Prerequisites for developing the computer vision system for drowning detection

Eduard Kozlov, and *Ruslan* Gibadullin*

Department of Computer Systems, Kazan National Research Technical University named after A. N. Tupolev – KAI, Kazan, Russia

Abstract. The problem of drowning is one of the most serious issues in terms of human life and health safety. Drowning victims often fall victim to accidental circumstances or unintentional actions, and the number of such incidents is significant. Thousands of drownings occur worldwide every year, resulting in a significant loss of lives, both among adults and children. This article focuses on exploring the prerequisites and developing a computer vision system for drowning detection. Drowning is a serious problem that poses substantial social, economic, and medical consequences for human life and health. The article discusses key computer vision technologies, image processing algorithms, and object recognition methods.

1 Introduction

In this review, we cover key methods and approaches to image analysis that are applied in the field of computer vision. By now, a multitude of different algorithms for image processing have been developed, including:
- Convolutional Neural Networks (CNN)
- Recurrent Neural Networks (RNN)
- Deep Generative Probabilistic Models
- Encoder-based Neural Networks

However, Convolutional Neural Networks (CNN) have particularly excelled in image classification tasks and object detection on images [1, 2, 3]. Their success can be attributed to their ability to consider the two-dimensional structure of an image, which is an advantage over multi-layer perceptrons. CNNs designed specifically for image analysis consist of several layers, each responsible for processing the image and extracting specific features. These networks employ three key architectural ideas to operate robustly against scale variation, rotation, translation, and spatial distortions. These ideas include the use of convolutional layers for image processing, subsampling for reducing spatial dimensionality, and activation layers for data normalization. Therefore, they utilize:
- Local receptive fields, providing local two-dimensional connectivity between neurons.
- Shared synaptic weights, enabling the detection of specific features anywhere in the image and reducing the overall number of weight coefficients.

* Corresponding author: landwatersun@mail.ru

- Hierarchical structure with spatial subsampling [4, 5, 6].

2 Classification of a drowning person and training

A drowning person may exhibit several signs that can be used to determine their condition and call for help. Some of these signs may include:
- Lack of movement: If a person is not attempting to swim or not moving in the water, it may indicate the possibility of drowning.
- Body posture and position: If a person's body is not floating or has an unnatural position, it may indicate the possibility of drowning.
- Attempts to reach the surface: If a person is trying to reach the surface but fails to do so, it may be a sign of drowning.
- Unnatural poses: If a person is in an unnatural pose, it may indicate that they cannot control their position in the water.
- Silent cries for help: If a person is in distress, they may cry for help, but their cries may be silent or barely audible [7, 8, 9].

These signs can be used to train a convolutional neural network (CNN) that can determine when a person is in danger and need of assistance. However, not all drowning individuals exhibit all the above-mentioned signs, so a comprehensive approach is needed for more accurate classification, taking into account other factors such as age, gender, physical fitness, and more.

Training a neural network requires a large dataset of images depicting drowning individuals and individuals in the water without the threat of drowning. Each image needs to be labeled as "drowning" or "safe" to allow the network to learn to distinguish between the two classes.

After training the network on such data, it can classify new images depicting people in the water. A computer vision system can analyze these images for signs of drowning, such as lack of movement, unnatural poses, changes in body shape, and other characteristics.

If the system detects signs of drowning, it can alert the possibility of drowning and call emergency services to save lives.

It is important to note that for the effective operation of a computer vision system, it needs to be trained on a large and diverse dataset to ensure high accuracy in recognizing drowning signs and avoiding false alarms [10, 11, 12].

3 Structure of a Convolutional Neural Network

Convolutional neural networks (CNNs) consist of several types of layers: convolutional layers, pooling layers, and fully connected layers. In a CNN, these layers are typically arranged sequentially, with convolutional and pooling layers alternating, ultimately leading to a feature vector as input to the fully connected layers. The name "convolutional network" stems from the use of the convolution operation in the convolutional layers.

