



# Interpretation of intelligence in CNN-pooling processes: a methodological survey

Nadeem Akhtar<sup>1</sup> · U. Ragavendran<sup>1</sup>

Received: 17 December 2018 / Accepted: 17 June 2019  
© Springer-Verlag London Ltd., part of Springer Nature 2019

## Abstract

The convolutional neural network architecture has different components like convolution and pooling. The pooling is crucial component placed after the convolution layer. It plays a vital role in visual recognition, detection and segmentation course to overcome the concerns like overfitting, computation time and recognition accuracy. The elementary pooling process involves down sampling of feature map by piercing into subregions. This piercing and down sampling is defined by the pooling hyperparameters, viz. stride and filter size. This down sampling process discards the irrelevant information and picks the defined global feature. The generally used global feature selection methods are average and max pooling. These methods decline, when the main element has higher or lesser intensity than the nonsignificant element. It also suffers with locus and order of nominated global feature, hence not suitable for every situation. The pooling variants are proposed by numerous researchers to overcome concern. This article presents the state of the art on selection of global feature for pooling process mainly based on four categories such as value, probability, rank and transformed domain. The value and probability-based methods use the criteria such as the way of down sampling, size of kernel, input output feature map, location of pooling, number stages and random selection based on probability value. The rank-based methods assign the rank and weight to activation; the feature is selected based on the defined criteria. The transformed domain pooling methods transform the image to other domains such as wavelet, frequency for pooling the feature.

**Keywords** Pooling intelligence · Hybrid machine learning tools · Supervised algorithms · Stable sampling · Deep learning methodology · Self-learning artificial neural network

## 1 Introduction

Convolutional neural network (CNN) architecture has different components like convolution and pooling. The pooling is crucial component placed after the convolution layer. It is also called as subsampling or down sampling layer which discard around 75% information, without affecting the information. It plays a vital role in visual recognition [1, 2], detection and segmentation course to overcome the concerns like overfitting, computation time and recognition accuracy. The few architectures of CNN

[3] do not use the pooling. The performance of deep learning architectures degrades substantially without pooling. The absence of pooling causes propagation of local feature to neighboring receptive fields which ultimately weakens the representation power of CNN, and network becomes very sensitive to input deformations. The activation in pooling regions does not have any weight and biases as like in convolution layer which do not affect the depth of feature map. The pooling operation shrinks feature map resolution and preserves the critical discriminative information required for recognition task.

It reduces the number of neuronal connections, size of feature map [4]. It does not need the zero padding and performs the defined operations on the input feature maps. Hence, it reduces the parameters, increases computational efficiency and regulates overfitting.

Ideally pooling operation preserves discriminative information while discarding irrelevant image details. It

---

✉ Nadeem Akhtar  
nadeem.riyaz@rcpit.ac.in

U. Ragavendran  
U.ragavendran@nmims.edu

<sup>1</sup> SVKM's NMIMS (Deemed to be University), Shirpur, India

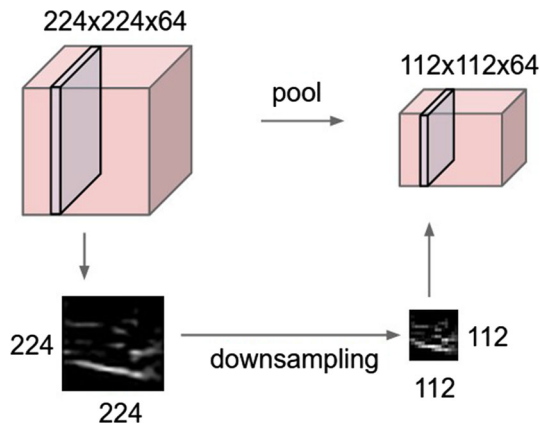


Fig. 1 Down sampling using pooling [5]

reduces the spatial dimension (see Fig. 1) which leads to loss of information. However, such a loss is beneficial for the network because the decrease in dimension results in less computational overhead for the forthcoming layers, and makes the algorithm robust to the translational invariance's.

Two hyperparameters, filter size  $F$  and stride  $S$ , are associated with pooling layer. The number of pixels skewed between each window is termed as stride. The equal values of spatial extent and stride define the overlapped [6, 7] and non-overlapped [8, 9] pooling methods. The overlapped pooling reduces two types of error rates by 0.4% and 0.3% [7]. The shape of filter varies between the square and rectangle [10]. Larger pooling window loses the important features of image [4]. An image of size  $224 \times 224$  pixels will output as a  $112 \times 112$  pixels image with pooling size of  $2 \times 2$  with a stride of 2 as shown in Fig. 1. The input image dimensions applied to the pooling layer are represented as  $W \times H \times D$  in which  $W$  is width,  $H$  is height, and  $D$  is depth of image. The output image dimensions after the application of pooling layer are

$$\text{New width} = \frac{(\text{Input width} - \text{Filter width})}{\text{Stride}} + 1 \quad (1)$$

$$\text{New width} = \frac{(\text{Input width} - \text{Filter width})}{\text{Stride}} + 1 \quad (2)$$

The most popular choices of pooling layers are average, max, sum and median pooling [11]. The max [12] pooling captures only the maximum activation [10] and skips the remaining in the pooling region, while average pooling calculates mean of all activation. The sum pooling adds all elements in the feature map, while median pooling [11] selects the median value from the pooling region of feature map. Median pooling [11] may capture the false value in the noisy environment. The max and average are the basic two variants of pooling operation. The max pooling can be represented mathematically as

