UDC: 004.93:004.8

R. Kovalchuk, O. Polshakova

CNN FOR SOLVING COMPUTER VISION TASKS

Abstract: The paper explores fundamental and current methods addressing computer vision tasks, particularly in classification, segmentation, and image processing implemented in computer vision (CV) systems. It is highlighted that the application of deep learning significantly enhances the speed and accuracy in processing large datasets within CV systems. It is noted that artificial vision systems are proficient in object detection, recognizing serial numbers, and detecting surface defects. Emphasis is placed on the significance of integrating diverse methods for effectively addressing CV tasks. Directions for further research are proposed, such as the application of CV systems in resource-constrained environments and enhancing real-time processing efficiency.

Keywords: artificial intelligence, computer vision, classification, deep learning, neural networks, convolutional neural networks, object recognition.

Introduction

We live in times of a new technological revolution, similar to the transformations that occurred in the past with industrialization or the spread of PCs and the Internet. This accelerates development, changes our lives, and improves productivity. One of the main catalysts of this revolution is the active use of neural networks in many fields. They are transforming forecasting, data analysis, and decision-making, opening new possibilities for automating various processes.

Convolutional Neural Networks (CNNs) indisputably stand as a pivotal stage in further development and scientific inquiry. Their application extends beyond the aforementioned domains; they are also integrated into mobile robotics to facilitate localization and interaction between machines and the surrounding environment. This unveils new opportunities for autonomous transportation systems, industrial frameworks, and numerous other facets of our everyday lives by assuming mundane tasks.

1. Problem Statement

Computer Vision (CV) is an essential field of artificial intelligence aimed at analyzing, classifying, and recognizing images, videos, and graphical data. CV employs Deep Learning algorithms to identify objects, distinguish patterns, and unveil regularities. These systems mimic human vision properties by recognizing patterns and markers in visual data. They combine image processing, analysis, and recognition methodologies, enabling work with large datasets, including image sequences and three-dimensional data.

While Computer Vision has made significant strides, its relevance persists due to unresolved issues. Existing algorithms aren't universally applicable, and enhancing processing

speed often compromises quality or significantly escalates computational resources. Key research directions involve improving algorithm processing speed, implementing them in real systems, ensuring functionality in resource-constrained environments, and refining neural networks for more precise and expanded real-time object recognition.

Within the context of computer vision, three primary problem domains encompass recognition, identification, and detection. Recognition involves defining objects within their positions in an image or scene. Identification entails comparing biometric data of an object, such as eye color or hand geometry. Detection is applied to determine conditions, like changes in cells during medical examinations or obstacles in robotics systems for localization and interaction with the environment through visual information or the creation of maps of terrain (Fig. 1).

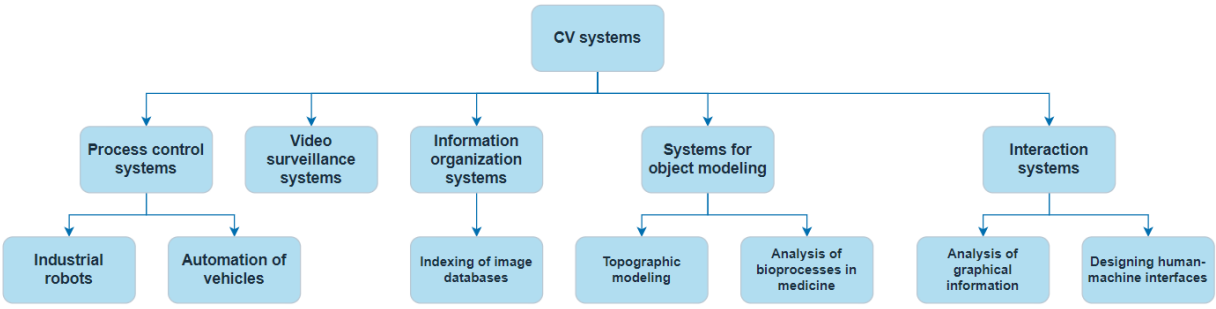

Fig. 1. Varieties of popular CV systems [1]

Various algorithms and methods are employed to address these challenges, including fuzzy logic, neural networks, hidden Markov models, and Convolutional Neural Networks (CNNs). This article will extensively analyze the architectures of Convolutional Neural Networks and their effectiveness in solving computer vision tasks.

