International Journal of Engineering, Science and Mathematics
Vol. 12 Issue 12, Dec 2023,
ISSN: 2320-0294 Impact Factor: 6.765
Journal Homepage: http://www.ijesm.co.in, Email: editorijmie@gmail.com
Double-Blind Peer Reviewed Refereed Open Access International Journal - Included in the International Serial Directories Indexed & Listed at: Ulrich's Periodicals Directory ©, U.S.A., Open J-Gage as well as in Cabell's Directories of Publishing Opportunities, U.S.A

Deep Neural Network Pooling Techniques Using CNN

Mr. Rajendra Singh[1], Dr. Shiv Kant[2]
1 Ph.D. Research Scholar, Department of Computer Science and Engineering, School of Engineering and Technology, Raffles University, Neemrana
2 Associate Professor, Department of Computer Science and Engineering, School of Engineering and Technology, Raffles University, Neemrana
raj21engg@gmail.com, dr.shivkant@rafflesuniversity.edu.in

Abstract

Several scientific fields rely on Deep Neural Networks these days. A specific subset of DNNs, Convolutional Neural Networks have many convolution layers, an activation function, and a pooling layer at the bottom. To create new feature maps with a reduced resolution, the pooling layer applies downsampling to the feature maps received from the preceding layer. This layer plays a crucial role in the process. This layer significantly flattens the input's spatial dimension. There are two primary uses for it. The first is to decrease the computing cost by reducing the number of parameters or weights. The second is regulating the network's tendency to overfit. In a perfect world, pooling methods would glean only pertinent data while ignoring superfluous minutiae. In Deep Neural Networks, the pooling process may be implemented in many different ways. Several well-known and practical pooling algorithms were discussed in this work.

Keywords: Pooling Methods, Convolutional Neural Networks, Deep learning, Down-sampling

1. Introduction

Computers and other electronic gadgets rely on machine learning as their foundation for intelligence. To do this, it makes use of predictive models, which are able to draw on past data in order to make predictions about new events, trends, and behaviours. In deep learning, a kind of machine learning, mathematical representations of models modelled after the human brain are used. Deep Neural Networks (DNNs) automatically learn, from data, the parameters that define the mathematical models; these parameters can range in number from a few thousand to one hundred million or more. DNNs are able to represent input-output interactions that are complicated and non-linear. Their designs provide compositional models in which the item is represented as a layered assembly of primitives. There are a lot of different takes on a few basic techniques in deep architectures. Deep neural networks (DNNs) use hierarchical structures in an effort to learn data abstractions at a high level. Semantic parsing[1], transfer learning[2,3], natural language processing[4], computer vision[5,6], and many more classic AI disciplines have made extensive use of this developing method. The tremendous improvements in machine learning algorithms, the drastically reduced cost of computer gear, and the considerably enhanced processing capabilities of chips are the three primary causes for the current boom of deep learning [7]. A number of models utilising DNNs for various purposes have been put out in recent years. Convolution Neural Networks (CNNs), Restricted Boltzmann Machines (RBMs), Auto encoders, Sparse Coders, and Recurrent Neural Networks are the five main types of these models [8, 9]. For tasks like object segmentation and classification, convolutional neural networks (CNNs) are among the most essential and practical forms of DNNs. The convolutional, pooling, and fully linked layers make up a convolutional neural network (CNN). This stack performs distinct spatial operations. To construct the feature maps, CNN convolves the input picture using various kernels in the convolution layers. It is common practice to install the pooling layer subsequent to the convolution layer. Feature maps and network parameters are getting smaller thanks to this layer's use. A flatten layer comes after the pooling layer, and then there are a number of completely linked layers. To make them acceptable for the next fully linked layers, the flatten layer converts the 2D feature maps created in the preceding layer into 1D feature maps. Later on, the photos may be classified using the flattened vector.

“To lower the feature maps' dimensionality, pooling is an essential step in convolutional a-based systems. It reduces the dimensionality of the feature map, which means it merges a collection of values into a smaller set of values. It takes the joint feature representation and turns it into usable information by retaining key details and removing unnecessary ones. In addition to lowering the computational cost for higher layers by deleting certain connections between convolutional layers, pooling operators provide a sort of spatial transformation invariance. The feature maps from the preceding layer are down-sampled in this layer, resulting in new feature maps with a reduced resolution. One of the primary functions of this layer is to regulate overfitting; another is to decrease the computational cost by reducing the number of parameters or weights. In a perfect world, pooling methods would glean only pertinent data while ignoring superfluous minutiae. Several convolutional neural network (CNN) pooling techniques were examined in this paper. There are two types of pooling methods that we identified: the more common ones and the more unusual ones. Average Pooling, Max Pooling, Mixed Pooling, □□ Pooling, Stochastic Pooling, Spatial Pyramid Pooling, and Region of Interest Pooling are all covered in commonly used techniques. New approaches cover a wide range of topics, including multi-scale order-less pooling, PCA networks, compact bilinear pooling, lead asymmetric pooling, edge-aware pyramid pooling, mixed pooling, spectral pooling, row-wise max pooling, inter-map pooling, rank-based average pooling, per pixel pyramid pooling, weighted pooling, and genetic swimming. After that, the paper is structured like this: Popular pooling strategies are presented in Section 2. are covered in that section”.

2. Popular Pooling Methods

2.1. Average Pooling

For feature pooling and extraction, the concept of average or mean was initially presented in [10] and utilised in the first convolution-based deep neural network [11]. Averaging the values of the input's rectangular pooling sections allows an average pooling layer to execute down-sampling, as seen in Figure 1.

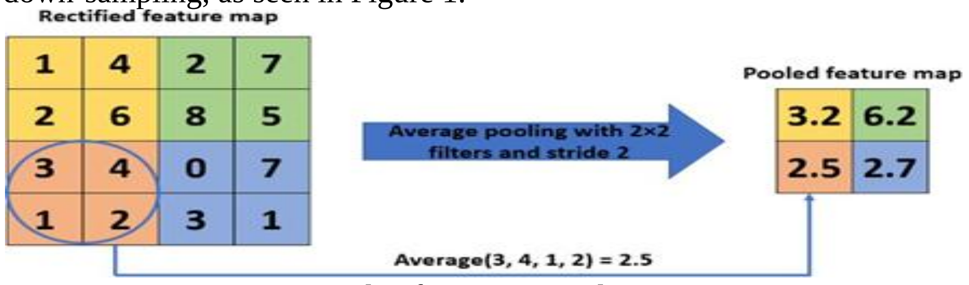

Fig. 1. Example of Average Pooling operation.

