles of leopards or greenish smooth surfaces that may represent leaves can be seen at Fig.9(a). For instance, filter in 1^{th} row and 1^{th} column are related to speckle of leopards. Activated neurons of that filter are seen in Fig.9(c), which are fired on the body of the leopard.

C. Results and Discussion

We have tested the trained models on 120 background and 120 leopard images. Performance of each model can be seen at Table II and Table III.

Two different classifiers were reported in Table II and Table III. One of them is original Softmax classifier which choses the class with maximum probability. In second classification method, we do not choose the maximum argument of Softmax classifier. If probability of being leopard is higher than 0,29, example is assigned to leopard class to prevent false-negative leopard classification. This is because the cost of false-negative is considerably higher than false-positive since false-negative means that image will not be examined by experts anymore.

To evaluate the performance for all possible threshold values, we plot ROC curves for 32x32 and 128x128 input images, which can be seen in Fig.10 and Fig.11. According to the ROC curves, deeper models have shown

	Number of misclassified samples			
	Softmax Classifier		Modified Classifier	
	background	leopard	background	leopard
Plain Network-A	7	15	16	5
Plain Network-B	9	14	22	6
Resnet20-Layers	1	5	2	2
Resnet32-Layers	1	4	4	2
Resnet56-Layers	0	5	1	3

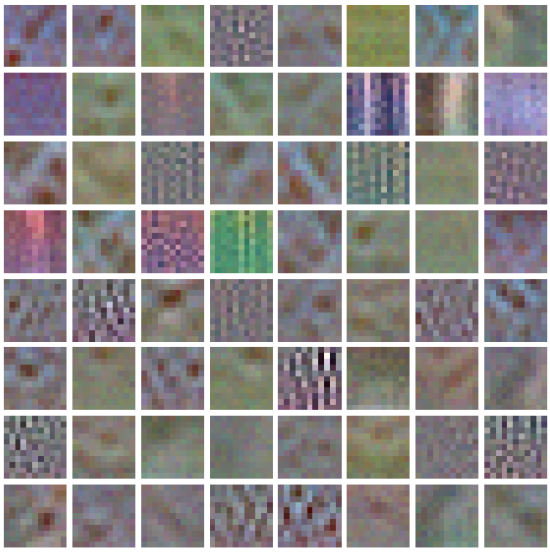
Table II: Number of misclassified samples on models which are trained with 32x32 input images. Numbers are out of 120 test images per class.

better performance than plain models. It can also be observed that increasing input image size (to 128x128) has a positive effect on the classification accuracy.

For 32x32 image size, Plain Network-B has shown lower performance than Plain Network-A, because output size of last convolution layer in Plain Network-A is too small to make two more convolution. Zero padding is added to make more convolution but it decreases the performance of Plain Network-B. Unlike with small image size (32x32), with bigger image size (128x128) Plain-Network-B got better results.

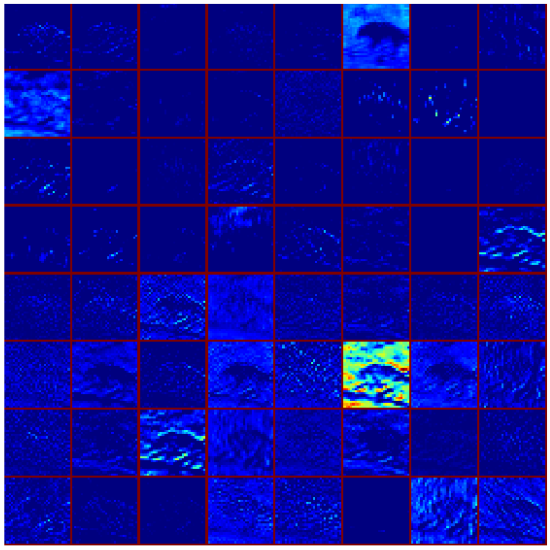
IV. CONCLUSION

In this work, several CNNs were trained and their performance were tested for the aim of distinguishing leopard images from background images. ResNet models showed excellent performance when compared to the plain networks. Leopard images which are similar to photo-trap photos are perfectly classified with 56-layer ResNet and softmax classifier. For less number of layers, one could decrease the threshold not to get any false-negatives (causing more false-positives) since the penalty for false-negatives is higher.



(a) Weights and bias of trained filters

(b) Input image



(c) Output of first convolution layer

Figure 9: Filter visualization

	Number of misclassified samples			
	Softmax Classifier		Modified Classifier	
	background	leopard	background	leopard
Plain Network-A	4	12	7	7
Plain Network-B	4	7	7	7
Resnet20-Layers	1	1	1	0
Resnet32-Layers	1	3	2	1
Resnet56-Layers	0	0	0	0

Table III: Number of misclassified samples on models which are trained with 128x128 input images. Numbers are out of 120 test images per class.