CNNs strike an effective balance between biologically inspired neural networks and classical multilayer perceptrons [13, 14, 15]. They have become one of the most efficient tools for image analysis today. An important aspect contributing to their success is the use of shared weight coefficients, which significantly reduces the number of parameters requiring adjustment.

CNNs are characterized by high speed in both computation and training, achieved through the ability to perform parallel processing during convolution and backpropagation (see Fig. 1).

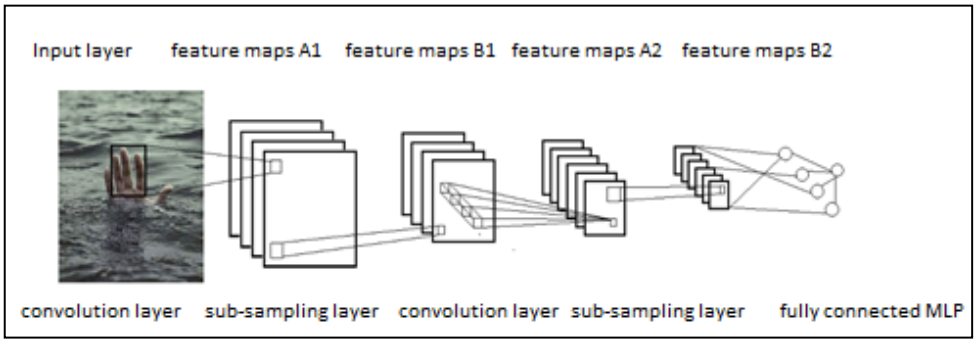

Fig. 1. Topology of a convolutional neural network.

3.1 Topology of a convolutional neural network

The topology of a convolutional neural network (CNN) typically consists of multiple convolutional layers, each containing several filters to extract features from input images, as well as pooling layers to reduce dimensionality and computational complexity. Following these layers are fully connected layers that connect the extracted features to the output classes. In the end, a softmax activation function is commonly used to convert the output values into probabilities for each class [16, 17, 18]. The number and sizes of the layers can vary depending on the specific task and data.

The choice of a neural network's topology depends on the task it aims to solve and the constraints imposed on that task, such as response speed and accuracy. It is necessary to determine the type and format of the input data (e.g., images, sound) and the output data (e.g., number of classes). For my image classification task (faces), I have chosen a specific topology, taking into account constraints on response speed and recognition accuracy (no more than 1 second and no less than 70% accuracy, respectively) (Fig. 2).