2.2. Max-Pooling

The convolutional output bands can be down-sampled using a max-pooling operator [12], which reduces variability. Within a set of □ activations, the max-pooling operator forwards the highest value. The □ related filters □□ = [□1, …, □□,□, …, □□,□] ∈ □ □ make up the □-th max-pooled band:

$$pj, = \max(h\ j,(m-1)N+r) \qquad (1)$$

when □ < □, the pooling shift □ ∈ {1, …, □} permits the overlap of pooling zones. The output dimensionality is reduced from □ convolutional bands to □ = (− □)/□ + 1 pooled bands by the pooling layer, and the resultant layer is □ = [□1, …, □□] ∈ □ □.□. Figure 2 shows a Max-Pooling procedure in action.

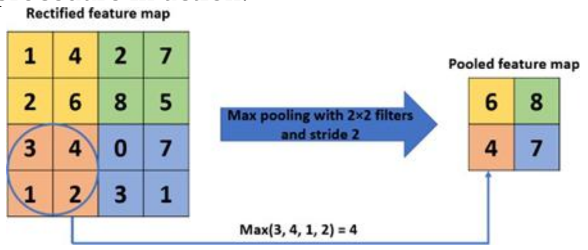

Fig.2. Exampleof Max-Poolingoperation.

2.3. Mixed Pooling

By integrating the non-maximal activations, average pooling reduces the activation, in contrast to max pooling, which only recovers the greatest activation. To get around this issue, Yu et al. [13] suggested a mixed method that combines average and max pooling. A lot of what we're doing here comes from Drop connects [15] and dropout [14]. Eq. 2 is a representation of mixed pooling:

The output will be: □□ = □ max □∈ □□ □□ + (1 − □) 1 |□□ | ∑ □□ □∈ □□ ,

where □ determines whether to use max pooling or average pooling. A random integer between zero and one is chosen as the value of □. At a value of 0, it operates similarly to average pooling, while at a value of 1, it mimics max pooling. The backpropagation method makes use of the recorded value of □ for forward-propagation order. By applying image classification to three separate datasets, Yu et al. demonstrated its superiority over max and average pooling.

2.4. *LP* Pooling

Claiming that it outperforms max pooling in terms of generalisation ability, Sermanet et al. [16] introduced the idea of □□ pooling. The inputs are averaged in the pooling region using a weighted formula in this pooling. The form it takes is provided by Eq. 3:

sj = (1 |*Rj* | ∑ *aipi*∈*Rj*) 1⁄*p* (3)

Where □□ is the pooling operator's output at location □ and □□ is the feature value at position □ within the pooling region □. There is a range of values for □ from 1 to infinity. The □□ operator follows the behaviour of average pooling when □ = 1, and max-pooling when □ = ∞. As a compromise between average and max pooling, □□ pooling is considered when □ > 1.

2.5. Stochastic Pooling

IZeiler and Fergus [17] first out the concept of stochastic pooling, which was influenced by the dropout [14]. By comparing each pooling zone, max pooling finds the one with the highest activation. “Whereas with average pooling low-activation areas are given more weight than high-activation ones, since all components in the pooling zone are looked at and their average is taken. A big issue with average pooling is this. With stochastic pooling, we can solve the problems with max and average pooling. In stochastic pooling, the value is randomly selected using a multinomial distribution. It contains the feature map's activations that are less than optimal”. As shown in Eq. (4), the initial step in stochastic pooling is to normalise the activations within each area □ and then calculate the probability □□ for each region □.

pi = *ai* ∑*k*∈*Rjak* (4)

Based on □, these probabilities form a multinomial distribution that chooses location □ and the matching pooled activation □□. The multinomial distribution chooses a spot □ inside the area:

$sj = alwh\ erel \sim (p1, \ldots, p|Rj|)$ (5)

Let me put it simply: the activations are chosen using probability computed using a multinomial distribution. In this, the odds of each activation are proportional to their respective probabilities. Overfitting is not allowed in stochastic pooling due to the random nature of the data. Stochastic pooling uses non-maximal activations and offers some of the same benefits as max-pooling.

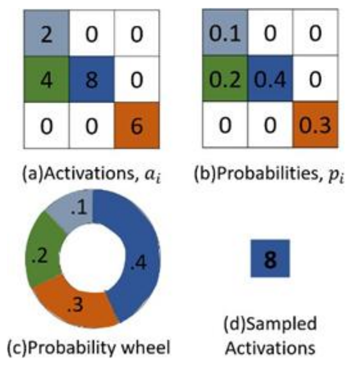

Fig.3.Exampleofstochasticpooling,(a)activationswithinagivenpoolingregion,(b)probabilitiesbasedonactivations,(c) probability wheel, (d) sampled activation

Keep in mind that the chosen element might not necessarily be the biggest one since stochastic pooling depicts the multinomial distribution of activations in the location. It promotes greater activations and downplays weaker ones. Figure 3 shows an instance of stochastic pooling in action.

2.6. Spatial Pyramid Pooling

"Spatial pyramid pooling is one of the new pooling layer approaches. As an expansion of the Bag-of-Words (BoW) model [20], one of the most effective approaches in computer vision is spatial pyramid pooling [18, 19], also known as spatial pyramid matching (SPM[19]). It takes the image and divides it into sections ranging from very fine to very coarse, then aggregates local characteristics within each section. He et.al. removed the network's fixed-size limitation by introducing a spatial pyramid pooling (SPP) [18, 19] layer in [21]. More precisely, they superimposed an SPP layer over the final convolutional layer. The fully-connected layers get their inputs from the SPP layer, which pools the features and produces outputs of a set length. So, to construct the YOLO detection technique without initial cropping or warping, Huang et.al. [22] aggregated some information between convolutional and fully-connected layers, which is farther down the network hierarchy. Figure 4 depicts a three-tiered spatial pyramid pooling layer".