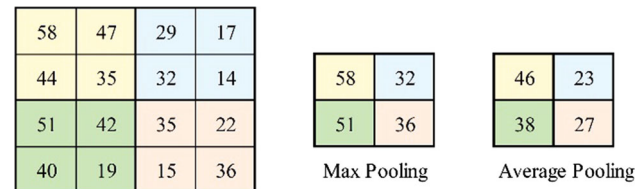


Fig. 2 Output feature after applying max and average pooling

$$a_{kij} = \max_{(p,q) \in R_{ij}} x_{kpq} \quad (3)$$

Here,  $y_{kij}$  represents the output of  $k$ th feature map of element  $x_{kpq}$  within pooling region  $R_{ij}$ . The max pooling is fast and quickly shrinks the hidden layer dimensions. It introduces the degree of invariances but suffers with generalization ability due to the disjoint nature [13]. The average pooling calculates the average of all activation. The average pooling in mathematical form is

$$y_{kij} = \frac{1}{|R_{ij}|} \sum_{(p,q) \in R_{ij}} x_{kpq} \quad (4)$$

Here,  $y_{kij}$  represents the output of  $k$ th feature map of element  $x_{kpq}$  within pooling region  $R_{ij}$ . Figure 2 represents the processed output after applying max and average pooling. The max pooling is able to extract the features like edge and textures, whereas average pooling may not extract these features, as it takes an average value that may not be important for object detection. A max pooling is used by [14] in 2007 when backward propagation is applied in CNN. The max pooling according to [15] is superior in capturing the invariance in image with good generalization and faster convergence capabilities. It is tested on the normalized uniform NORB [16] dataset with half percent improvement in results. A detailed analysis of max and average pooling is carried out by [17]. They found that the performance of either max or average pooling depends on the data and its feature and either pooling strategy may not be optimal for classification problem. A probabilistic max pooling is proposed by [18] for convolutional deep belief networks for full-sized and high-dimensional images. It is tested on several classification benchmarks such as MNIST [19] and CALTECH-101 [20].

### 1.1 Bottlenecks with max and average pooling

Generally, CNN employs two types of pooling, namely average and max pooling due to their computational efficiency. Although the use of max pooling has resulted in excellent empirical results [7, 21], it can overfit the training data and does not guarantee generalization on test data. Average pooling, on the other hand, considers all the elements in the pooling region, and thus, areas of low activation may lessen the effect of areas of high activation

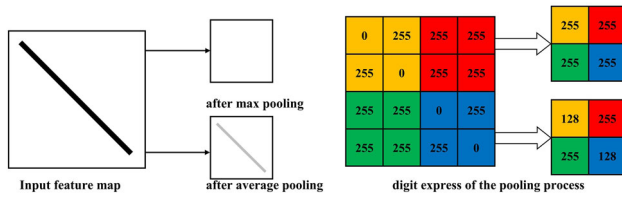


Fig. 3 Failure state of max pooling [9]

[22, 23]. According to [24], average pooling produces better classification results on the CALTECH-101 [20] datasets. These two types of pooling work well on some known datasets, but have few drawbacks. The max pooling picks the highest value in pooling region, but fails in a situation, when the intensity of main element is lesser than the insignificant elements and if most of the elements have the High magnitude (see Fig. 3). Another situation arises when the image has many noises with larger values in the pooling region (see Fig. 5).

In this situation stated 101 is noise selected by the max pooling operation. The other situation arises when there are many larger values in pooling region having lesser difference (such as 11 and 13). These difference values are relatively less, but it may bring large effects through millions of net parameters. This leads to overfitting of training data and results in an unacceptable result.

The average pooling takes all low and high magnitudes of the elements and calculates the mean value of all elements in the pooling region. It may fail in situation when multiple zeros are present in pooling region (see Fig. 4).

## 2 Variants of pooling processes in CNN

The max and average pooling fails in the certain state (Figs. 3, 4, 5); hence, researcher proposes the different variants of pooling for accuracy improvement. These variants are defined based on the choice of output activation chosen, value and nature of hyperparameters, number of stages required for pooling, down sampling approach, relations between the neighboring activation, level of feature, order of feature, position of activation and sharing of same filter to all feature maps. This article reviews (see Table 1) the pooling process based on the four major categories based on value, rank, probability and transformed domain pooling methods. The value-based criteria again categorized as pixel and patch-based pooling. The patch-based pooling, such as subclass [25] and series multi-pooling [26], considers the patch area to be pooled. Alike values of stride and pooling kernel define the overlapped and non-overlapped pooling methods.

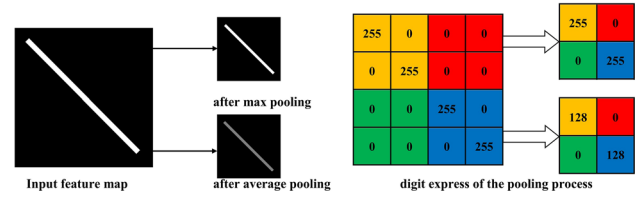


Fig. 4 Failure state of average pooling [9]

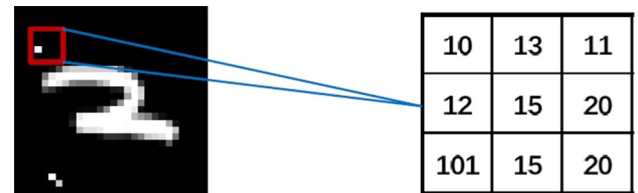


Fig. 5 Failure state of max pooling [62]

### 2.1 Value-based pooling methods

The value-based pooling down samples the pooling region and selects the single activation based on its value. It is further classified as significance, patch-based and multi-sampling methods. The patch-based method selects different patches, and pooling operations are applied on these patches. The down sampling operation in pooling discards the 75% information that leads to the loss of information. The multisampling better scales the spatial resolution of the output feature map while preserving the benefits of traditional subsampling layers such as increasing receptive field and reducing computational costs.

