IRSTI 28.23.15
UDC 004.93'1

USING CONVOLUTIONAL NEURAL NETWORKS IN SOLVING PROBLEMS OF IMAGE ANALYSIS AND RECOGNITION

A.Astanayeva[1,2] A.Kozbakova[1]

[1]Institute of Information and Computational Technologies,Almaty, Kazakhstan
[2]Al-Farabi Kazakh National University, Almaty, Kazakhstan

[1]astanayeva@mail.ru, [2] ainur79@mail.ru
[1]ORCID ID: https://orcid.org/0000-0002-6395-5838
[2]ORCID ID: https://orcid.org/0000-0002-5213-4882

Abstract. Convolutional Neural Network (CNN) is a special type of Neural Networks, which has shown exemplary performance on several competitions related to Computer Vision and Image Processing. Some of the exciting application areas of CNN include Image Classification and Segmentation, Object Detection, Video Processing, Natural Language Processing, and Speech Recognition.

The purpose of the work, the results of which are presented in the article, was to research of modern architectures of convolutional neural networks for image recognition. The article discusses such architectures as AlexNet, ZFnet, VGGNet, GoogleNet, ResNet. Based on the results obtained, it was revealed that at the moment the network with the most accurate result is the ResNet convolutional network with an accuracy rate of 3.57%. The advantage of this research is that the given article gives a brief description of the convolutional neural network, and also gives an idea of the modern architectures of convolutional networks, their structure and quality indicators.

Keywords: filter, convolution, neural networks, architecture, deep learning, convolutional neural networks

Introduction

Recognition of visual images is one of the most important components of control systems and information processing, automated systems and decision-making systems. Tasks related to classification and identification of objects, phenomena and signals characterized by the final a set of certain properties and characteristics, arise in such industries as robotics, information retrieval, monitoring and analysis of visual data, artificial intelligence research. With the growth of the computing power of personal computers, as well as the emergence of image databases, it became possible to train deep neural networks. In the task of image recognition, convolutional neural networks (Convolutional Neural Networks) are used. The purpose of the article is to review modern convolutional neural network architectures for the image classification problem.

One of the tasks of machine learning is the task of image classification. To classify an object in an image means to indicate the number to which it belongs recognizable object. For evaluating machine learning algorithms, it is commonly used tagged image databases, e.g. CIFAR-10, ImageNet, PASCAL VOC.[1]

One of the most successful models, considered a recognized leader in the field of image recognition, is convolutional neural network.

Convolutional neural networks (CNNs) are used for optical pattern recognition, image classification, object detection, semantic segmentation, and other tasks. The foundations of the modern SNS architecture were laid in one of the first networks - LeNet-5 by Jan LeKun.

Convolutional neural network structure

The convolution network is a multilayer perceptron (perceptron, English perceptron from

Latin perceptio - perception [2]) - a mathematical or computer model of information perception by the brain, created for recognizing 2D surfaces with high the degree of resistance to scaling, transformations and other types of deformation [3].

Learning to solve such a problem is carried out with reinforcement, with the use of networks of the form, the architecture of which corresponds to the following constraints.

Extraction of features. Each neuron receives an input signal from a local receptive field in the previous layer, extracting its local features. Once a feature is retrieved, its location does not matter, since its location relative to other signs has been approximately established.

Display of signs. Each computational layer of the network consists of many feature maps. Each feature map is shaped like a plane on which all neurons must share the same set of synaptic weights. This form of structural constraint has advantages.

Displacement invariance. Displacement invariance is realized through feature maps using convolution with a small kernel that performs flattening function.