2. Analysis of recent research and publications

Currently, research in the field of computer vision is being conducted both internationally and in Ukraine [3-4], where scientists are utilizing deep learning methods to develop and explore CV methods and models. Particularly noteworthy advancements have been made in the military domain by integrating these systems into FPV kamikaze drones. One of the established directions initiated by researchers is Deep Learning, which is based on the implementation of neural networks, specifically Deep Neural Networks (DNN) (Fig. 2).

The essence of training DNN models lies in selecting an optimal method based on mathematical transformations of processing input data to obtain an output result, regardless of linear or nonlinear correlations. Such training can occur with the involvement of a teacher (supervised learning), without one (unsupervised learning), or in a hybrid form combining both methods.

Comparing different models of deep neural networks studied by researchers can be quite challenging due to their varying working principles and structures. However, in modeling, it's crucial to evaluate the efficiency of their application on different datasets or

 ISSN 1560-8956

subsets of a specific dataset. Currently, for addressing computer vision tasks, convolutional neural networks (CNNs) are widely utilized, proving to be the best solution for working with images and audio files. Among the most successful models applied for object detection, classification, and image analysis are AlexNet, ResNets, EfficientNets, YOLO, R-CNN, LambdaNetworks, and VGG.

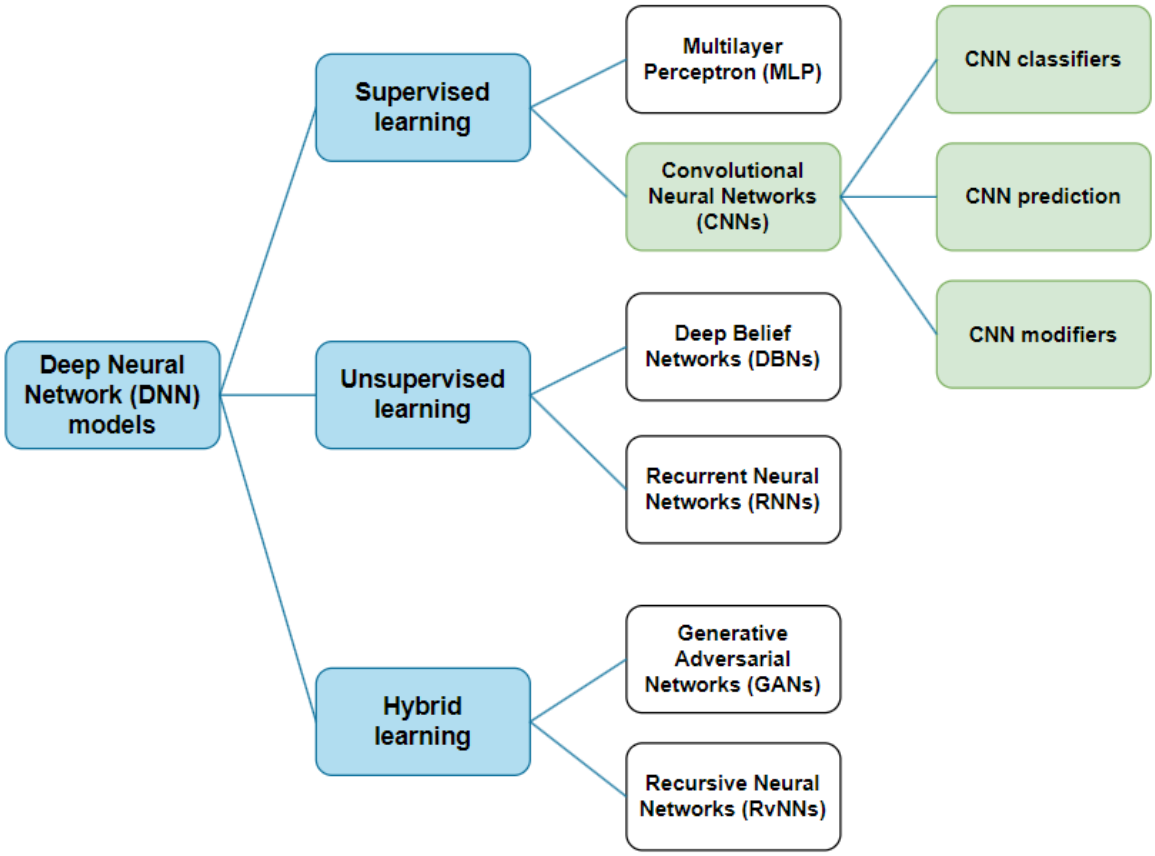

Fig. 2. Classification of Deep Neural Network (DNN) models in computer vision tasks [4]

These models are known for their efficiency and successful application in solving various computer vision tasks, including object detection, image classification, and analysis. The diversity of deep learning models provides a foundation for further research and improvement of approaches by combining the strengths of each approach.