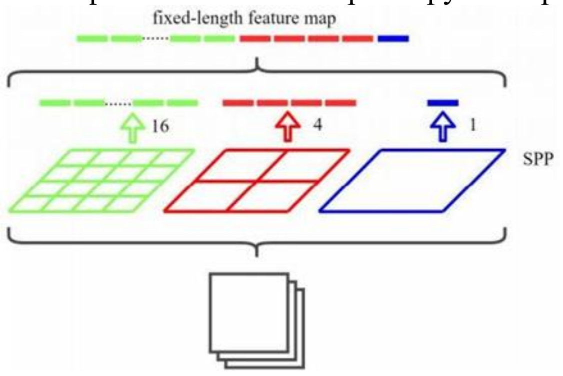

Fig. 4. Spatial pyramid pooling structure [23]

2.7. Region of Interest Pooling

Convolutional neural networks rely heavily on the Region of Interest (RoI) Pooling layer for tasks such as object identification [24] and segmentation [25]. One way the ROI pooling layer contributed to the overall network design was by deferring operations that was bounding-box specific to a later stage. After feeding an input picture into the deep network, it produces CNN feature maps that are intermediate in quality and have lower spatial dimensions than the original. The input feature map of the full picture and the coordinates of each ROI are sent into the ROI pooling layer. The features associated with a particular item may be roughly located using the ROI coordinates. However, due to the fact that each ROI might be of a different dimension, the features that are thus collected have variable spatial sizes.

An ROI pooling layer takes in feature maps of varying sizes (representing various object proposals) and outputs one feature map of fixed size for each proposal; this is necessary since CNN layers can only process inputs with a set number of dimensions. One hyperparameter that stays constant throughout training is the fixed-size output dimensions. In particular, each ROI is partitioned into a collection of identically sized cells to get this consistent output size. The quantity of these cells matches the necessary output dimensions. The next step is to find the place in the output feature map that corresponds to the greatest value in each cell, which is done via max-pooling.

A deep network's performance is substantially enhanced by the ROI pooling layer, which uses a single set of input feature maps to produce feature representations for each region proposal.

3. Novel Pooling Methods

3.1. Multi-scale order-less pooling (MOP)

“One method that has been suggested by Gong et al. [26] is multi-scale order-less pooling, or MOP. The discriminative capacity of CNNs is unaffected by this pooling strategy, which increases their invariance. To get the deep activation characteristics, MOP processes the whole signal as well as local patches. In order to have a better idea of the overall spatial layout, we record the activation characteristics of the complete signal. To get a better idea of the finer details in the picture and to make sure that everything stays the same, we record the activation features of individual patches. One way to combine activation characteristics from different patches is to utilise vectors of locally aggregated descriptors (VLAD) encoding [27]. At various sizes, this pooling layer extracts deep activation characteristics from local patches to start its work. because it catches more local, fine-grained features of the picture at higher scales and keeps the global spatial arrangement at coarser sizes, which is the entire image. The results were then aggregated at the smaller scales using VLAD encoding of the local patches [27]. A more invariant representation may be constructed with the aid of VLAD due to its order-less nature. Lastly, it creates a new picture representation by merging the initial global deep activations with the VLAD characteristics designed for smaller scales”. You can see how this pooling mechanism works in Fig. 5.

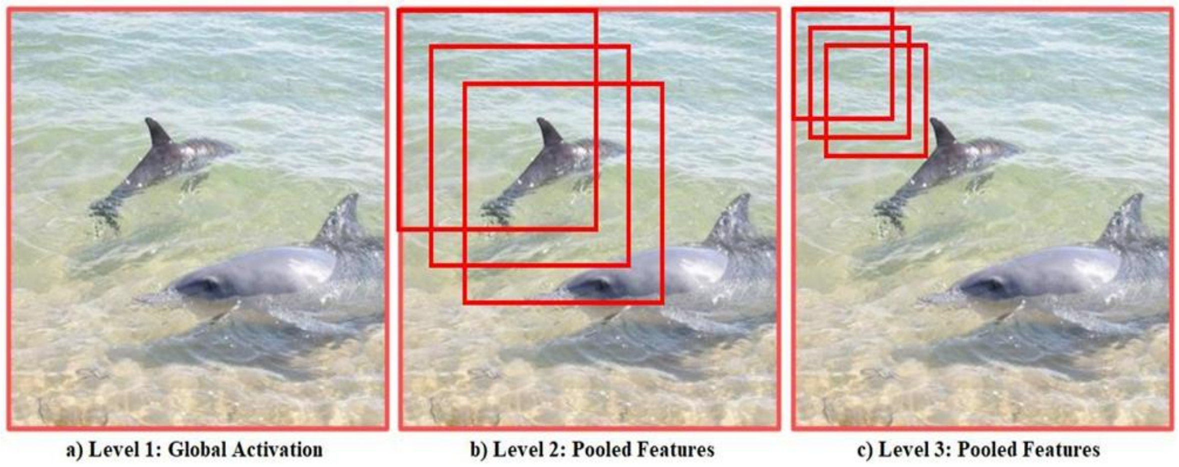
a) Level 1: Global Activation
b) Level 2: Pooled Features
c) Level 3: Pooled Features

Fig.5. Overview of multi-scale order-less pooling for CNN activations (MOP-CNN). It is the result of merging feature vectors from three distinct tiers: level 1 corresponds to the 4096-D CNN activation for the complete 256×256 image; level 2 is generated by extracting activations from 128×128 patches and VLAD pooling with a codebook of 100 centres; and level 3 is formed in the same manner as level 2 but with 64 × 64 patches [26].

3.2. Super-pixel Pooling

By clustering picture pixels according to low-level picture characteristics, super-pixels are created, which are an over-segmentation of an image [28]. By lowering the amount of picture primitives needed for future processing, they offer a tessellation of image material that is perceptually significant. Computer vision algorithms like object detection [29, 30], semantic segmentation [31-34], saliency estimation [35-38], optical flow estimation [39, 40], depth estimation [41, 42], and object tracking [43] make extensive use of super-pixel as a mid-level image representation because of its computational and representational efficiency. Figure 6 illustrates the super-pixel segmentation model in action.