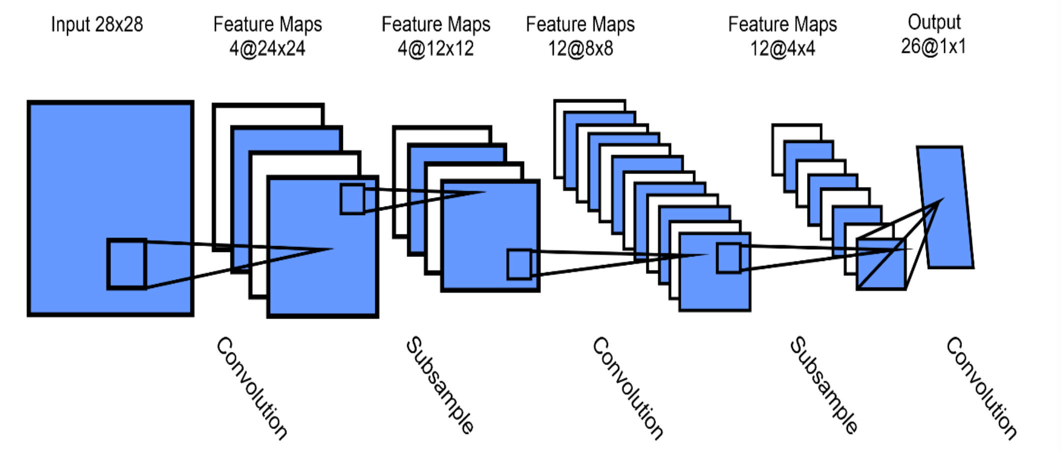

Figure 1 – Convolution network for image processing

Subsampling. Each layer of convolution is followed by a computational layer that implements local averaging and subsampling. By means of local averaging, a reduction in resolution for feature maps is achieved. Such an operation leads to a decrease in the sensitivity of the output signal of the display operator signs, as well as displacement and other forms of transformation.

Figure 1 shows a diagram of a convolutional network consisting of one input, four hidden, and one output layers of neurons. This network was created for image processing, in particular for handwriting recognition. The input layer, consisting of a matrix of 28 × 28 sensor nodes, receives images of various symbols, which are pre-offset to the center and normalized in size. After that, the computational layers alternately implement the convolution and subsample operations.

The first hidden layer is folding. It consists of four feature maps, each of which is a 24 × 24 matrix of neurons. Each neuron has a 5 × 5 sensitivity field.

The second hidden layer performs the local averaging operation as well as subsampling. It also consists of 4 feature maps containing 12 × 12 matrices. Each neuron corresponds to a 2 × 2 receptive field, a sigmoidal activation function, an adjustable threshold, and an adjustable coefficient. The adjustable coefficient and threshold determine the working area of the neuron. For example, with a small coefficient, the neuron operates in a quasilinear mode.

The third hidden layer performs a re-convolution operation. The layer consists of 12 feature maps, each of which is an 8 × 8 matrix of neurons. Each neuron of the third hidden layer can have synaptic connections with different feature maps from the previous hidden layer. The fourth hidden layer performs a second subsampling and repeated local averaging. The layer consists of 12 feature maps, however each feature map contains a matrix of 4 × 4 neurons.

The output layer performs the final convolution step. The layer consists of 26 neurons, each of which corresponds to one of the 26 letters of the Latin alphabet. Each neuron corresponds to a 4 × 4 receptive field [4].

Convolutional neural network architectures for image classification

AlexNet is a convolutional neural network that won the ImageNet LSVRC2012 competition with an error of 15.3%. The network had a partially similar architecture to the LeNet network from J.Lekun, but it was deeper, with more convolutional layers. The network consisted of 11 × 11, 5 × 5, 3 × 3 convolution, max merge, dropout, data increase, function activation of ReLU, SGD (Stochastic Gradient Descent) with an impulse. Function application activation occurred after each convolutional and fully connected layers.