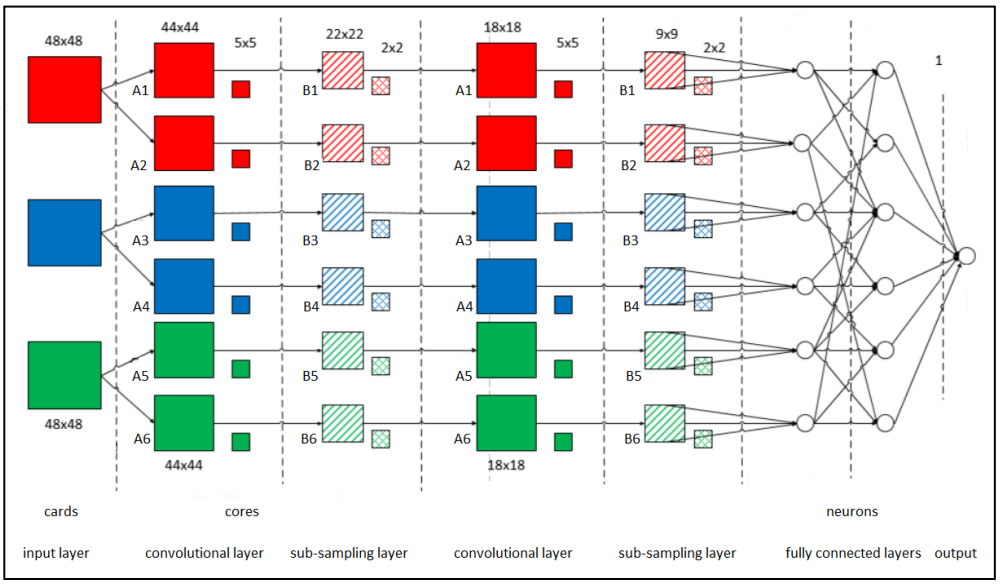

Fig. 2. Topology of a convolutional neural network.

3.2 Input Layer

The input layer of a convolutional neural network (CNN) represents the image matrix, where each pixel is treated as an individual input parameter of the network [19, 20, 21]. Pixels can be divided into multiple channels, for example, three channels are used for color images - red, green, and blue. The input layer can also contain multiple feature maps, each corresponding to a specific channel. Before processing, the input data is typically normalized to bring them within the range of values from 0 to 1.

3.3 Convolutional Layer

The convolutional layer of a convolutional neural network performs convolution on the input data using a set of filters (convolutional kernels) that scan the image and extract its features. Each filter is a small matrix that moves across the image with a certain stride, multiplying with pixel values. The result of the multiplication is recorded in a matrix called a "feature map" [22, 23, 24], which is then passed to the next layer of the network. (Fig. 3)

The convolutional layer provides partial spatial invariance to transformations such as translation, rotation, and scaling, making convolutional neural networks effective for image classification and object recognition. They also aid in capturing local features such as edges, textures, and shapes, enabling the network to recognize objects even if they are located in different parts of the image.

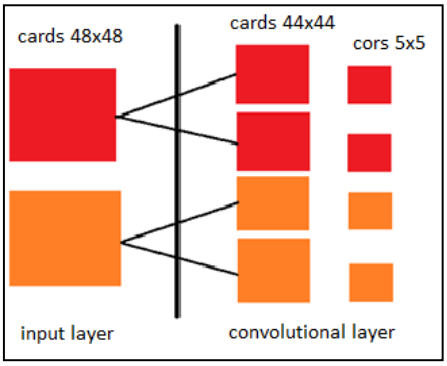

Fig. 3. Relationships between maps of the convolutional layer and the previous one.

The convolutional layer is a fundamental component of a convolutional neural network. This layer contains a kernel or filter that moves across the feature map of the previous layer, searching for specific attributes of objects, such as facial details. The kernel serves as a collection of shared weights, which facilitates reducing the number of connections between neurons and increases the speed of feature detection.

The size of the kernel is chosen to avoid losing essential information when downsampling in the pooling layer. Typically, the kernel size ranges from 3×3 to 7×7 pixels. A smaller kernel size may limit the ability to extract features, while a larger size increases the number of connections between neurons.

The area of the image marked by a red square (see Fig. 4) demonstrates a high response, indicating the presence of a specific feature in that region of the image.

In the initial phase of the convolutional layer, all values in its feature map are initialized as zeros. Then, the kernel weights are randomly selected within the range of -0.5 to 0.5. From this point, the kernel starts moving across the previous feature map with a certain stride, and at each position, a convolution operation is performed.

The convolution operation involves stepping a window, the size of which matches the kernel, through the entire image. The content of the window and the kernel are element-wise multiplied, and the results are summed and recorded in a new matrix. This process is repeated to obtain a new convolutional feature map. The kernel moves across the image, performing convolutions at each step, which allows the formation of a new feature map of the same size as the original one.

