X-ray Image Classification Using State-of-the-art Networks

Hamid Hassanpour

Email: h.hassanpour67@gmail.com

Abstract: In this paper, we investigate capability of state-of-the-art networks in classification of X-Ray images. We probe their performance in correct classification of X-Ray images which are labelled as having infection with virus or bacteria. In order to match model's architecture with our expectations and enhance compatibility of models with our dataset, additional layers are applied at top of these models where each model is connected to stack of fully connected dense layers. Final models are evaluated in percentage of correct classification in restricted computation time.

Keywords: X-ray image, image classification, VGG, Inception, Xception, ResNet

1 Introduction

One can consider advent of convolutional neural networks as the greatest leap in image classification and object localization. It was introduced by Lecun et al. (1995) and later on, many architectures are designed based on convolution layer. The objectives of these developed architectures are mainly to achieve maximum capability in classification and, at the same time, considering and addressing computational contrarians. After more than 25 years, convolution layers are still the major basic block for these models.

As today, many attempts are made to design efficient models for image classification. The idea of convolution in image classification was born in Lecun et al. (1995) based on the notion that similar features can appear in different places of an image. These features can be grasped by convolution operation and stored as "feature map". Then, more complex architectures were designed. Simonyan and Zisserman (2014) describes very deep convolutional network, known as "VGG", where small 3*3 convolution layers are stacked. Without having aggressive preprocessing, the network outperformed many suggested architectures at the time. Later on, two models were designed with having better performance than VGG. The ground idea for making these networks was in Lin et al. (2013) as "Network in Network" concept, where instead of operating generalized linear operation on input, several non-linear operations are performed in layers, where each layer consists of a multilayer perceptron with non-linear function approximation which slides over input space, and then few stacked fully connected layers after perceptron layer. This sequence of operations can be considered as one network which then can be stacked to create Network in Network architecture.

In the first one, Szegedy et al. (2015) introduces "Inception" architecture based on designing network consisting of stacked convolution modules, where for designing optimal modules two goals including considering computational power constrains and achieving maximum representational power is followed. In the second one, He et al. (2016a) addresses degradation problem of deep networks by introducing residual learning block,

showing that stack of these block eliminates degradation problem, and at the same time, network can benefit from deeper layers. He et al. (2016b) proposed a better residual network, known as “ResNet”, which has lower training error and is faster compare to previous proposed residual network. The main idea was creating a direct path for flowing information through entire network. Other studies developed networks based on Inception. Szegedy et al. (2016) discusses method for designing Inception-like model that achieve reasonable computational time while maintaining model’s representational capability, especially by replacing large size convolution layer with multiple small size convolution layers. Chollet (2017) introduces “extreme” version of Inception (“Exception”), where instead of mapping input into several segments, each one consists of 1×1 followed by 3×3 convolutions, it maps input by one 1×1 convolution layer and then each channel of this convolution is mapped with segments of 3×3 convolutions. Results indicated that this new version outperforms traditional Inception network.

In this article, we compare performance of these models in X-Ray image classification. At the same time, we put time constraint in model training time, i.e., the model learns faster the better. For having better customized classification model, while benefiting from pretrained state-of-the-art models, we apply several fully connected layers at top of these models, where trainable parameters of these layers are aimed to customize final model to our purpose. Besides, although state-of-the-art models have proven their performance on regular images, X-Ray images have their own specifications and despite regular images, their shapes are very smooth and therefore edge detection and feature extraction are harder than usual.

2 Methodology

Our data set is collected from Kaggle Data Set (2021) consists of 5284 X-Ray images for training and 624 images for validation. Images are classified in 3 classes including normal condition, virus infection and bacterial infection. Figure 1 shows examples of these classes.

Figure 1: 3 X-Ray images of normal condition(a), bacterial infection(b), and virus infection(c)

In our method, we examine well-known networks which are described in introduction in order to compare and evaluate their performance. As X-Ray images are smooth and object detection is harder than usual images, we stack 4 fully connected layers on top of state-of-the-art models. These fully connected layers are trainable and aimed to grasp smooth changes in X-Ray images. Besides, it allows us to customize pretrained networks to our criteria. Figure 2 shows these architectures. Numbers in front of FC show number of nodes in each layer. As the picture shows, each model is stacked with 4 fully connected layers. On top of FC layers, softmax layer classifies images into 3 sets.

Softmax			
FC128			
FC256			
FC512			
FC1024			
Inception	Xception	VGG	ResNet

Figure 2: Models architecture consists of pretrained models and 4 FC layers followed by Softmax layer

3 Results

In this study, we consider training time valuable and consider it limited, therefore each model is trained for specific time, which we choose to be 40 minutes. At the end of this time, model's performance on validation dataset is reported. We examined these models in TensorFlow environment, where these pretrained models are available and they can be stacked with FC layers (Abadi et al. (2015)). With special thanks to Coursera for their education (TensorFlow Specialization (2021)), source code is developed and it is available in State-of-the-art Networks (2021).

Performance of networks are shown in table 1. As the data shows, all models have acceptable performance with classification accuracy more than 80%, except VGG. It was expected since other networks than VGG are more advanced and they are designed after VGG development.

Table 1: Model's accuracies

Models	Inception-FC	Xception-FC	VGG-FC	ResNet-FC
Accuracy (%)	83	83	75	83

Reference

- Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.s, Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Kaiser, L., Kudlur, M., Levenberg, J., Zheng, X. (2015) *TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems*. <https://arxiv.org/abs/1603.04467> (Accessed 1 September 2020).
- Chollet, F. (2017) "Xception: Deep Learning with Depthwise Separable Convolutions", *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, pp. 1800-1807.

H. Hassanpour

- He, K., Zhang, X., Ren, S. and Sun, J. (2016a) “Deep Residual Learning for Image Recognition”, *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, pp. 770-778.
- He, K., Zhang, X., Ren, S. and Sun, J. (2016b) *Identity Mappings in Deep Residual Networks*.
<https://arxiv.org/abs/1603.05027> (Accessed 3 February 2021)
- Kaggle Data Set (2021) [online] <https://www.kaggle.com/praveengovi/coronahack-chest-xraydataset> (Accessed 3 February 2021)
- Lecun, Y., Jackel, L., Cortes, C., Denker, J., Drucker, H., Guyon, I., Muller, U., Sackinger, E., Simard, P., Vapnik, V. (1995) *Learning Algorithms For Classification: A Comparison On Handwritten Digit Recognition*.
<http://yann.lecun.com/exdb/publis/index.html#fulllist> (Accessed 3 February 2021)
- Lin, M., Chen, Q. and Yan, S. (2013) *Network In Network*.
<https://arxiv.org/abs/1312.4400> (Accessed 3 February 2021)
- Simonyan, K., and Zisserman, A. (2014) *Very Deep Convolutional Networks for Large-Scale Image Recognition*.
<https://arxiv.org/abs/1409.1556> (Accessed 3 February 2021)
- State-of-the-art Networks (2021) [online] <https://github.com/HamidHassanpour/State-of-the-art-Networks> (Accessed 3 February 2021)
- Szegedy, C., Wei, L., Yangqing, J., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2015) “Going deeper with convolutions”, *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, USA, page 1-9.
- Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J. and Wojna, Z. (2016) “Rethinking the Inception Architecture for Computer Vision”, *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, pp. 2818-2826.
- TensorFlow Specialization (2021) [online] <https://www.coursera.org/professional-certificates/tensorflow-in-practice?> (Accessed 3 February 2021)