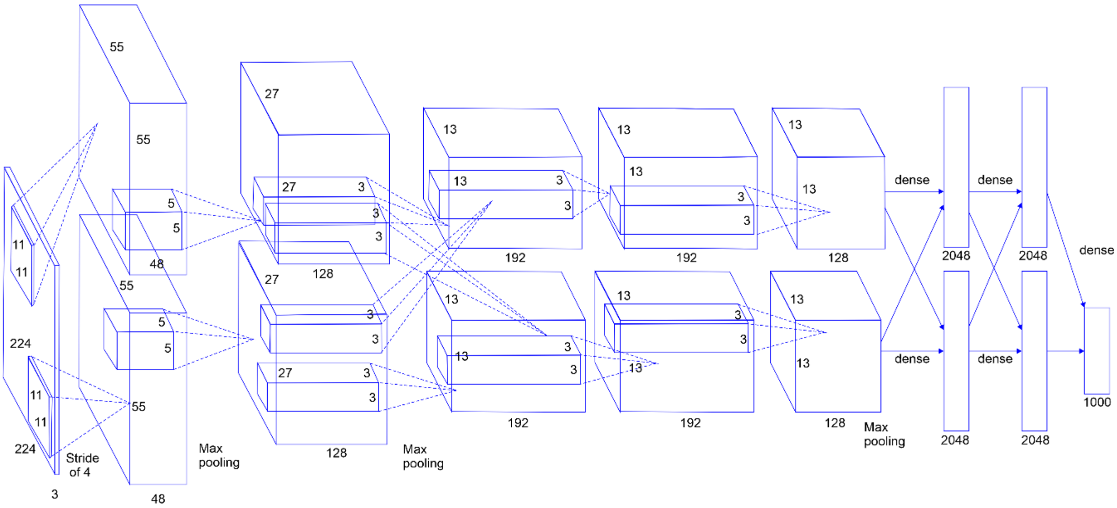

Figure 2 - AlexNet architecture

Shown in Fig. 2. AlexNet architecture, includes 8 layers with weight coefficients, where the first 5 layers are convolutional, the next 3 are fully connected. The output of the last fully connected layer is fed to the softmax activation function, which distributes to the class labels. Further, the network maximizes the goal of the polynomial logistic regression, which is equivalent to maximizing the average over the training cases of the logarithmic probability of the correct label in the prediction distribution.

Using stochastic gradient descent with a learning rate of 0.01, momentum equal to 0.9 and a weight loss of 0.0005. The learning rate is divided by 10 times the accuracy plateau, also decreasing by 3 times during the learning process.

Formulas for updating weight coefficients. Updating the weights (w), where i is the iteration index, v is the momentum variable, and epsilon is the learning rate shown in the diagram. The learning rate was selected equal for all layers, and was also adjusted manually during the entire learning process. The next step was to divide the learning rate by 10, when the number of validation errors stopped decreasing. AlexNet shows the result of top-5 errors - 15.3%, respectively. ZFNet is the winner of ILSVRC 2013 with a top-5 error of 11.2%. The main role in this is played by the adjustment of hyperparameters, namely the size and number of filters, packet size, learning rate, etc. M. Zieler and R. Fergus proposed a system for visualizing kernels, weights, and hidden image representation. The system was named DeconvNet.

The network architecture of ZFNet is almost identical to that of AlexNet. The significant differences between them in architecture are as follows:

- the size of the ZFNet filter and the step in the first convolutional layer in AlexNet is 11 × 11, the step is 4; in ZFNet - 7 × 7, step is 2;
- number of filters in pure convolutional layers of the network (3, 4, 5).

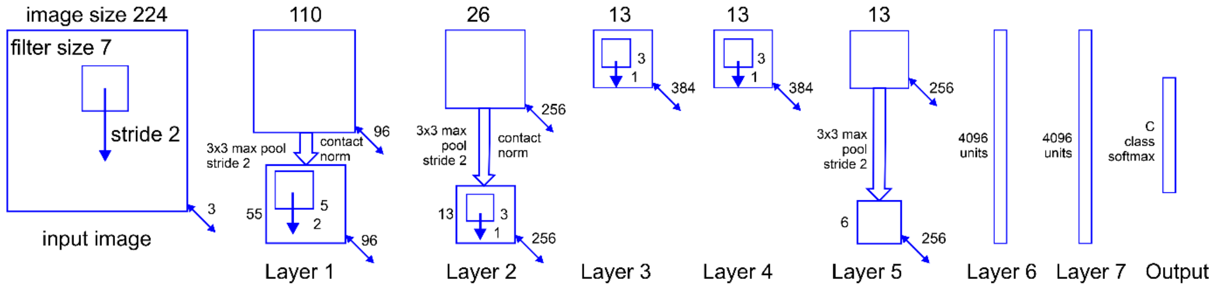

Figure 3 - ZFNet architecture

VGGNet.

In 2014 K. Simonyan and E. Tsisserman from Oxford University proposed a neural network architecture called VGG (Visual Geometry Group). VGG16 is an improved version of AlexNet in which the large filters (size 11 and 5 in the 1st and 2nd convolutional layers) are replaced with several 3x3 filters one after the other.

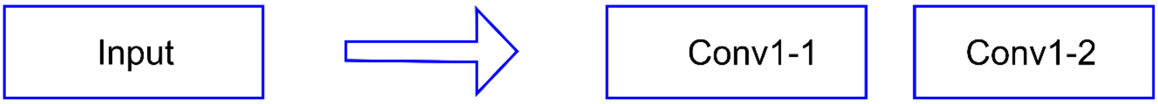

Figure 4 - Architecture of the VGGNet network.