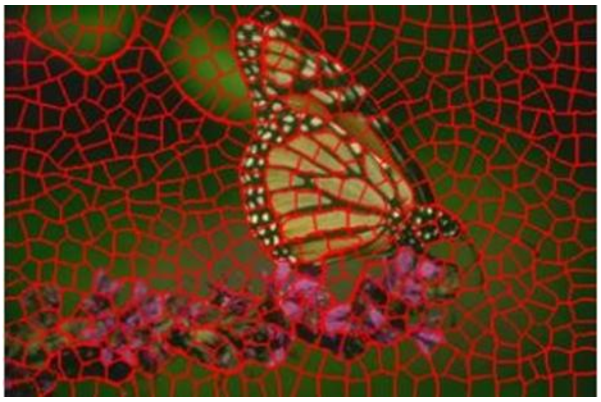

Fig. 6. An example of super-pixel segmentation [44]

There is an introduction to super-pixel pooling in [45, 46]. Using super-pixel segmentation as a pooling architecture, Super-pixel Pooling Network (SPN) learns and infers semantic segmentation in a weakly supervised environment by reflecting low-level picture structures.

3.3. PCA Networks

As a pooling stage, PCA is employed in [47]. Using principal component analysis (PCA), this approach teaches multistage filter banks. After that, we index and pool using block histograms and do basic binary hashing. Because of its simplicity and efficiency, this design has been given the name PCA network (PCANet). "As a pooling layer, PCA is employed in [48]. First suggested for use in audio processing was two-stage oriented principal component analysis (OPCA). One key distinction between OPCA and PCANet is that the latter does not use output layer coupling for hashing or local histograms. Opca becomes much more noise- and distortion-resistant when fed the noise covariances. It is possible to improve the baseline PCANet's resilience to interclass variability by adding OPCA's strengths.

3.4. Compact Bilinear Data Combination

When it comes to various visual tasks like semantic segmentation, fine-grained identification, and face recognition, bilinear models have proven to be quite effective. Bilinear features, on the other hand, are impracticable for further analysis due to their large dimensionality, which can range from hundreds of thousands to several million. In [49], the topic of bilinear models for picture categorization is covered. Presented in [50] is a compact bilinear pooling approach. A low-dimensional yet very discriminative picture representation is made possible by the compact bilinear pooling technique, which is learnt by means of end-to-end back propagation. A similar pooling strategy is employed in references [51–53].

The state-of-the-art results in several fine-grained datasets were accomplished by using bilinear pooling to produce rich and order-less global representation for the final convolutional feature. The calculation of pairwise interactions between channels, however, leads to the high dimensionality problem; hence, solutions for reducing dimensions are suggested. In particular, compact bilinear pooling [50] suggested a sampling-based approximation strategy that may decrease feature dimensions by two orders of magnitude without a performance decline, while low-rank bilinear pooling [54] suggested reducing feature dimensions before performing bilinear transformation. While second-order pooling convolutional networks [55] just employ bilinear features for channel weighting, they likewise advocate for the integration of bilinear interactions into convolutional blocks.

Figure 7 is a block diagram depicting compact bilinear pooling. This pooling technique allows for a low-dimensional but very discriminative picture representation; it is learnt via end-to-end back-propagation. The top pipeline displays the activation at one spatial region with the Tensor Sketch projection applied, where ∏ indicates circular convolution. The process of getting a global compact descriptor by sum pooling is illustrated in the bottom pipeline.

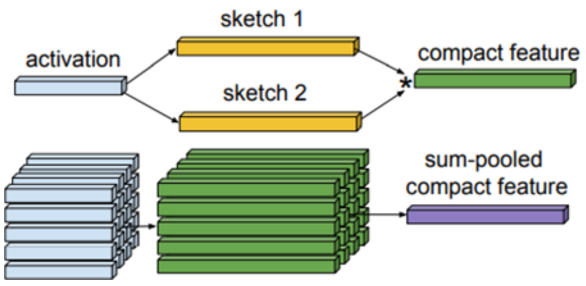

Fig. 7. Diagram of Compact bilinear pooling method for image classification [50].

3.5.Lead Asymmetric Pooling (LAP)

The pooling layers of traditional 2-dimensional convolutional neural networks (CNNs) provide a single output for each non-overlapping sub-region after down-sampling the input feature maps by a certain factor called the pooling factor. Importantly, due to the use of a single pooling factor, the conventional pooling method is unable to capture multi-scale patterns in multi-lead ECG. “Several image recognition experiments have shown that multilevel pooling strategies can improve CNN performance by making better use of multi-scale features [26] and increasing the invariance of local characteristics [19, 21]. Therefore, the LAP method is meant to take the role of regular pooling in order to handle the variety of multi-lead ECG [25]. By dividing the levels in accordance with the number of pooling factors applied, LAP can use multi-scale features as an extension of this multilevel pooling approach.

3.6.Edge-aware Pyramid Pooling

Another approach to pooling is edge-aware pyramid pooling. The pedestrian motion detection task incorporates the edge-aware feature map, which was suggested by Xu et.al. [56] as a means of preserving additional information about edge structures. The goal of edge detection is to locate the borders and edges of objects in photographs taken in the wild. Segmentation and target detection rely on edge detection, a fundamental computer vision job. To aid with pedestrian contour identification and motion prediction tasks, the authors of [56] utilised supplementary information in conjunction with edge-related data.

3.7.Spectral Pooling

A novel pooling approach was proposed by Rippel et al. [57] that incorporates the concept of dimensionality reduction through frequency domain input cropping. The desired dimensions of the output feature map are denoted by $h \times$ □, and □ is an input feature map that belongs to □ □×□. Following the application of the discrete Fourier transform (DFT) [58] on the input feature map, the centre of the frequency representation is chopped from the h × □ size submatrix. Lastly, the h × □ submatrix is transformed back into a spatial domain by applying inverse DFT. In comparison to max pooling, spectral pooling applies a linear lowpass filtering operation, which saves more information for the same output dimensionality. It gets around the issue of output map dimensionality being drastically reduced.

In most cases, lower frequencies account for a disproportionate share of the input spectrum power, whereas higher frequencies are more commonly used to encode noise [59]. The removal of high frequencies may be accomplished with minimum harm to input information due to the non-uniformity of spectrum power. Spectral pooling is based on the principle of matrix truncation, which uses quick Fourier transformation for convolutional kernels to decrease the computation cost in convolutional neural networks (CNNs) [60].

In Figure 8, we can observe a spectral pooling scheme and a Max pooling one. When you want to shorten the Fourier basis, spectral pooling is the way to go. This allows for the selection of any possible output map dimensions and maintains much more information, as seen in Figure 8.