#### 2.1.1 Significance pooling

In significance pooling methods, kernel swifts and aggregates the information within pooling region and replaces it with single value. The other process involves the region of interest detection (patch) and applying pooling operation on patch. The pooling operation aggregates the information in pooling region and transforms it into a single value based on criteria of pooling method. The few significance pooling methods introduce the randomness in selection of average and max pooling methods, since these two methods have better performance in certain conditions

*Mixed pooling* [9] uses the combination of both max and average pooling (hence named mixed pooling) for boosting the regularization abilities of CNNs and addresses the bottlenecks of average and max pooling [22, 23]. The selection of max and average pooling is based on a parameters  $\lambda$ ; its value reflects either max or average pooling selection. The mixed pooling is expressed by Eq. 5.

**Table 1** Review chart of pooling variants

Pooling variants		
Value based	Significance	Max pooling [22] Average pooling [62] Mixed pooling [9] Global average pooling [28] LEAP pooling [31] Spatial pyramid pooling [8] Kernel pooling [40] Fractional max pooling [13] Dynamic correlation pooling [4] Multi-activation pooling [45] Combined pooling [46] Detail preserving pooling [30] Concentric circle pooling [47] <i>K</i> support pooling [44] Transformed invariant pooling [36] Subclass pooling [25] Series multi-pooling pooling [26] Partial mean pooling [50] Region of interest pooling [99] Multi-scale orderless pooling [49] Checker board pooling [54] Parallel grid pooling [55]
	Patch	
	Multisampling	
Probability-based pooling methods	Lp pooling [100] Stochastic spatial sampling pooling (S3Pool) [63] Hybrid pooling [66] Stochastic pooling [59] Dropout max pooling [10] Sparsity stochastic pooling [64] Failure density probability pooling [72] Mixed, gated and tree pooling [69]	
Rank-based pooling methods	Multipartite pooling [41] Ordinal pooling [70] Global weighted rank pooling [71] Rank-based average pooling [65] Rank-based weighted pooling [65] Rank-based stochastic pooling [65]	
Transformed domain pooling methods	Time domain-based pooling [96] Frequency domain-based pooling [80] Wavelet domain-based pooling [79]	

$$y_{kij} = \lambda \cdot \max_{(p,q) \in R_{ij}} x_{kpq} + (1 - \lambda) \cdot \frac{1}{|R_{ij}|} \sum_{(p,q) \in R_{ij}} x_{kpq} \quad (5)$$

Here,  $y_{kij}$  represents the output for  $k$ th feature map. The element at location  $(p, q)$ , within the pooling region  $R_{ij}$

with size  $R_{ij}$ , is represented by  $x_{kpq}$ . The value of  $\lambda$  is random either one or zero. During forward propagation process,  $\lambda$  is recorded and will be used for the backward propagation operation. The mixed pooling [9] found superior to max and average pooling in terms of accuracy, and in addressing the overfitting issues. The mixed pooling fails in reflecting the advantages of max and average

pooling at the same time, since it selects either method in pooling process. The other random pooling is alpha pooling proposed by [27]. It introduce the parameter  $\alpha$  which uses alpha integration to decide the selection between max and arithmetic average pooling.

Global average pooling [28] replaces the fully connected (FC) layer. The basic idea of global average (GA) pooling is to gross the average value of each feature map and fed this vector to the softmax layer. The absence of FC layer reduces the number of parameter and computation time, since FC layer requires around 90% of parameters as compared other layers of CNN. The feature map here corresponds to the class of classification task. Thus, the feature maps are inferred as class confidence maps. It does not require the optimization of parameters, hence avoids the overfitting issue and robust to spatial translations of input feature map. The variant of GA pooling is [29] which uses log mean exponential function (AlphaMEX) to extract the features.

*Detail preserving pooling* (DPP) [30] uses inverse bilateral filter for preserving the important details of feature maps. A learnable parameter controls the downscaling of feature map in order to preserve important structural details. Asymmetric and symmetric are two variants of DPP. The symmetric version enhances all the details, while asymmetric enhances the features higher than the average activation. The DPP incurs minor computational overhead and performs similar to max/extremum or average pooling, or on a nonlinear continuum of intermediate functions. DPP can be combined with stochastic pooling [22] methods with further accuracy gains as detail preservation and regularization complement each other.

*LEAP pooling* [31] uses the shared linear filter applied on each feature map, hence reduces the number of parameters and training error. The LEAP filter learns a shared linear filter in every feature map and aggregates the features within pooling region (see Fig. 6). The shared LEAP operator is rationalized with backward propagation algorithm during the end-to-end training stage. The computational complexity of LEAP pooling is much smaller than other pooling methods.