During training, the input to ConvNets (Convolutional networks) is an RGB image of a fixed size of 224 × 224 px. On the in the next step, the image is passed through a 3x3 stack of convolutional layers. In one of the VGGNet network configurations uses 1 × 1 filters that can considered as a linear transformation of the input channels.

The convolution step is fixed at 1 pixel. The spatial padding of the input of the convolutional layer is chosen so that the spatial resolution is preserved after convolution, that is, the padding is 1 for 3x3 convolutional layers. Spatial pooling is done using five max-pooling layers that follow one of the convolutional layers (not all convolutional layers have subsequent maxpooling layers). The max-pooling operation is performed on a 2x2 pixel window with a step of 2.

After the stack of convolutional layers, there are 3 fully connected layers: the first two layers have each 4096 channels, the third layer - 1000 channels (because in the ILSVRC competition it is necessary distribute objects into 1000 categories). The last layer is softmax. All hidden layers are equipped with the ReLU activation function.

The authors have demonstrated that building blocks can be used to achieve specific results in the ImageNet competition. Top-5 errors dropped to 7.3% [5].

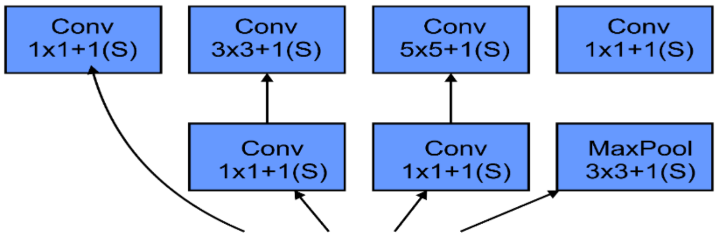

Figure 5 - GoogleNet architecture

A convolutional network from Google (GoogLeNet) known as Invеption-v1 is the winner of the ILSVRC 2014 with a top-5 error of 6.7% [Szegedy et al., 2015].

All convolutions on the network, including those inside Inception modules, use straight-line linear activation. The network has 22 layers when counting only layers with parameters. The total number of layers used to build the network is 100. Moving from fully connected layers to a medium pool improved the accuracy of the top 1 by about 0.6%, but the use of dropout remained necessary even after removing the fully connected layers.

Given the depth of the mesh, being able to propagate gradients back across all layers was an effective challenge. The high performance of smaller networks in this task suggests that the functions created by the layers in the middle of the network must be highly discriminatory. By adding auxiliary classifiers associated with these intermediate levels, discrimination at the lower levels in the classifier was expected. This helped to overcome the vanishing gradient problem by providing regularization. Classifiers in a network take the form of small convolutional networks that are placed on top of the output of Inception modules. During training, their loss is added to the overall weight loss of the net. During inference, these auxiliary networks are discarded. Later control experiments showed that the influence of auxiliary networks is relatively small (about 0.5%) and that only one of them is required to achieve the same effect.

The GoogLeNet architecture uses the Inception module, and the network is built based on modules of this type [1].