3. Objective and Considered Questions

This article aims to conduct an analysis of contemporary solutions related to Convolutional Neural Networks (CNNs) to uncover their functionality, operational principles, and comparative assessment for selecting an optimal topology that optimizes solving typical computer vision (CV) tasks. The primary focus lies in understanding the construction, functioning, and utilization of these networks, achieved through addressing the following key inquiries:

– Defining specific tasks (classification, prediction, modification) addressed by convolutional neural networks.

– Identifying constraints arising from these tasks (speed, training accuracy, computational resource utilization) and their impact on model selection.

– Determination of input data types (images, sound), their dimensions, format (e.g., 100x100, 40x40, 30x30, RGB, or Grayscale), as well as the characteristics of output data such as the number of classes.

– Definition and selection of convolutional network parameters, including the quantity and type of convolutional, subsampling, connected layers, feature maps, kernel sizes, as well as the learning models utilized for activation and loss functions.

The primary goal involves an in-depth exploration and review of classification methods based on well-known convolutional neural networks specifically tailored for image processing and addressing computer vision tasks. Such an analysis aims to pinpoint the optimal model for a particular task (recognition), considering the limitations and characteristics of input and output data.

4. Results of CNN Solutions Review

The analysis of Convolutional Neural Networks (CNNs) confirms their high efficiency in classification tasks. CNNs have demonstrated superior performance, evolving sequentially from the cognitive topology of Fukushima's neocognitron. Scientists regard these CNN networks as the most optimal in terms of precision and learning speed, especially concerning object detection algorithms in space.

These networks offer partial resilience to scale changes, shifts, rotations, perspective alterations, and other displacements. Their architecture is grounded on three fundamental concepts: scaling changes, shift rotations, and spatial translations. These concepts enable CNNs to:

- Local Receptive Fields: Neurons within the network interact within limited regions, enabling the detection of local correlations between pixels in an image.
- Shared Synaptic Weights: This architecture identifies specific features within any spatial image while minimizing the overall number of weight coefficients, thus contributing to optimization.
- Hierarchy of Spatial Subsets: CNNs organize a hierarchy of abstraction levels for selective data, aiding in detecting more intricate functional characteristics.

These networks consist of various types of layers such as convolutional, subsampling, and fully-connected layers that can be sequentially altered in their output. The convolution operation serves as a fundamental operation in CNNs for image processing and involves using a kernel for weighted mapping of pixels in an image. The latter two types (convolutional, subsampling) alternate and shape the input feature vector in fully-connected layers (F-layers).

The primary types of layers in convolutional neural networks are convolutional, pooling, and fully-connected layers. Each of these layers performs specific operations on data and influences the final classification result. The implemented convolution operation in CNN is used for image processing and can be described by the following formula [4]:

$$C_{i,j} = \sum_{u=0}^{m_x-1} \sum_{v=0}^{m_y-1} A_{i+u} B_{u,v} ,$$

$C_{i,j}$ represents the calculated value of a pixel in the processed image; $B_{u,v}$ denotes the value of a convolution kernel element (u, v); A_{i+u} stands for the value of a pixel in the input image, where $m_x - 1, m_y - 1 -$ refers to the size of the convolution kernel.

The convolution operation is applied to pairs of matrices: A, which has dimensions of $n_x \times n_y$, and B with dimensions of $m_x \times m_y$. Hence, each pixel is computed by the scalar product of matrix B and a specific subset of matrix A, forming the resulting matrix C = A·B with dimensions $(n_x\text{-}m_x + 1)\cdot(n_y\text{-}m_y + 1)$ [4].