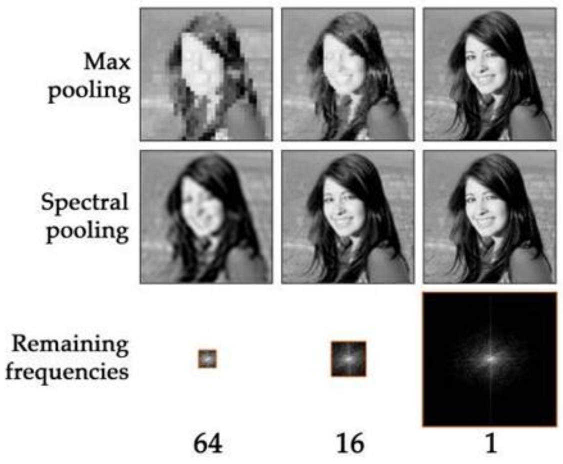

Fig. 8. Approximations for different pooling schemes, for different factors of dimensionality reduction [57].

3.8.Row-Wise Max-Pooling

The article introduces a novel method referred to as row-wise max pooling [61]. For each row of the input map, it finds the biggest value and adds it to the vector of outputs. Since the Row-wise max-pooling (RWMP) layer is insensitive to changes in the input map, its output is unaffected by the 3D form's orientation. The authors introduce DeepPano, a deep representation for rotation-invariant 3D form retrieval and classification, in [61]. Representations of three-dimensional shapes, both learned and recovered, are the fundamental elements of panoramic perspectives. When it comes to retrieval and classification, DeepPano is far ahead of the pack. Additionally, experimental confirmation of the representation's rotation invariance has been obtained.

3.9.Intermap Pooling

Grouping the filters and then pooling the feature maps inside each group is what the Intermap Pooling (IMP) layer does [62]. One function of an IMP layer is to classify feature maps. Afterwards, the greatest activation value at each place is propagated by each group. In formal terms, Eq. 6 gives the output of the □th group, which consists of □ consecutive feature maps:

$(i,j,k)\ (l) = max\gamma{=}{-}r{+}1,\ldots,0\tilde{H}\ (i,j,kr{+}\gamma)\ (l)$ (6)

Let $(\boldsymbol{l})$ stand for input to the lth convolution layer having K filters

Here, the feature maps of the several groups are combined after the filters in each group extract shared but spectrally different characteristics. Hence, the suggested IMP CNN is able to attain spectral variation insensitivity, which is indicative of various speakers and utterances. We show that the IMP CNN architecture is effective on many LVCSR tasks. By itself, the architecture's 12.7% WER on the SWB subset of the Hub5'2000 assessment test set is comparable with other top-tier approaches; this is without even considering speaker adaption strategies.

3.10. Per-pixel Pyramid Pooling

One alternative to employing a small pooling window with a stride to obtain the appropriate receptive field size is to utilise a big pooling window. Loss of finer features can occur while using a single huge pooling window. As a result, new feature maps are generated by merging the results of many pooling operations with different window widths. Information from coarse to fine scales is contained in the feature maps that are produced. Without any breaks, the multiscale pooling procedure is executed on each pixel. In formal terms, per-pixel pyramid pooling is defined as follows:

$P\ 4(F, s) = [P(F, s1\ ), \ldots , P(F, sM)]$ (7)

The pooling operation with size □□ and stride one is represented as (□, □□), and □ is a vector with □ number of entries. Figure 9 only displays one channel of the feature maps because to space constraints, but the pooling process is the same for all of them.

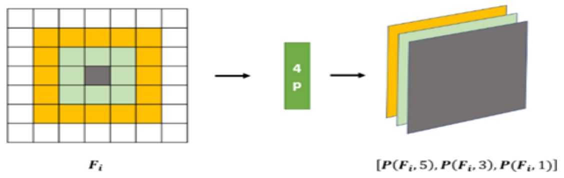

Fig. 9. the 4P module with pooling size vector s = [5, 3, 1] is visualized

3.11. Rank-based Average Pooling

Because average pooling takes into account the average operation for near-zero negative activations, it may lose discriminative information and downplay larger activation levels. The loss of information occurs when max-pooling completely discards non-maximum activations. To get around these issues with max and average pooling losing essential information, rank-based average pooling (RAP) [64] can be employed. Eq. 8 represents the RAP's output:

$$Sj = 1\ t \sum aii \in Rj,i9 \qquad (8)$$

where □ is the rank threshold that decides which activations are used for averaging. A feature map's pooling region is denoted by □, while the index of each activation inside it is denoted by □. In this context, □□ and □□ denote the rank and value of activation, respectively. Here, maxpooling occurs when □ = 1. In order for RAP to achieve a satisfactory compromise between average pooling and max pooling, it is necessary to correctly select □. You can exclude low-value or negative activations and retain high-response activations by using the median value of □. Figure 10 shows a toy example of rank-based pooling in action.