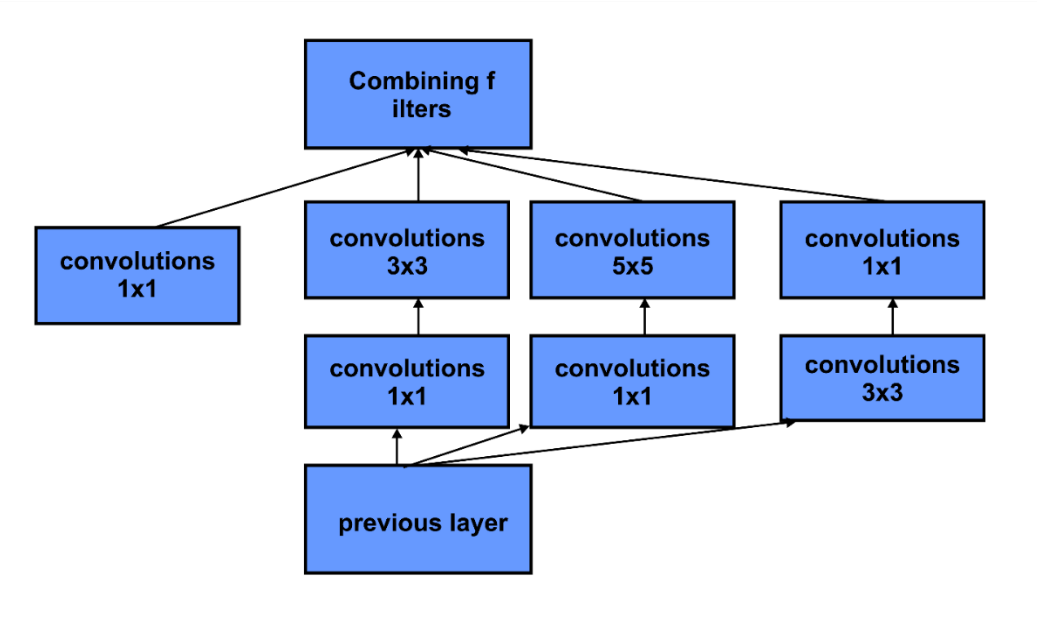

Figure 6 - Module Inception

Inception module. Inception uses multiple (parallel) branches that compute different properties based on the same input and then merge the results together. A 1 × 1 convolution is a way to reduce the dimension of a property map. This type of convolutional layers is presented in the work "Network" in M. Lin's network. As a result, this architecture allows to reduce the number of errors for the top-5 categories by another 0.5% - to the value 6.7%.

Module Inception-v2 and Inception-v3. In the next iteration of the Inception module (Inception-v2 [6]), the layer with a 5 × 5 filter is decomposed into two 3 × 3 layers. The next stage is the use of Batch Normalization [Ioffe, Szegedy, 2015], which allows to increase the learning rate by normalizing the distribution of layer outputs within the network. In the same article, the authors proposed the concept of the Inception-v3 module. In the Inception-v3 module, the principle of filter decomposition is incorporated, namely, the decomposition of an N × N filter with two successive filters 1 × N and N × 1. Also, Inception-v3 uses RMSProp (Adaptive Moving Average Gradient Method) [Hinton, Srivasta, Swersky, 2012], instead of gradient descent, it uses gradient truncation [Pascanu et al., 2013], which is used to improve training stability. A combination of four Inception-v3 modules showed a result in the top-5 category, an error of 3.58% at ILSVRC 2015, Inception-v2 - a top-5 result - 5.60%

ResNet.ResNet is the abbreviated name for the Residual Network (literally - “residualnet").

Convolutional layers have 3 × 3 filters and follow the design rules:

- with the same size of the map of the output objects, the layers have the same number of filters;
- if the size of the feature map is halved, the number of filters, on the contrary, is doubled in order to preserve the complexity of the time for the layer.

Table 1. CNN ResNet characteristics

Layer name	Output size	18-layer	34-layer	50-layer	101-layer	152-layer
Conv1	112×112	7×7, 64, Stride 2				
Conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix}3\times3,64\\3\times3,64\end{bmatrix}\times2$	$\begin{bmatrix}3\times3,64\\3\times3,64\end{bmatrix}\times3$	$\begin{bmatrix}1\times3,64\\3\times3,64\\1\times1,256\end{bmatrix}\times3$	$\begin{bmatrix}3\times3,64\\3\times3,64\\1\times1,256\end{bmatrix}\times3$	$\begin{bmatrix}3\times3,64\\3\times3,64\\1\times1,256\end{bmatrix}\times3$
Conv3_x	28×28	$\begin{bmatrix}3\times3,128\\3\times3,128\end{bmatrix}\times2$	$\begin{bmatrix}3\times3,128\\3\times3,128\end{bmatrix}\times4$	$\begin{bmatrix}1\times1,128\\3\times3,128\\1\times1,512\end{bmatrix}\times4$	$\begin{bmatrix}1\times1,128\\3\times3,128\\1\times1,512\end{bmatrix}\times4$	$\begin{bmatrix}1\times1,128\\3\times3,128\\1\times1,512\end{bmatrix}\times8$
Conv4_x	14×14	$\begin{bmatrix}3\times3,256\\3\times3,256\end{bmatrix}\times2$	$\begin{bmatrix}3\times3,256\\3\times3,256\end{bmatrix}\times6$	$\begin{bmatrix}1\times1,256\\3\times3,256\\1\times1,1024\end{bmatrix}\times6$	$\begin{bmatrix}1\times1,256\\3\times3,256\\1\times1,1024\end{bmatrix}\times23$	$\begin{bmatrix}1\times1,256\\3\times3,256\\1\times1,1024\end{bmatrix}\times36$
Conv5_x	7×7	$\begin{bmatrix}3\times3,512\\3\times3,512\end{bmatrix}\times2$	$\begin{bmatrix}3\times3,512\\3\times3,512\end{bmatrix}\times3$	$\begin{bmatrix}1\times1,256\\3\times3,256\\1\times1,2048\end{bmatrix}\times3$	$\begin{bmatrix}1\times1,256\\3\times3,256\\1\times1,2048\end{bmatrix}\times3$	$\begin{bmatrix}1\times1,256\\3\times3,256\\1\times1,2048\end{bmatrix}\times3$
	1×1	average pool, 1000 - dfc, softmax				
FLOPs		$1{,}8\times10^9$	$3{,}6\times10^9$	$3{,}8\times10^9$	$7{,}6\times10^9$	$11{,}3\times10^9$