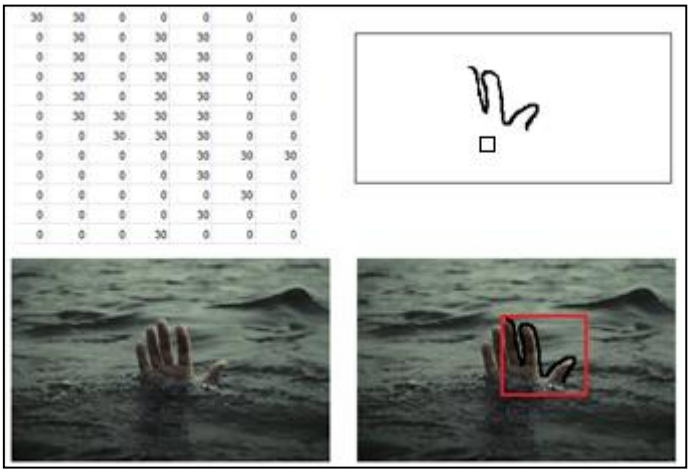

Fig. 4. Core with the trained feature.

However, the resulting size of the formed feature map may differ from the size of the original image. This depends on the chosen approach for handling the borders of the original matrix, and as a result, the size of the obtained feature map can be smaller, equal to, or even larger than the original image.

3.4 Pooling Layer

The pooling layer is a layer in a convolutional neural network (CNN) that reduces the size of its input matrix, thereby reducing the number of parameters, and computation time, and preventing overfitting. To achieve this, the pooling layer divides the input matrix into multiple smaller blocks, and for each block, an aggregation operation is performed, which computes a single value to replace the input values in that block. For example, in the case of the maximum operation, the maximum element is selected from each block and used as the output value for the entire block.

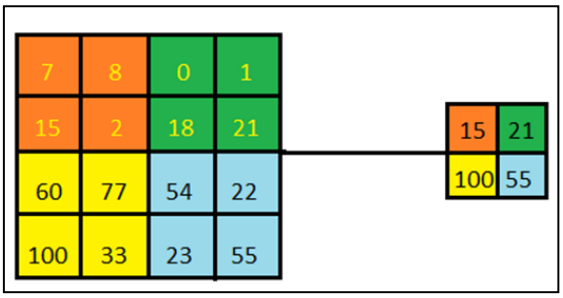

Fig. 5. Formation of a new map of the subsampling layer.

The most commonly used aggregation operations in pooling layers are maximum, average, or sum. The pooling layer can be added to a CNN after each convolutional layer to progressively reduce the size of the matrix and increase the level of abstraction in the output features (Fig. 5) [25, 26, 27].

3.5 Fully Connected Layer

The fully connected layer is a crucial component of a convolutional neural network (CNN) in which each neuron in this layer is connected to all neurons in the preceding layer [28, 29, 30]. This layer operates by receiving input data from all neurons of the previous layer with corresponding weights and then applying a non-linear activation function to these inputs.

Fully connected layers are typically employed in the final part of a CNN to perform classification or regression tasks [31, 32, 33]. Additionally, they can be used for tasks related to image generation and segmentation.

The number of neurons in the fully connected layer is determined based on the specific task and can be set based on experience or empirical observations. In general, increasing the number of neurons in the fully connected layer can enhance prediction accuracy. However, it can also lead to longer training times and an increased risk of overfitting.

3.6 Output Layer

The output layer of a convolutional neural network is the layer that predicts the value of the output variable for a given input.

Depending on the task being performed by the convolutional neural network, the output layer can take different forms. For example, in image classification tasks, the output layer can be a fully connected layer with softmax activation, which converts the output values from the previous layer into probabilities belonging to each class.

For regression tasks, the output layer can be a single neuron that predicts a continuous value for the output variable. In such cases, the activation function of the neuron can be linear or nonlinear, such as ReLU [34, 35, 36].

An important aspect of the output layer is the loss function, which measures the discrepancy between the predicted values and the actual values of the output variable. The choice of the loss function depends on the task at hand. For classification tasks, cross-entropy can be used, while mean squared error can be used for regression tasks [37, 38, 39].

4 Conclusion

Through the research, the prerequisites for developing a computer vision system for drowning detection have been identified. Early detection of drowning is an important task that can save many lives and prevent numerous tragedies [40, 41, 42].