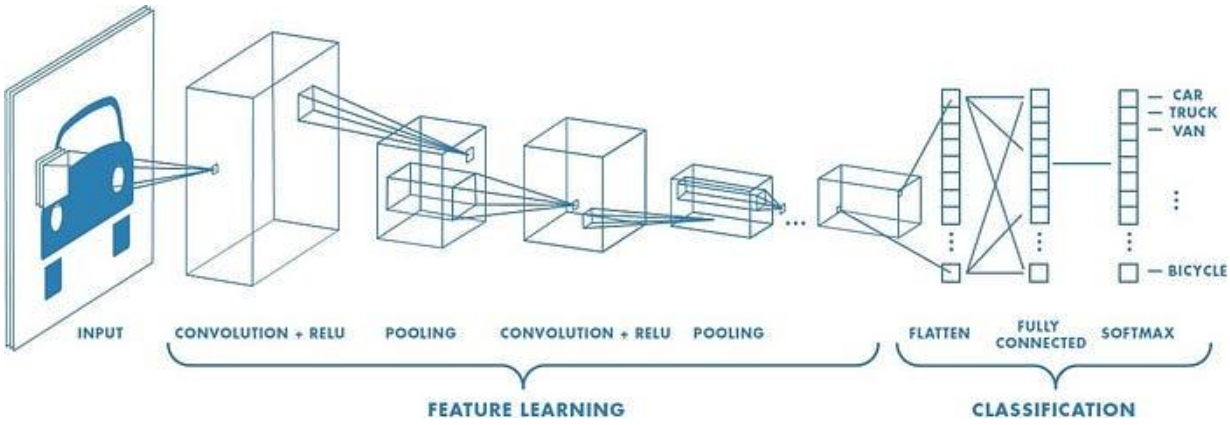

Fig. 3. General architecture of a Convolutional Neural Network (CNN) [1]

In CNNs, the algorithm for the convolution operation can be described as follows: the input image A is scanned by the kernel B with a specified step on a certain map (image); at each iteration, the element-wise summation of results is performed and transmitted into matrix C (Fig. 4). Thus, depending on the chosen method for processing, the output matrix size can be smaller (valid)/larger (full) than the input image or even match the dimensions of the working image.

In the convolutional layer of a neural network, convolution is applied to the outputs from the previous layer. The weight coefficients of the convolution kernel are learning parameters, to which an additional weight coefficient in the form of a constant bias offset is added. When modifying a neural network, it's important to consider the following parameters:

1. In a convolutional layer, there can be multiple convolutions. For instance, when an image of size w × h is input and n convolutions with a kernel size of $k_x \times k_y$are specified, the output will result in an image of size $n\cdot(w-k_{x+1})\cdot(h-k_{y+1})$.

2. Typically, convolutional kernels are three-dimensional, combining information from the R-, G-, B-channels of the input color image. This implies that in the first layer, a convolution of size d × w × h can be used, resulting in only one formed image at the output of this layer.

3. Creating padded images is crucial. Using convolution reduces the size of the image; therefore, in convolutional layers, padded images are employed. Outputs from the preceding layer are padded with pixels so that after convolution, the original image size is retained. Padded convolutions are termed "same convolution," while convolutions without padding are termed "valid convolution."

4. The stride in the convolutional layer is also significant. Normally, convolution operates on each pixel, but at times a stride other than 1 is used. In such cases, the scalar product calculation takes into account not all possible kernel positions but only those that are multiples of a certain stride, s. For instance, if the input image's dimensions are w × h, the convolution kernel size is $k_x \times k_y$, and a stride s is used, the resulting output image size is determined as:

$$\left|\frac{w + k_x}{s}\right| + 1 \cdot \left|\frac{h + k_y}{s}\right| + 1$$

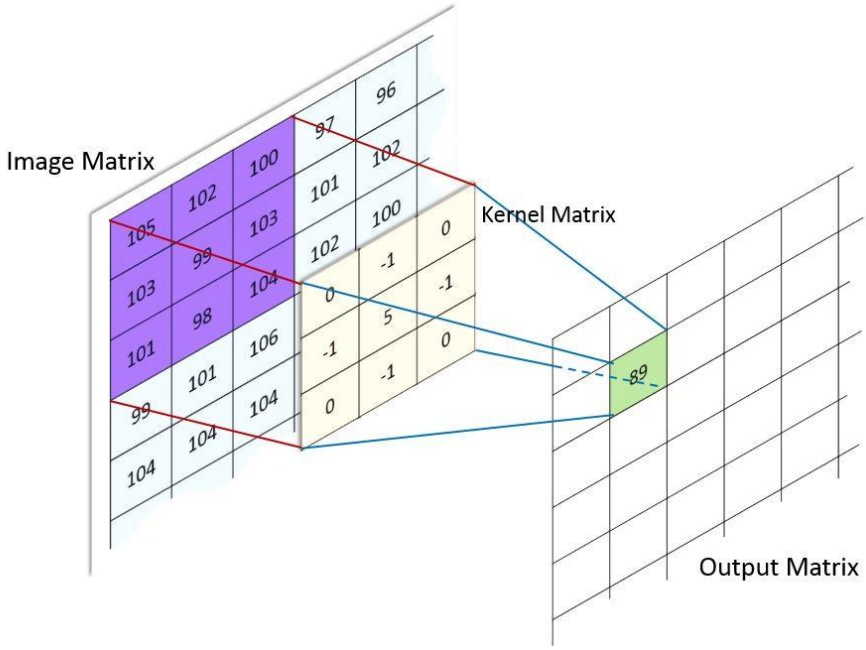

Fig. 4. Convolution Operation and Obtaining Convolutional Map [5]

The pooling layer can reduce the dimensionality of an image. The output image is divided into blocks of size w × h, and each of these blocks is computed using a specific function. Typically, functions like maximum pooling or (weighted) average pooling are employed. Integrating a pooling layer in a network allows:

 ISSN 1560-8956

- Reducing the dimensionality of the image before subsequent convolutional layers, enhancing the accuracy in recognizing larger areas within the original image.
- Increasing the invariance of the neural network's output layer concerning the introduced data and their characteristics.
- Accelerating data processing.

The fully-connected layer, known as the Inception module, introduced within the GoogLeNet network, is a concept proposed by researchers [6]. Each element in the previous layer corresponds to a specific section of the output image, and each convolution applied to these elements gradually expands the area of the output image until the elements in the final layers represent the entire output image. However, when convolutions reach a size of 1×1, recognition elements are not directly on the output image. To address this, the "inception module" layer was developed. The configuration of convolutional networks leads to a sudden increase in the number of layers, complicating the construction of deep neural network models. To tackle this issue, authors suggest using a modified "inception module" layer with additional dimension reduction. A 1×1 convolutional layer is added to each filter on the image, merging all convolutional layers into one. This approach ensures the retention of all characteristics when recognizing images using a limited number of convolutional layers.

In various scientific works, several other types of convolutions are used in convolutional neural network architectures:

– Dilated convolution: This type of convolution allows for exponentially expanding the receptive field, increasing it without losing image quality and preserving network resources.

– Partial convolution: This convolution enables processing the input image using a binary mask, acting as an additional feature during recognition. For example, utilizing a mask indicating pixel occlusion resolves the problem of partially fitting/restoring the image in selected areas.

– Gated convolution: This convolution type retains additional features from the input image across convolutional layers according to a defined mask. Instead of using a fixed mask that updates based on certain rules, gated convolution learns to automatically determine the mask from the provided data. This approach dynamically selects features in each logical region of the mask on the image, significantly enhancing the quality of recognized data.

Deep convolutional neural networks have proven more effective for image classification than other network types. Applying a multi-layered feedforward approach in CNNs enables the extraction of low-level, mid-level, and high-level features. Increasing the number of layers leads to enhanced feature hierarchies, significantly impacting the quality and accuracy of recognizing complex and extensive images. Let's examine several well-known CNN models and identify their architectural specifics:

LeNet 5 [7]: This network comprises five internal layers, including three convolutional layers (with feature maps of sizes 6, 16, and 120) and two fully connected

layers. Each convolutional layer utilizes a 5x5 pooling size with a stride of 1 and is activated by the hyperbolic tangent function. LeNet 5 achieves a 5% validation accuracy and attains 91.5% training accuracy using categorical cross-entropy as the loss function.

AlexNet [8]: It consists of two loosely interacting parallel parts, allowing their simultaneous use on different GPUs, thereby further parallelizing computations. Overall, the network comprises 60 million parameters and 650 thousand neurons, incorporating five convolutional layers, max-pooling layers, three fully connected layers with softmax activation. Additional activation functions are employed in the convolutional layers to accelerate learning, while regularization methods are utilized to prevent overfitting in the fully connected layers.

VGGNet [9]: VGG-13, VGG-16, and VGG-19 networks support 13/16/19 layers and can classify images into 1000 object categories. One of the advantages of this network is its topology simplicity. It includes multiple convolutional layers grouped into one to reduce image size and utilizes 64 filters that can be increased to 128/256 in certain cases if required. The later layers employ 512 filters. The VGG16 model was trained using Nvidia Titan Black GPUs over several weeks.