Each ResNet block has two levels of depth (used in small networks such as ResNet 18, 34) or 3 levels (ResNet 50, 101, 152) (Table 1).

50-layer ResNet**:** Each 3-layer block is replaced in the 34-layer network by this 3-layer bottleneck, resulting in a 50-layer ResNet (see Table 1). They use option 2 to increase the dimensions. This model has 3.8 billion FLOPs.

ResNet with 101 and 152 layers: They create ResNet with 101 and 152 layers using more 3-layer blocks (see Table 1). After increasing the depth, 152 layer ResNet (11.3 billion FLOPs) has less complexity than VGG-16 and VGG-19 networks (15.3 / 19.6 billion FLOPs). ResNet - 152 achieves a top 5 result of 3.57%.

Comparison of convolutional neural network models

To assess the performance of convolutional neural network models, indicate the type of error (top-5). The images in the ImageNet database may contain many objects, but only one of them is annotated. The main error criterion is a top-5 error.

The results of comparing the results of various convolutional neural networks are presented in Table 2.

Table 2. Comparison of CNN indicators in image recognition tasks

Neural network	**Top-5**
AlexNet	15,30%
ZF Net	11,20%
VGG Net	7,30%
GoogleLeNet	6,70%
Inception-v2	5,60%
Inception-v3	3,58%
ResNet-152	3,57%

Conclusion

The spread and development of computer vision technologies entails a change in other professional areas of human life. Convolutional neural networks (CNNs) are used in object and face recognition systems, special medical software for image analysis, navigation of cars equipped with autonomous systems, in security systems, and other areas. With the growth of computing power of computers, the advent of image databases, it became possible to train deep neural networks. One of the main tasks of machine learning is the task of image classification. SNS are used for optical recognition of images and objects, object detection, semantic segmentation, etc. In this article, the most common architectures of convolutional neural networks for the task of image recognition, their structure and features were considered. As a result of the analysis of the architectures, it was revealed that the convolutional neural network

ResNet-152 showed the best result in the task of image recognition, with indicator top-5 equal to 3.57%, which indicates a fairly accurate definition of the object.

A feature of the ResNet architecture is that convolutional layers have 3 × 3 filters, and also the fact that a fast connection has been added to the network, which turns the network into its residual version.

References

[1] Sikorsky O.S. Review of convolutional neural networks for the problem of image classification. New information technologies in automated systems. 20. 37–42.

[2] Perceptron. Wikipedia. Last modified 2021. https://en.wikipedia.org/wiki/Perceptron.

[3] Bengio Yoshua, and Yann LeCun. Convolutional Networks for Images, Speech and Time Series, in M.A. Arbib, Ed., The Hand Book of Brain Theory and Neural Networks. Cambridge, MA: MIT Press, 1995.

[4]. Haykin, Simon. Neural Networks: Complete Course, 2nd Edition. Publishing house “Williams”, 2006.

[5] Simonyan Karen, Andrew Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. International Conference on Learning Representations, ICLR. (2015). ArXiv preprint arXiv:1409.1556.

[6] Loffe Sergey, Christian Szegedy. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. ArXiv e-Prints. International conference on machine learning (2015). 448-456.

[7] He Kaiming, Zhang Xiangyu, Shaoqing Ren, Sun Jian. Deep Residual Learning for Image Recognition. Proceedings of the IEEE conference on computer vision and pattern recognition (2016). 770-778.

[8] Lee Chen-Yu, Patrick W Gallagher, Zhuowen Tu. Generalizing Pooling Functions in Convolutional Neural Networks: Mixed, Gated, and Tree, Artificial Intelligence and Statistics (2016). 464-472.