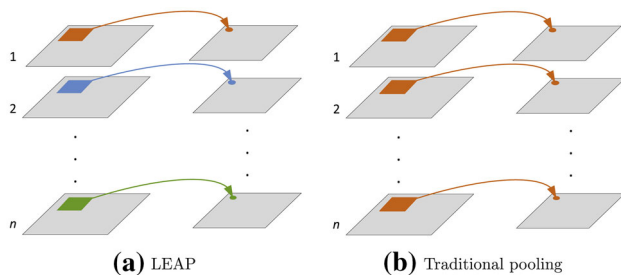


Fig. 6 Illustration of LEAP and traditional pooling [31]

*Spatial pyramid pooling* [8] conventionally, deep neural network requires fixed size input. In some applications, such as recognition and detection, input images are usually cropped and warped, and then fed into the deep neural network. Crop operator cannot obtain the whole object which means crop may lead to some information loss in some sense, and warp operator would introduce unwanted geometric distortion. These limitations will harm the recognition accuracy of neural network. A new pooling strategy called spatial pyramid pooling (SP pooling) is proposed by [8], which borrow the idea from the spatial pyramid matching model [32, 33]. The outstanding contribution of this structure is to generate fix length output regardless of the input size, while previous networks cannot. This layer is placed between the final convolutions/pooling layer and the first FC layer, hence performs information aggregation to avoid fixed size of input image (see Figs. 7, 8). The SP pooling divides the feature map into subimage and extracts the maximum value from pooling region.

It is similar to the bag of words which pools in local bin to maintain spatial information. It generates the fixed sized output irrespective of input size, which means the scale of image does not affect the final performance, and it would extract scale invariant feature. SP pooling customs multi-level spatial bins, while sliding window method customs only a single window size.

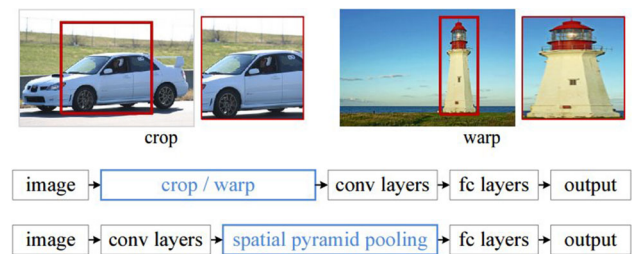


Fig. 7 CNN with and without spatial pyramid pooling [8]

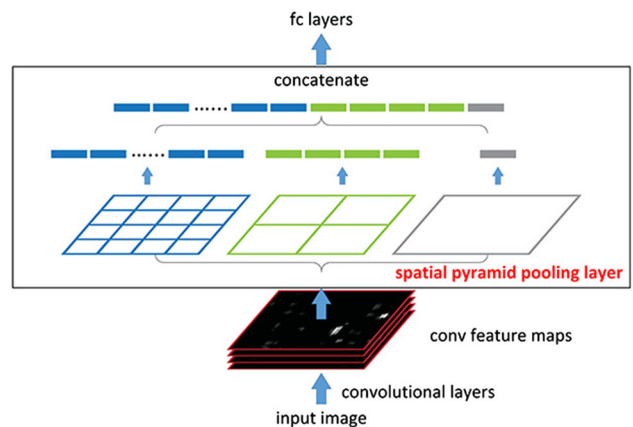


Fig. 8 Network structure of spatial pyramid pooling [8]



**Table 2** Window size for required bins [34]

Window	$H \times W$ size	Stride
Window1	$2 \times 2$	2
Window2	$4 \times 4$	2
Window3	$3 \times 6$ and $6 \times 3$	1

**Table 3** Pyramid structure [34]

Pyramid	Layers
Pyramid 1	Window1 + Window3
Pyramid 2	Window2 + Window3
Pyramid 3	MAX + Window3
Pyramid 4	MAX + Window1 + Window2
Pyramid 5	MAX + Window2 + Window3

The other variant of SP pooling is pyramid pooling [34] that integrates the spatial information into the feature vectors. The pyramid pooling reduces the dimensions without losing the important information. Initially, the feature map is divided into a  $6 \times 6$  dimensional different subregions (bins). Tables 2 and 3 summarize the window sizes required for forming bins and different pyramid structures used by [34]. They used the Alex net for experimentation which has 5 layers followed by FC layer.

The experimental results on INRIA Holidays and Oxford buildings dataset show the superiority in image retrieval. The other variant of SP pooling named atrous spatial pyramid pooling (ASP pooling) is proposed by [35] for segmentation of objects at manifold scales. ASP pooling reviews feature map at manifold sampling rates and effective fields of views, hence efficient in capturing the objects and context at manifold scales.

*Transformation invariant pooling* (TI) [36] is inspired by max pooling [17] and multiple instance learning [37]. It is applied on the top layer before FC layer. It generates new feature from a predefined set of possible transformations which is independent of nuisance variations such as rotation or scale of input. TI pooling passes multiple transformed versions of the input separately through the network to get feature representations for each transformed instance aggregated together by a max pooling operation to achieve transformation invariance. The aggregated feature representations are later passed to the rest of the network for the downstream task.

The FC layer introduces dense computation and increases the amount of parameter, making it slow and prone to overfit. Inception [38] and residual learning [39] use the GA pooling [28] to overcome this issue. It is

computationally efficient but unable to capture higher-order feature interactions, which plays a vital role in recognition task.