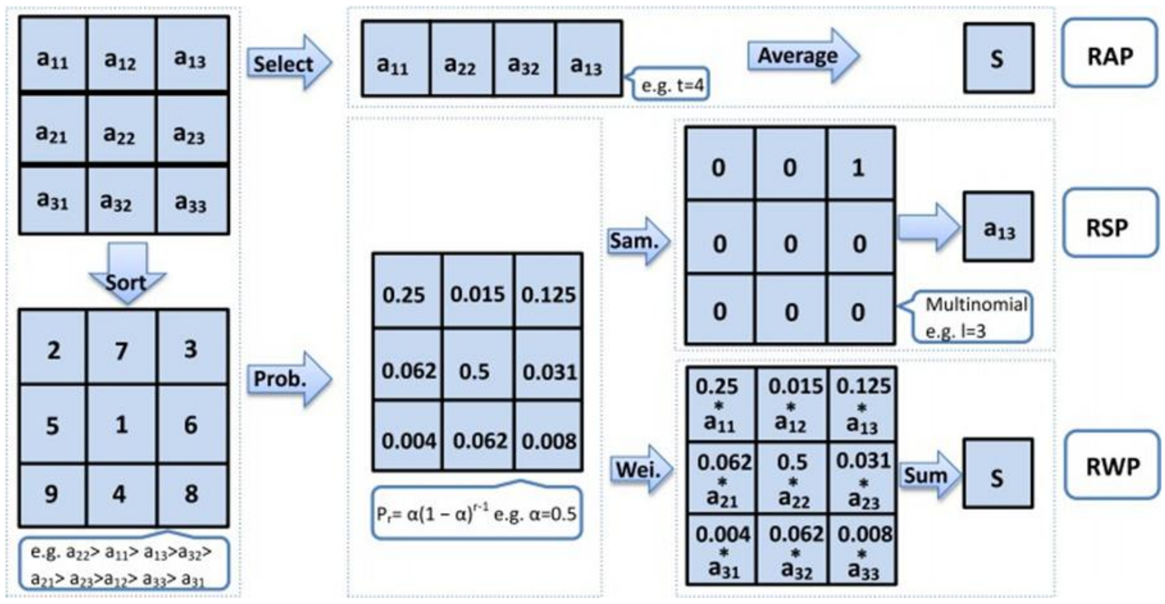
a_{11} a_{12} a_{13} / a_{21} a_{22} a_{23} / a_{31} a_{32} a_{33}
Sort
2 7 3 / 5 1 6 / 9 4 8
e.g. $a_{22} > a_{11} > a_{13} > a_{32} > a_{21} > a_{23} > a_{12} > a_{33} > a_{31}$
Select
a_{11} a_{22} a_{32} a_{13}
e.g. t=4
Average
S
RAP
Prob.
0.25 0.015 0.125 / 0.062 0.5 0.031 / 0.004 0.062 0.008
$P_r = \alpha(1-\alpha)^{r-1}$ e.g. α=0.5
Sam.
0 0 1 / 0 0 0 / 0 0 0
a_{13}
Multinomial e.g. l=3
RSP
Wei.
0.25 * a_{11} 0.015 * a_{12} 0.125 * a_{13} / 0.062 * a_{21} 0.5 * a_{22} 0.031 * a_{23} / 0.004 * a_{31} 0.062 * a_{32} 0.008 * a_{33}
Sum
S
RWP

Fig. 10. A toy example illustrates rank-based pooling. Activations within each pooling region are first sorted according to their activation values to obtain the rank,

and then their rank is reasonably used by RAP, RWP, and RSP. Here Prob., Sam. and Wei. are used as the shorthand of probability, sample and weighting, respectively [64].

3.12. Weighted Pooling

To account for both the response of each neuron and the utility of that response, Dong et.al. [65] proposed weighted pooling. What this means is that the pooling area assigns a weight to each neuron whose response is considered valuable. For each neuron, the response value is □□, where □ = 1, …, □ and □ = 1, …, h, and the pooling window has dimensions □□ × □h. Next, using Eq. 9, we can get the pooling results of the □□ × □h window.

$$Presult = wi, *ai,j \qquad (9)$$

where □□,□ represents the mass of □□,□. The suggested weighted pooling will improve the local representation by capturing various proportions of each neuron's local information in the original feature map.

3.13. Genetic-Based Pooling

The attention weights in the prior pooling approaches were generated using extremely thick layers. Thus, the size of the model that has to be trained increases. "Bhattacharjee et.al. [66] reduce the size and difficulty of training the model by using the Genetic Algorithm (GA) for pooling. In 1992, John Holland put forth the GA proposal [67]. It is a method for solving difficult optimisation issues that is based on the same principles as biological evolution. In GA, mutation, selection, and crossover are the three primary processes. By selecting parents from within a generation, we can increase our chances of producing offspring with desirable traits through processes like genetic crossover and mutation. Over the course of several generations, the population reaches a state of perfection.

As an initial step, this technique randomly generates a population of attention weights from the interval [0,1]. For every population set of attention weights, the model is trained and the appropriate loss functions are used to determine the error. Then, these attention weights are fine-tuned via generations until they cause the least amount of loss. In Algorithm 1, you may find the genetic pooling algorithm".

Algorithm 1: Genetic Pooling

For every individual bag

• Initialise the population with P bags. • Assign instance weights □□ to each bag. • Set the iteration number □ to 1. • Set the bag number in the population □ to 1. • While □ is less than or equal to the maximum iteration number: - While □ is less than or equal to □: - Calculate □ □ as the sum of □□ □ *h* □ □ □. - Perform a feed-forward pass on the neural network. - Calculate the loss. - Calculate the fitness value □□□□.

• Terminate the loop. • Select the most suitable half of the population based on their fitness value. • Execute the crossover operation between the selected individuals, considering the type of crossover and the probability of crossover. • Perform the mutation operation. • Replace the least fit half of the population with the newly generated offspring. • Terminate the loop. • End the iteration. • Return the value of □.The user's text is not clear or understandable.

4. Conclusion

Now, there are a great deal of DNN structures available. In terms of architectural, these buildings are distinct from one another, although they share fundamental components. One of the fundamental components of a convolutional-based deep neural network (DNN) is the pooling layer. A great number of different approaches were suggested by researchers for the implementation of this layer. Within the scope of this article, we investigated a number of well-known and practical pooling approaches from 1989 to 2020. After dividing those approaches into two categories—popular methods and new ways—we provided a brief description of each method.

5. References

1. Goyal, A., Kanyal, H. S., Kaushik, S., & Khan, R. (2021). IoT based cloud network for smart health care using optimization algorithm. Informatics in Medicine Unlocked, 27, 100792. https://doi.org/10.1016/j.imu.2021.100792

2. Kaushik, S., & Singh, R. (2016). A new hybrid approch for palm print recognition in PCA based palm print recognition system. 2016 5th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), 239–244. https://doi.org/10.1109/ICRITO.2016.7784958

3. Kaushik, S., Poonia, R. C., & Khatri, S. K. (2022). Cryptographic key distribution using artificial intelligence for data security and location privacy in VANET. Journal of Discrete Mathematical Sciences and Cryptography, 25(7), 2195–2203. https://doi.org/10.1080/09720529.2022.2133256

4. Kaushik, S., Poonia, R. C., Khatri, S. K., Samanta, D., & Chakraborty, P. (2022). Transmit range adjustment using artificial intelligence for enhancement of location privacy and data security in service location protocol of vanet. Wireless Communications and Mobile Computing, 2022, 1–13. https://doi.org/10.1155/2022/9642774