GoogLeNet (Inception Network): The core structure of this network involves Inception modules and comprises 22 parameter layers that self-adjust, along with 5 pooling layers. This network isn't distinguished by high training accuracy but is efficient in terms of size and computational complexity. Implementing 9 Inception blocks in the AlexNet network reduced the input parameters by nearly 10 times, enhancing computational speed and improving image recognition compared to previous CNN models.

ResNet: This network impresses with its depth, boasting over 150 layers [10]. Each ResNet block has two depth levels in ResNet 18/34 or three depth levels in ResNet 50/101/152. In the 50-layer ResNet, each 3-layer block can be substituted with a 34-layer network, where two size increments can be utilized. This architecture maintains structural simplicity even after increasing the depth to 152 layers. Compared to VGG-16/19 type networks, ResNet proves to be more efficient, particularly the 152-layer version, which has fewer FLOPs (floating-point operations) and offers a simpler structure.

Table 1. Key performance parameters of popular convolutional neural networks

Model Name	Training Time (s)	Learning Speed (s)	Training Accuracy (%)	Loss Function (%)
LeNet 5	122.73	0.001	98.4	0.44
AlexNet	162.75	0.010	84.5	15.4
VGGNet	>500.00	0.900	92.76	7.34
GoogLeNet	102.5	0.001	93.30	6.71
ResNet	150.5	0.001	96.43	3.56

 ISSN 1560-8956

5. Conclusions and Recommendations

Summarizing the review of CNNs, it's important to note certain challenges: while increasing their depth initially enhances processing accuracy, this accuracy diminishes significantly afterward. However, the decline in training accuracy complicates the optimization process. To enhance the characteristics of CNNs, recommendations often include using methods like dimension reduction, batch normalization, augmentation, and increasing sample size, advanced optimization techniques, and various activation functions. Additionally, when modeling CNN architectures, it's crucial to consider classification issues in computer vision tasks. Combining multiple methods could yield better results in these tasks. Currently, research is focused on addressing challenges related to applying CV systems in resource-constrained environments while maintaining satisfactory real-time accuracy, aiming for integration into diverse portable systems and mobile robotics.

REFERENCES

1. *Zahra Elhamraoui*, Introduction to convolutional neural network, https://medium.com/analytics-vidhya/introduction-to-convolutional-neural-network-6942c189a723, May 28, 2020

2. *Jon Krohn, Gnand Beyleveld, Aglae Bassens* // Deep Learning Illustrated: A Visual, Interactive Guide to Artificial Intelligence, Pearson Education, Inc, 2020.

3. *Kovalchuk A.M., Marchuk G.V., Marchuk D.K.* Application of a convolutional neural network for recognition of handwritten symbols // coll. of science pr. "Scientific notes of TNU named after V.I. Vernadsky". Series: technical sciences, Volume 30 (69) Part 1 No. 4 2019, p. 68-73.

4. *Tymchyshyn R.M., Volkov O.E., Gospodarchuk O.Yu., Bogachuk Y.P.,* Modern approaches to solving computer vision problems // coll. of science pr. "Control systems and computers", USyM, 2018, No. 6, p. 46-73.

5. *Theodore Bluche*, São Paulo Deep Neural Networks – Applications in Handwriting Recognition Meetup - 9 Mar. 2017.

6. *Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, Andrew Rabinovich,* Going Deeper with Convolutions // Computer Vision Foundation, 2015, https://www.cs.unc.edu/~wliu/papers/GoogLeNet.pdf 1-9

7. *Kaiming He Xiangyu Zhang Shaoqing Ren Jian Sun, Deep Residual* Learning for Image Recognition // https://arxiv.org/pdf/1512.03385.pdf 10 Dec 2015.

8. *Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton*, ImageNet Classification with Deep Convolutional Neural Networks // Communications of the ACM, June_2017,pp.84–90,

https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf.

9. *Gaudenz Boesch*, VGG Very Deep Convolutional Networks (VGGNet)

10. *Jon Krohn, Gnand Beyleveld, Aglae Bassens* // Deep Learning Illustrated: A Visual, Interactive Guide to Artificial Intelligence, Pearson Education, Inc, 2020.