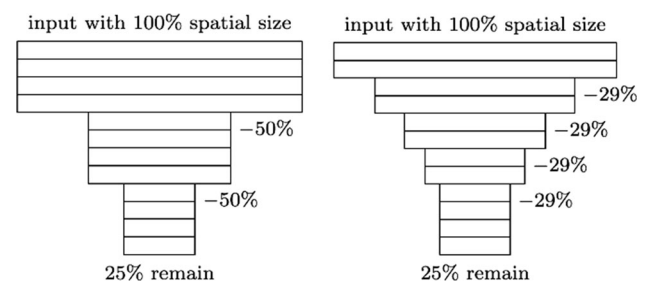
*Kernel pooling* [40] captures the higher-order features using Gaussian radial basis function, and GA pooling creates the final feature vector across all spatial locations. Two feature vectors  $x$  and  $y$  with  $\Phi$  as a kernel pooling, the inner-product between  $\Phi(m)$  and  $\Phi(n)$  approximate a kernel up to a certain order  $p$

$$\Phi(m)^T \Phi(n) \approx \sum_{i=0}^p \alpha_i^2 (m^T n)^i \approx \mathcal{K}(m, n) \quad (6)$$

The kernel composition identified by the coefficients is either predefined or learned from data. The kernel function with Hilbert space boosts the performance of classifier

*Fractional max pooling* The conventional non-overlapped pooling method down samples the feature map by discarding 75% information. This sudden reduction in spatial information may abandon some useful evidence required for consequent operations, especially when small input image is used. The fractional max pooling (FMP) [13] shrinks the spatial dimensions in more gradual way as shown in Fig. 9. FMP reduces the spatial size by introducing a factor  $\alpha$  ( $1 < \alpha < 2$ ). The value of  $\alpha$  is selected randomly or pseudo-randomly in the specified range for spatial dimension reduction. Like stochastic pooling [22], FMP [13] introduces a degree of randomness to the pooling method. The authors [41] call the FMP as spatial stochastic max pooling. However, unlike stochastic pooling [22], the randomness is mostly related with region of pooling, not the way of execution in pooling circle.

FMP is tested on the CIFAR [42] datasets and promising results are found; however, their observations lacked suitable motivation and the technique still needs to be tested on other architectures such as inception and residual networks. The FMP introduces randomness in terms of choice of pooling region that can be chosen in a random or pseudorandom manner. Pooling regions can be disjoint or overlapping. It is found that Random FMP is good on its own but may underfit when shared with dropout or training data augmentation. Pseudorandom approach generates

**Fig. 9** Comparison of down sampling process between conventional and fractional max pooling method [43]

more stable pooling regions. Overlapping FMP performs better than disjoint FMP. A variant of FMP as bi-linearly weighted fractional max pooling (BW-FMP) [43] reduces the spatial size more gradually. The BW-FMP is applied on ResNet (50 layered) and VGG Net (19 layered) with compact number of filters on four datasets such as FGVC-Aircraft, Oxford-IIIT Pet, STL-10 and CIFAR-100. Experimental results show that the use of BW-FMP improves the memory consumption and processing time by 18% and 13%, at the cost of classification accuracy. Accuracy is still higher for same configuration with low memory and faster computation time. While comparing with stochastic pooling [22], computation time and memory usage are same, but yields higher accuracy. The BW-FMP method offers flexibility in pooling size though trading off memory constraint (and computation time) for classification.

*Dynamic correlation pooling* [4] uses non-overlapping window with correlation information of adjacent activation is based on Mahalanobis distance. The pooling method operation is tabbed between average, max and mixed pooling. The Mahalanobis distance is measured and compared between two adjacent activations with a reference value  $\gamma$ . If the Mahalanobis distance is lesser than  $\gamma$ , average pooling is applied. The lesser value of Mahalanobis distance indicates that pooling regions are highly correlated with strong similarity. Hence, it is required to reserve the general characteristics and rise the error value, to prevent local constraints. The larger value of Mahalanobis distance indicates no correlation of data with adjacent pooling region; hence, max pooling is selected to preserve edge features. The Mahalanobis distance between the two adjacent (block 2 and block 3) is calculated and compared with the reference value  $\gamma$  (see Fig. 10). If the compared value between the  $d(x_1, x_2)$ ,  $d(x_1, x_3)$  is lesser than  $\gamma$ , then average pooling is opted while for less than or equal to  $\gamma$  indicates the selection of max pooling. When  $d(x_1, x_2) - \gamma$  and  $d(x_1, x_3) - \gamma$  have unlike signs, then the mixing operation will be performed by using Eq. 7 in which coefficients are calculated using Eq. 8.

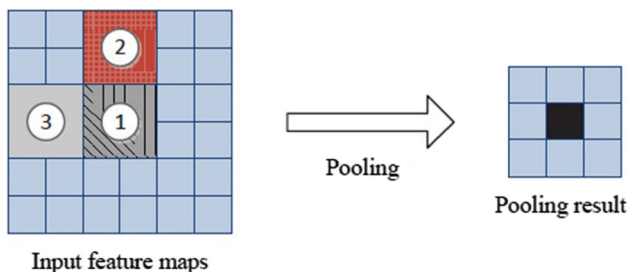


Fig. 10 Dynamic correlation pooling [4]

$$y_{i,j,k} = \lambda \cdot \max_{(m,n) \in R_{ij}} a_{m,n,k} + (1 - \lambda) \cdot \frac{1}{|R_{ij}|} \sum_{(m,n) \in R_{ij}} a_{m,n,k} \quad (7)$$

$$\lambda = \begin{cases} \frac{|d(x_1, x_3) - \gamma|}{|d(x_1, x_2) - \gamma| + |d(x_1, x_3) - \gamma|}, & d(x_1, x_3) > \gamma, d(x_1, x_2) < \gamma \\ \frac{|d(x_1, x_2) - \gamma|}{|d(x_1, x_2) - \gamma| + |d(x_1, x_3) - \gamma|}, & d(x_1, x_2) > \gamma, d(x_1, x_3) < \gamma \end{cases} \quad (8)$$

Dynamic correlation pooling is tested with Lenet-5 on MNIST [19], USPS and CIFAR-10 [42] dataset and compared with max, average, stochastic and mixed pooling. Experimental results prove the superiority of dynamic correlation pooling in terms of accuracy, rate of accuracy and lower error rate.