5. H., M., & A., H. (2014). A secured framework for geographical information applications on web. International Journal of Advanced Computer Science and Applications, 5(2). https://doi.org/10.14569/IJACSA.2014.050203
6. Ansari, A. A., Mishra, B., Gera, P., Khan, M. K., Chakraborty, C., & Mishra, D. (2022). Privacy-enabling framework for cloud-assisted digital healthcare industry. IEEE Transactions on Industrial Informatics, 18(11), 8316–8325. https://doi.org/10.1109/TII.2022.3170148
7. Bordes,A.,etal.*Jointlearning ofwordsand meaning representationsforopen-textsemanticparsing*.in*Artificial Intelligence and Statistics*. 2012.
8. Cireşan,D.C.,U.Meier,andJ.Schmidhuber. *TransferlearningforLatinandChinesecharacterswithdeepneural networks*. in *The 2012 International Joint Conference on Neural Networks (IJCNN)*. 2012. IEEE.
9. Ren,J.S.andL.Xu.*Onvectorizationofdeepconvolutionalneuralnetworksforvisiontasks*.in*Twenty-NinthAAAI Conference on Artificial Intelligence*. 2015.
10. Mikolov, T., et al.*Distributed representations of words and phrases and their compositionality*. in *Advances inneural information processing systems*. 2013.
11. Krizhevsky,A.,I.Sutskever,andG.E.Hinton.*Imagenetclassificationwithdeepconvolutionalneuralnetworks*. in *Advancesinneuralinformationprocessingsystems*.2012.
12. Ciregan, D., U. Meier, and J. Schmidhuber. *Multi-column deep neural networks for image classification*. in *2012 IEEE conference on computer vision and pattern recognition*. 2012. IEEE.
13. Deng,L., *Atutorialsurveyof architectures,algorithms,and applicationsfordeep learning.* APSIPATransactions on Signal and Information Processing, 2014. **3**.
14. Guo,Y.,etal.,*Deeplearningforvisualunderstanding:Areview*.Neurocomputing,2016.**187**:p.27-48.
15. Pascanu,R.,etal.,*Howtoconstructdeeprecurrentneuralnetworks.*arXivpreprintarXiv:1312.6026,2013.
16. LeCun,Y.,etal.*Handwrittendigitrecognitionwithaback-propagationnetwork*.in*Advancesinneuralinformation processing systems*. 1990.
17. LeCun,Y.,etal.,*Gradient-basedlearningappliedtodocumentrecognition.*ProceedingsoftheIEEE,1998.**86**(11): p.2278-2324.
18. Ranzato,M.A.,Y.-L. Boureau,andY.L.Cun.*Sparsefeaturelearningfordeepbelief networks*.in*Advancesin neural information processing systems*. 2008.
19. Yu,D.,etal.*Mixedpoolingforconvolutionalneuralnetworks*.in*Internationalconferenceonroughsetsand knowledge technology*. 2014. Springer.
20. Hinton, G.E., et al., *Improving neural networks by preventing co-adaptation of feature detectors.* arXiv preprintarXiv:1207.0580, 2012.
21. Wan,L.,etal.*Regularizationofneuralnetworksusingdropconnect*.in*Internationalconferenceonmachine learning*. 2013.
22. Sermanet,P.,S.Chintala,andY.LeCun.*Convolutionalneuralnetworksappliedtohousenumbersdigitclassification*.in*Proceedingsofthe21stInternationalConferenceonPatternRecognition(ICPR2012)*.2012.IEEE.
23. Zeiler, M.D. and R. Fergus, *Stochastic pooling for regularization of deep convolutional neural networks*.arXivpreprint arXiv:1301.3557, 2013.
24. Grauman,K.andT.Darrell.*Thepyramidmatchkernel:Discriminativeclassificationwithsetsofimagefeatures*.in *TenthIEEEInternationalConferenceonComputerVision(ICCV'05)Volume1*.2005.IEEE.
25. Lazebnik,S., C.Schmid,andJ.Ponce. *Beyondbagsoffeatures: Spatialpyramid matchingforrecognizingnatural scene categories*. in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*. 2006. IEEE.
26. Sivic, J. andA. Zisserman. *Video Google: A text retrieval approach to object matching in videos*. in *null*. 2003.IEEE.
27. He, K.,etal., *Spatialpyramidpooling in deep convolutionalnetworksforvisualrecognition.* IEEEtransactionson pattern analysis and machine intelligence, 2015. **37**(9): p. 1904-1916.
28. Huang,Z.,etal.,*DC-SPP-YOLO:denseconnectionandspatialpyramidpoolingbasedYOLOforobjectdetection.* InformationSciences,2020.
29. Zhou,J.,etal.,*Improved UAVOpiumPoppyDetection UsinganUpdatedYOLOv3Model.* Sensors,2019.**19**(22): p.4851.
30. Girshick,R.*Fastr-cnn*.in*ProceedingsoftheIEEEinternationalconferenceoncomputervision*.2015.
31. Liu,W.,etal.,*Real-timemultileadconvolutionalneuralnetworkformyocardialinfarctiondetection.*IEEEjournal of biomedical and health informatics, 2017. **22**(5): p. 1434-1444.
32. Gong,Y.,etal.*Multi-scaleorderlesspoolingofdeepconvolutionalactivationfeatures*.in*Europeanconferenceon computer vision*. 2014. Springer.
33. Jegou,H.,etal.,*Aggregatinglocalimagedescriptorsintocompactcodes.*IEEEtransactionsonpatternanalysis

and machine intelligence, 2011. **34**(9): p. 1704-1716.