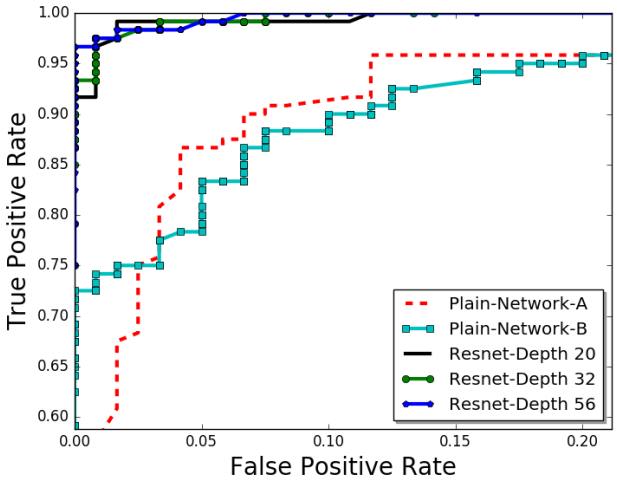


Figure 10: ROC curve of trained models that are trained with 32x32 input images.

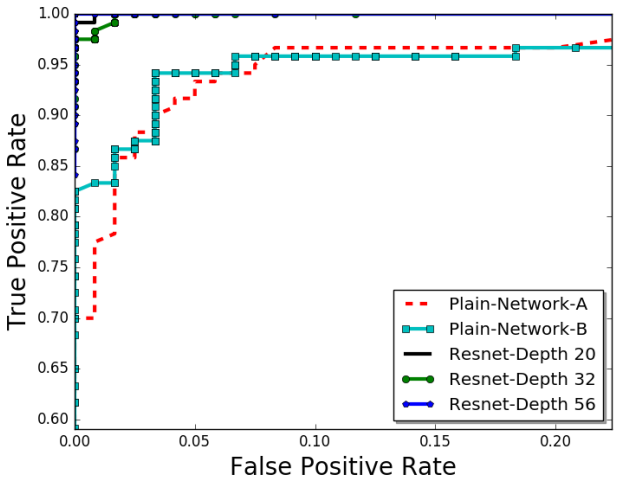


Figure 11: ROC curve of trained models that are trained with 128x128 input images.

V. ACKNOWLEDGEMENT

This work is supported by the TUBITAK project 115E918.

REFERENCES

[1] Warren S. McCulloch and Walter Pitts, "A Logical Calculus of Ideas Immanent in Nervous Activity," vol. 5, no.4, pp. 115-133, 1943.

[2] LeCun, Y., Jackel, L. D., Boser, B., Denker, J. S., Graf, H. P., Guyon, I., Henderson, D., Howard, R. E., and Hubbard, W., "Handwritten digit recognition: Applications of neural network chips and automatic learning. IEEE Communications Magazine", 27(11), 41–46. 368, 1989.

[3] A. Krizhevsky, I Sutskever, and G.E Hinton. "Imagenet classication with deep convolutional neural networks," In NIPS, 2012.

[4] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," In ICLR, 2015.

[5] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D.Erhan, V. Vanhoucke, and A.Rabinovich, "Going deeper with convolutions," In CVPR, 2015.

[6] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," In ECCV, 2014.

[7] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," In ECCV, 2014.

[8] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependen-cies with gradient descent is difficult," IEEE Transactions on Neural Networks, vol. 5, no. 2, pp. 157-166, 1994.

[9] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," In AISTATS, 2010.

[10] Y. LeCun, L. Bottou, G. B. Orr, and K.-R.Müller, "Efficient backprop. In Neural Networks: Tricks of the Trade," Springer, pp. 9-50, 1998.

[11] A. M. Saxe, J. L. McClelland, and S. Ganguli, "Exact solutions to the nonlinear dynamics of learning in deep linear neural networks," arXiv:1312.6120, 2013.

[12] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," In ICCV, 2015.

[13] K. He and J. Sun, "Convolutional neural networks at constrained time cost," In CVPR, 2015.

[14] R. K. Srivastava, K. Greff, and J. Schmidhuber, "Highway networks," arXiv:1505.00387, 2015.

[15] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," In CVPR, 2016.

[16] K. He, X. Zhang, S. Ren, and J. Sun, "Identity Mappings in Deep Residual Networks," arXiv:1603.05027, 2016.

[17] V. Nair and G. E. Hinton, "Rectified Linear Units Improve Restricted Boltzmann Machines," In ICML, 2010.

[18] Olga Russakovsky et al., "ImageNet Large Scale Visual Recognition Challenge," International Journal of Computer Vision (IJCV), vol. 115, no. 3, pp. 211-252, 2015.

[19] Michael Nielsen, "Neural Network and Deep Learning", [Online]. Available: http://neuralnetworksanddeeplearning.com/chap5.html

[20] Y. LeCun and C. Cortes, "MNIST handwritten digit database, 2010.

[21] M. Minsky and S. Papert. "Perceptrons. An Introduction to Computational Geometry.", M.I.T. Press, Cambridge, Mass., 1969.

[22] P. Werbos. "Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences" PhD thesis, Harvard University, Cambridge, MA, 1974.

[23] D. E.Rumelhart, G. E.Hinton, and R. J Williams, "Learning representations by back-propagating errors", Nature, 323, 533–536, 1986.

[24] K.-F. Lee, "Automatic Speech Recognition: The Development of the SPHINX SYSTEM," Kluwer Academic Publishers, Boston, 1989.

[25] Convolution image, [Online]. Available: http://cs231n.github.io/convolutional-networks