*K support spatial pooling* [44] was proposed for HEP-2 cell classification. The conventional CNNs mostly rely on the stable size data that hint the structure deformation. This issue is resolved by using SP pooling, but it does not consider the frequency of activation which is a vital mark for recognizing different forms of images. It mostly relies on aggregating activations in predefined spatial region, which retains only the strongest activation. This method sorts the activations in ascending order in pooling region and retains only the first  $k$  larger activations. The final degree of activation is the mean value of the retained  $k$  values. This procedure is repeated for each neuron to produce the activation degrees of all neurons in a distinct region.

*Multi-activation pooling* (MAP) [45] is applied for accurate classification without increasing depth and trainable parameters. MAP picks top- $K$  activation in every pooling region to assure that the maximum information can pass through subsampling gates. The pooling region is larger in size such as  $4 \times 4$ ,  $8 \times 8$ ,  $16 \times 16$ , and even larger can be used with max pooling. They used more number of convolution layers before certain pooling layer followed by ReLU. This arrangement reduces the noise even absence of few ignored information. The top- $K$  activations are picked, clubbed and constrained the summation by a constant  $\sigma$ , which ranges from zero to one. The  $\sigma$  value of  $1/k$  reflects average pooling of top- $K$  activation. Generally, the value of  $\sigma$  is little greater than  $1/k$  to avoid weakening of activation with limited active features. The MAP method achieves higher accuracy on classification task without increasing the depth and trainable parameters. In plain networks, such as VGG and ALL-CNN, this method of pooling is competitive in achieving higher accuracy on classification tasks with depth and trainable parameters not increased.

*Combination of max and average pooling* is used by [46] to achieve the better performance for traffic sign recognition. The pooling method selection in each

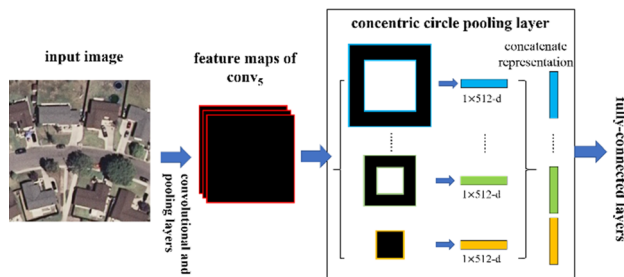


Fig. 11 Concentric circle pooling [47]

subsampling layer affects the recognition accuracy. Out of the different combination proposed by the author [46], average–average–max combination achieves faster and smaller convergence speed, value and lowermost error rate indicates a good classification ability.

**Concentric circle pooling** The SP pooling layer improves the image classification, but not effective on rotated image scenes. This sensitivity to the rotation of images degrades the classification performance in remote sensing images. The concentric circle pooling (CCP) [47] screens the feature map into a series of annular subregions and groups the confined features within the annular sub-region. The CCP layer is added before the FC layer (see Fig. 11). The response of pooling region in annular sub-region is pooled using average and max pooling. The output of last convolutional layer is divided into the number of the annular subregion. This division is square ring shaped, since square kernels are computationally effective at the partial expense of rotational invariance. The division ranges between 1 and  $R$ , ( $R$  represents square kernel number). The output dimensions of CCP layer are  $r \times K$  where  $r \in \{1, R\}$ , and  $K$  represents number of filters applied in last convolutional layer. The pooling proposed by [48] uses Choquet integral for pooling.

### 2.1.2 Patch-based pooling methods

In these types of pooling methods, initially objects are detected and the pooling operations are applied on these patches. The examples of patch pooling methods are multi-scale orderless pooling [49], subclass pooling [25], partial mean pooling (PMP) [50] and series multi-pooling [26].

**Multi-scale orderless pooling (MOP)** [49] is inspired from spatial and feature space pooling of local descriptors [33]. MOP improves the invariance property of pooling without lowering the discriminative power, fine tuning on target datasets. The deep activation features are extracted at different scales. These scales are coarser and local patch scales. The coarser scale corresponds to the whole image, and fine scale corresponds to the local region. The coarser

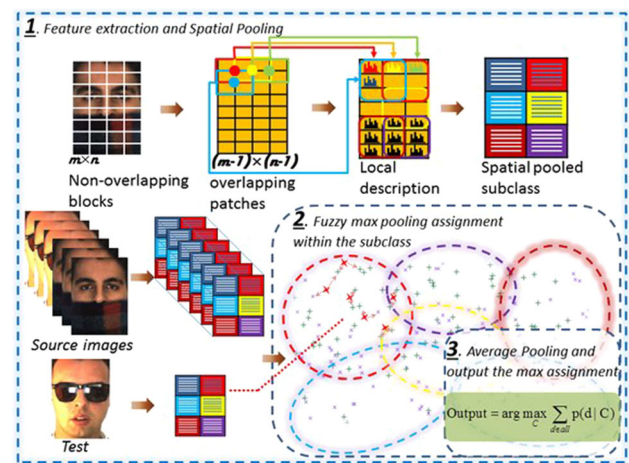


Fig. 12 Framework of subclass pooling [25]

scale preserves the global spatial layout, while finer scales allow to capture more local, fine-grained details of data. These fine-grained details are aggregated via vectors of locally aggregated descriptors (VLAD) encoding, which has an orderless nature and thus contributes to amore invariant representation. Finally, the initial global deep activations and the VLAD encoded features are concatenated to form a new image representation. The MOP for CNN is more potent translation, rotation and scaling [49]. Their method proved successful at a wide variety of applications, including scene classification, data retrieval and, most significant, image classification producing competitive results on MIT indoor scenes classification datasets.