34. Ren,X.andJ.Malik.*Learningaclassificationmodelforsegmentation*.innull.2003.IEEE.
35. Shu,G.,A.Dehghan,andM.Shah.*Improvinganobjectdetectorandextractingregionsusingsuperpixels*.in *ProceedingsoftheIEEEConferenceonComputerVisionandPatternRecognition*.2013.
36. Yan,J.,etal.*Objectdetectionbylabelingsuperpixels*.in*ProceedingsoftheIEEEConferenceonComputerVisi on and Pattern Recognition*. 2015.
37. Gadde,R.,etal.*Superpixelconvolutionalnetworksusingbilateralinceptions*.in*EuropeanConferenceonCo mputer Vision*. 2016. Springer.
38. Gould, S., et al., *Multi-class segmentation with relative location prior*. International Journal of Computer Vision, 2008. **80**(3): p. 300-316.
39. Sharma,A.,O.Tuzel,andM.-Y.Liu.*Recursivecontextpropagationnetworkforsemanticscenelabeling*.in *AdvancesinNeuralInformationProcessingSystems*.2014.
40. Wang,X.,H.Ma,andS.You,*Deepclusteringforweakly-supervisedsemanticsegmentationinautonomousdriving scenes*. Neurocomputing, 2020. **381**: p. 20-28.
41. He, S., et al., *Supercnn: A superpixelwise convolutional neural network for salient object detection*. International journal of computer vision, 2015. **115**(3): p. 330-344.
42. Perazzi,F.,etal. *Saliencyfilters:Contrastbasedfilteringforsalientregiondetection*.in*2012IEEEconferenceon computer vision and pattern recognition*. 2012. IEEE.
43. Yang, C., et al. *Saliency detection via graph-based manifold ranking*. in *Proceedings of the IEEE conference oncomputer vision and pattern recognition*. 2013.
44. Zhu, W., et al. *Saliencyoptimization from robust background detection*. in *Proceedings ofthe IEEE conferenceon computer vision and pattern recognition*. 2014.
45. Hu, Y.,etal.,*Highlyaccurateoptical flowestimationonsuperpixeltree*.Imageand VisionComputing,2016.**52**: p.167-177.
46. Yamaguchi, K., D. McAllester, and R. Urtasun. *Robust monocular epipolar flow estimation*. in *Proceedingsof the IEEE conference on computer vision and pattern recognition*. 2013.
47. Van den Bergh, M., D. Carton, and L. Van Gool. *Depth SEEDS: Recovering incomplete depth data using superpixels*. in *2013 IEEE Workshop on Applications of Computer Vision (WACV)*. 2013. IEEE.
48. Song, D., et al., *Integration of super-pixel segmentation and deep-learning methods for evaluating earthquake- damaged buildings using single-phase remote sensing imagery*. International Journal of Remote Sensing, 2020. **41**(3): p. 1040-1066.
49. Yang,F.,H.Lu,andM.-H.Yang,*Robustsuperpixeltracking*.IEEETransactionsonImageProcessing,2014.**23**(4): p.1639-1651.
50. Wang, M., et al., *Superpixel segmentation: A benchmark*. Signal Processing: Image Communication, 2017. **56**: p. 28-39.
51. Liu,F.,etal.,*Learningdepthfromsinglemonocularimagesusingdeepconvolutionalneuralfields*.IEEE transactions on pattern analysis and machine intelligence, 2015. **38**(10): p. 2024-2039.
52. Kwak,S.,S.Hong,andB.Han.*Weaklysupervisedsemanticsegmentationusingsuperpixelpoolingnetwork*.in *Thirty-FirstAAAIConferenceonArtificialIntelligence*.2017.
53. Chan,T.-H.,etal., *PCANet:A simpledeeplearningbaselineforimageclassification?* IEEEtransactionsonimage processing, 2015. **24**(12): p. 5017-5032.
54. Burges, C.J., J.C. Platt, and S. Jana, *Distortion discriminant analysis for audio fingerprinting*. IEEE Transactions on Speech and Audio Processing, 2003. **11**(3): p. 165-174.
55. Lin,T.-Y.,A.RoyChowdhury,andS.Maji.*Bilinearcnnmodelsforfine-grainedvisualrecognition*.in*Proceedings of the IEEE international conference on computer vision*. 2015.
56. Gao, Y., et al. *Compact bilinear pooling*. in *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.
57. Chetouani, A., et al., *Classification of engraved pottery sherds mixing deep-learning features by compact bilinear pooling*. Pattern Recognition Letters, 2020. **131**: p. 1-7.
58. Wang, Y., et al., *Attention boosted bilinear pooling for remote sensing image retrieval*. International Journal of Remote Sensing, 2020. **41**(7): p. 2704-2724.
59. Min, S., et al., *Multi-Objective Matrix Normalization for Fine-Grained Visual Recognition*. IEEE Transactions on Image Processing, 2020. **29**: p. 4996-5009.
60. Kong, S. and C. Fowlkes. *Low-rank bilinear pooling for fine-grained classification*. in *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017.
61. Gao, Z., et al. *Global second-order pooling convolutional networks*. in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019.
62. Xu, L., et al., *Motion Recognition Algorithm Based on Deep Edge-Aware Pyramid Pooling Network in*

Human– Computer Interaction. IEEE Access, 2019. **7**: p. 163806-163813.

63. Rippel, O., J. Snoek, and R.P. Adams. *Spectral representations for convolutional neuralnetworks*. in *Advances in neural information processing systems*. 2015.
64. Duhamel,P.,Y.Mahieux,andJ.Petit.*Afastalgorithmfortheimplementationoffilterbanksbasedon'timedoma in aliasing cancellation'*. in *[Proceedings] ICASSP 91: 1991 International Conference on Acoustics, Speech, and Signal Processing*. 1991. IEEE.
65. Torralba,A.andA.Oliva,*Statisticsofnaturalimagecategories*.Network:computationinneuralsystems,2003. **14**(3):p.391-412.
66. Mathieu,M.,M.Henaff,andY.LeCun,*Fasttrainingofconvolutionalnetworksthroughffts*.arXivpreprint arXiv:1312.5851, 2013.
67. Shi,B.,etal.,*Deeppano:Deeppanoramicrepresentationfor3-dshaperecognition*.IEEESignalProcessingLetters, 2015. **22**(12): p. 2339-2343.
68. Lee, H., et al., *Deep CNNs along the time axis with intermap pooling for robustness to spectral variations*. IEEE Signal Processing Letters, 2016. **23**(10): p. 1310-1314.
69. Park,H.andK.M.Lee,*Lookwidertomatchimagepatcheswithconvolutionalneuralnetworks*.IEEESignal Processing Letters, 2016. **24**(12): p. 1788-1792.
70. Shi,Z.,Y.Ye,andY.Wu,*Rank-basedpoolingfordeepconvolutionalneuralnetworks*.NeuralNetworks,2016, **83**:p.21-31.
71. Dong,L.,etal.,*CUNet:ACompactunsupervisednetworkforimageclassification*.IEEETransactionson Multimedia, 2017. **20**(8): p. 2012-2021.
72. Bhattacharjee,K.,etal.,*MultipleInstanceLearningwithGeneticPoolingforMedicalDataAnalysis*.Pattern Recognition Letters, 2020.
73. John,H.,*Holland.geneticalgorithms*.Scientificamerican,1992.**267**(1):p.44-50.