The study examined key computer vision technologies used in computer vision systems, as well as the algorithm employed by a convolutional neural network. To develop the system, it is necessary to determine shape and positional features that can be used to identify a drowning person in an image. This article explored various image processing techniques, including convolutional neural networks, which enable automatic feature extraction for drowning detection.

Further improvement of the system is possible through training on a larger volume of diverse images and considering additional parameters such as weather conditions, lighting, and other factors that can influence the drowning detection process.

References

1. A. Krizhevsky, I. Sutskever, G. E. Hinton, Advances in Neural Information Processing Systems **25**, 1097-1105 (2012)
2. Y. LeCun, Y. Bengio, G. Hinton, Nature **521**, 436-444 (2015)
3. X. Yu, Z. Cao, Z. Wu, C. Song, J. Zhu, Z. Xu, 2022 17th International Conference on Control, Automation, Robotics and Vision (ICARCV) 659-664 (2022)
4. U. Handalage, N. Nikapotha, C. Subasinghe, T. Prasanga, T. Thilakarthna, D. Kasthurirathna, 2021 3rd International Conference on Advancements in Computing (ICAC), 240-245 (2021)
5. R. F. Gibadullin, M. Y. Perukhin, A. V. Ilin, 2021 International Conference on Industrial Engineering, Applications and Manufacturing (ICIEAM), 398-403 (2021)
6. O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, International journal of computer vision **115**, 211-252 (2015)
7. R. F. Gibadullin, G. A. Baimukhametova, M. Yu. Perukhin, 2019 International Conference on Industrial Engineering, Applications and Manufacturing (ICIEAM), 1-7 (2019)
8. I. Goodfellow, Y. Bengio, A. Courville, MIT Press (2016)
9. N. A. Staroverova, M. L. Shustrova, Yu. N. Zatsarinnaya, Journal of Physics: Conference Series **1399(4)** (IOP Publishing, 2019)
10. N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, The journal of machine learning research **15(1)**, 1929-1958 (2014)
11. R. Girshick, J. Donahue, T. Darrell, J. Malik, Proceedings of the IEEE conference on computer vision and pattern recognition, 580-587 (2014)
12. X. Glorot, A. Bordes, Y. Bengio, Proceedings of the fourteenth international conference on artificial intelligence and statistics, 315-323 (2011)
13. R. R. Muhamadiev, N. A. Staroverova, M. L. Shustrova, Journal of Physics: Conference Series **2032(1)** (IOP Publishing, 2021)
14. V. A. Raikhlin, I. S. Vershinin, R. F. Gibadullin, Journal of Physics: Conference Series **2096(1)**, 012160 (2021)
15. A. Kh. Rakhmatullin, R. F. Gibadullin, Lobachevskii Journal of Mathematics **43(2)**, 473-483 (2022)
16. S. N. Cherny, R. F. Gibadullin, 2022 International Conference on Industrial Engineering, Applications and Manufacturing (ICIEAM), 965-970 (IEEE, 2022)
17. V. A. Raikhlin, R. F. Gibadullin, I. S. Vershinin, Lobachevskii Journal of Mathematics **43(2)**, 455-462 (2022)
18. R. F. Gibadullin, M. Yu. Perukhin, B. I. Mullayanov, 2020 International Multi-Conference on Industrial Engineering and Modern Technologies (FarEastCon), 1-6 (IEEE, 2020)
19. R. F. Gibadullin, D. V. Lekomtsev, M. Y. Perukhin, Scientific and Technical Information Processing **48(6)**, 446-451 (2021)
20. R. F. Gibadullin, I. S. Vershinin, M. M. Volkova, 2020 International Multi-Conference on Industrial Engineering and Modern Technologies (FarEastCon), 1-7 (IEEE, 2020)
21. A. Semenov, I. Yakushev, Y. Kharitonov, V. Shevchuk, E. Gracheva, S. Ilyashenko, International Journal of Technology **11(8)**, 1537-1546 (2020)