**Subclass pooling (SCP)** [25] is three layered which addresses the issue of double obstruction with a partial training data. This scheme preserves the high-level spatial information and suppresses occlusions and other noises, hence boosts the overall performance. Initially, local features are pooled to preserve the spatial correlation into subclasses according to spatial areas. The fuzzy max pooling is applied during the test phase in order to conquer the erratic local features from obstructed areas (see Fig. 12). The final average pooling enhances the robustness by routinely weighting on every subclass. This method is found robust to various occlusions in random patterns.

**Series multi-pooling** [26] scheme is inspired by the SP pooling [8] and associated with selected patch of feature map. It creates the multi-scale features and extracts rich features with expanded the patch area. This will overcome the confines of local (see Fig. 13) series structure in the input data. The expanded patch range is used due to different lesion sizes of each patient. It captures the surrounding area features aligned on central pixel which is expressed as  $f = [f_0, f_1, f_2]$ , and corresponding areas are  $R_0, R_1, R_2$ .



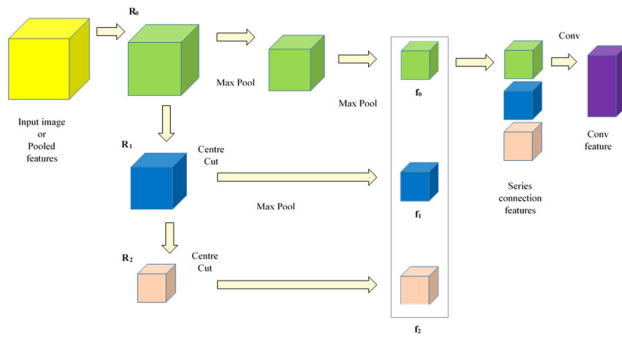


Fig. 13 Framework of series multi-pooling [26]

The region size  $R_0$  is  $w \times w \times n$ , where  $n$  represents the dimension. The relationship between region size and feature is expressed by following

$$f_i = \text{MaxPooling}^{(2-i)}\{R_i\} \quad (9)$$

In above Eq. 9,  $(2-i)$  represents maximum pooling operation times.

The central region  $R_1, R_2$  has dimensions of  $(w/2) \times (w/2) \times n$  and  $(w/4) \times (w/4) \times n$ , respectively. The different regions  $R_0, R_1$  and  $R_2$  are input original image, area cut out from the center of  $R_0$  and  $R_1$ . The regions  $R_0, R_1$  and  $R_2$  correspond to  $f_0, f_1$  and  $f_2$  with twice, single and without max pooling operations. The  $f_2$  is directly connected in series with  $f_1$  and  $f_0$ . The multi-pooling is attached to the top of input series connection design that expands input patch, eludes the introduction of redundant information. It is tested on BRATS2015 dataset which shows the accuracy improvement by 17%.

**Partial mean pooling (PMP)** [50] uses two stages to pool the patch-based features (see Fig. 14). The two stages of PMP are intra- and inter-patch pooling. The intra-patch step captures the discriminative responses and filters the harmful belongings of position variance on feature maps, while inter-patch step transforms these feature maps to low dimensions. The input feature map is sorted in descending order and evaluates the average value of top- $K$  responses to get the pooling feature from input feature map. The PMP seek a trade-off between max pooling and average pooling. In inter-patch pooling stage, different patch features are accumulated to form a global demonstration. The L2 normalization and PCA can be applied to achieve the improvement in the discriminating ability. The

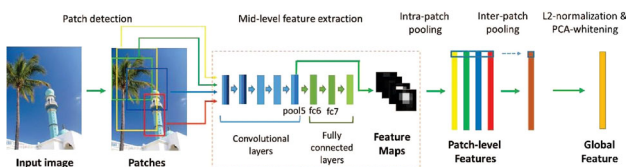


Fig. 14 Framework of partial mean pooling [50]

experimental results prove the superiority over max and average pooling in terms of accuracy on several benchmark datasets.

**Top-K pooling** [51] is the variant of PMP pooling [50]. During the forward pass, the feature map size reduces and noise influence become more severe [50]. Top-K pooling improves the performance by reduces the effects noise on feature map during training phase. This pooling scheme then calculates the mean of top- $K$  features from organizing responses in pooling region. The max pooling schemes are mostly susceptible to noise, while top- $K$  pooling performs well in fetching the statistics of responses. The top- $K$  pooling response changes from max to average pooling with variation in  $K$  from unity to window size. The def-pooling [52] layers learn the deformation of object chunks of various sizes and semantic meanings. After training the def-pooling layer, the object part will result in high activation value. A scale-dependent pooling is proposed by [53] to tackle the scale variation with improvement in detection accuracy on small objects.

### 2.1.3 Multisampling pooling methods

The down sampling operation in pooling learns spatially invariant features and reduces computational costs, depends on the tuned hyperparameters. The down sampling operation does not use the full spectrum of input features, since it rejects around 75% information, while resolution scales down quadratically in a 2D CNN. To overcome these issues, two types of sampling and hence pooling methods are proposed checkerboard subsampling and parallel grid pooling by [54, 55]. The multisampling better scales the spatial resolution of the output feature map while preserving the benefits of traditional subsampling layers such as increasing receptive field and reducing computational costs. This results in advantages like forward pass producing higher resolution feature maps, better gradient updates for deep layers during training and streamlining CNN design by reducing the need for dilated convolutions. It improves the accuracy of image classification by simply applying multisampling with no data augmentation is used. Initially, the feature map is fragmented into  $k \times k$  sampling windows. Consider the input image having size of  $6 \times 6$  is applied to pooling layer with stride value of two. Suppose we have chosen the blue element as shown in Fig. 15. This operation results output feature map having the size of  $3 \times 3$  called as submap. These two submaps are processed separately resulting in a total of four submaps. The process of generating multiple submaps at a subsampling layer is called multisampling. This particular instance of multisampling is called checkerboard subsampling [54] due to the checkerboard.