22. S. R. Khasanov, E. I. Gracheva, M. I. Toshkhodzhaeva, S. T. Dadabaev, D. S. Mirkhalikova, E3S Web of Conferences **178**, 01051 (2020)
23. V. Dovgun, S. Temerbaev, M. Chernyshov, V. Novikov, N. Boyarskaya, E. Gracheva, Energies **13(18)**, 4915 (2020)
24. G. Marin, D. Mendeleev, B. Osipov, A. Akhmetshin, E3S Web of Conferences **178**, 01033 (2020)
25. Y. I. Soluyanov, A. I. Fedotov, D. Y. Soluyanov, A. R. Akhmetshin, IOP Conference Series: Materials Science and Engineering **860(1)**, 012026 (2020)
26. A. Kryukov, K. Suslov, L. Van Thao, T. D. Hung, A. Akhmetshin, Energies **15(21)**, 8249 (2022)
27. R. Zaripova, I. Gaisin, M. Tyurina, O. Rocheva, E. Kubyshkina, Proceedings of the International Symposium on Sustainable Energy and Power Engineering, 319-327 (2021)
28. M. Tyurina, A. Porunov, A. Nikitin, R. Zaripova, and G. Khamatgaleeva, Proceedings of the International Symposium on Sustainable Energy and Power Engineering, 391-402 (2021)
29. O. Soloveva, S. Solovev, R. Zaripova, F. Khamidullina, M. Tyurina, E3S Web of Conferences **258**, 11010 (2021)
30. J. L. Ordoñez-Avila, I. A. Magomedov, A. M.Bagov, Journal of Physics: Conference Series **2094(4)**, 042093 (2021)
31. A. L. Zolkin, I. A. Magomedov, O. V. Kucher, 2020 International Multi-Conference on Industrial Engineering and Modern Technologies (FarEastCon), 1-6 (2020)
32. I. A. Magomedov, H. A. Murzaev, A. M. Bagov, IOP Conference Series: Materials Science and Engineering **862(5)**, (IOP Publishing, 2020)
33. Z. Gizatullin, M. Nuriev, 2022 International Conference on Industrial Engineering, Applications and Manufacturing (ICIEAM), 321-326 (IEEE, 2022)
34. Z. M. Gizatullin, R. M. Gizatullin, M. G. Nuriev, Journal of Communications Technology and Electronics **66(6)**, 722-726 (2021)
35. Z. M. Gizatullin, R. M. Gizatullin, M. G. Nuriev, 2020 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus), 120-123 (2020)
36. M. M. Lyasheva, S. A. Lyasheva, M. P. Shleymovich, Cyber-Physical Systems: Intelligent Models and Algorithms, Cham: Springer International Publishing, 233-244 (2022)
37. M. M. Lyasheva, S. A. Lyasheva, M. P. Shleymovich, 2021 International Russian Automation Conference (RusAutoCon), 256-260 (2021)
38. M. M. Lyasheva, S. A. Lyasheva, M. P. Shleymovich, 2021 International Russian Automation Conference (RusAutoCon), 448-452 (2021)
39. S. A. Solovev , O. V. Soloveva, I. G. Akhmetova, Y. V. Vankov, D. L. Paluku, ChemEngineering **6(1)**, 11 (2022)
40. O. V. Soloveva, S. A. Solovev, Y. V. Vankov, R. Z. Shakurova, Processes **10(11)**, 2257 (2022)
41. S. A. Solovev, O. V. Soloveva, D. L. Paluku, A. A. Lamberov, Chemical Product and Process Modeling **17(6)**, 583-602 (2022)
42. R. M. Shakirzyanov, A. A. Shakirzyanova, 2021 International Russian Automation Conference (RusAutoCon), 714-718 (2021)