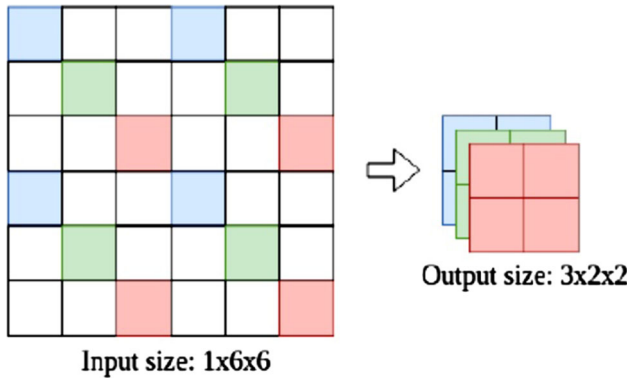


Fig. 15 Checkered sampling [54]

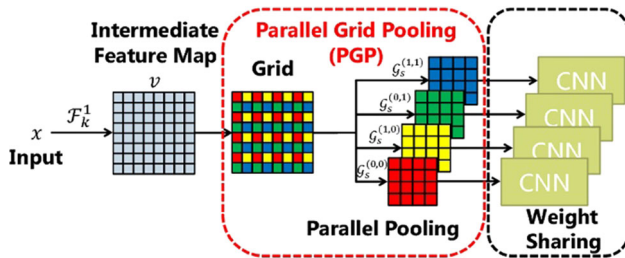


Fig. 16 Parallel grid pooling scheme [55]

*Parallel grid pooling* (PGP) [55] is applicable to various CNN models without altering their learning strategy (see Fig. 16). The PGP down samples the feature map without discarding any intermediate feature and can be regarded as a data augmentation technique. Furthermore, they demonstrate that a dilated convolution can naturally be represented using PGP operations, which suggests that the dilated convolution can also be regarded as a type of data augmentation methods.

The pooling operation depends on two hyper parameters, viz. stride and kernel size. Initially, pooling is applied on input feature map with unity stride value which makes full use of the input feature map information. It will produce an intermediate feature map. This intermediate feature map is separated into  $(w/s) \times (h/s)$  blocks (grid) of size  $s \times s$ . All these down sampled grids are processed in parallel for pooling operation, hence named as parallel grid pooling. Note that the weights (i.e., parameters) of the following network are shared between all branches; hence, there is no additional parameter throughout the network. PGP performs down sampling while maintaining the spatial structure of the intermediate features, thereby producing output features shifted by several pixels. This works as data augmentation; with PGP, the layers are trained with  $s^2$  times larger number of mini-batches compared to the case without PGP. Experiments on CIFAR-10/100 [42] and SVHN [15] with six network models demonstrated that PGP.

## 2.2 Probability-based pooling methods

This type of pooling calculates the probability to make trade-off between the average and max pooling. The features of both max pooling and average pooling are reflected in the pooling process by introducing the mixing mechanism. The various methods based on probability are stochastic, fractional max, dropout max, failure density, hybrid, region of interest and mixed gated and tree pooling method. The probability-based pooling method helps in improving the error rate and prevents the overfitting.

The max pooling produces better result [7, 21], but it suffers with over fitting issue and affects the generalization ability on test data. The average pooling takes the average of all activation in pooling region. A biological inspired pooling is proposed by [56] and is modeled on complex cells. The theoretical analysis is done by [57] and suggests better regularization of  $L_p$  pooling [58] over max pooling. The weighted average of activation over the pooling  $R_j$  is given as

$$S_j = \left( \sum_{i \in R_j} a_i^p \right)^{\frac{1}{p}} \quad (10)$$

The variation rate of  $p$  decides the type of pooling region, such as  $p = 1$  corresponds to max-average pooling, while  $p = \infty$  results in max pooling. The  $L_p$  pooling improves the error rate compared to max pooling; it resulted in exceptional image classification results and a new state of the art on the Street View House Numbers (SVHN) [15] classification benchmark.

The regularization method plays crucial role for the successful applications of neural networks. The max and average pooling suffers with regularization effect of dropout. A new dropout inspired regularization method named stochastic pooling [22, 59] replaces the deterministic average and max pooling techniques, since this pooling method suffers with regularization effect of dropout. It is simple and applicable to CNN with positive nonlinearities and achieves good performance on several tasks. The stochastic pooling arbitrarily preferences the activations according to a multinomial distribution; hence, non-maximal activations of feature maps are utilized. A set of probabilities  $p$  are evaluated for each region  $j$  by normalizing the activation.

$$p_i = \frac{a_i}{\sum_{k \in R_j} a_k} \quad (11)$$

A location from multinomial distribution is occupied based on probability value. The pooled activation is formulated as,

$$s_j = a_l \quad \text{where } l \sim P(p_1, \dots, p | R_j) \quad (12)$$

Although stochastic pooling has the same benefits as max pooling, its stochastic nature helps in improving error rate and prevents overfitting, thus making it an effective network regularization technique that can be combined with approaches like dropout [60, 61] and data augmentation [21]. It does not require any hyperparameter for tuning and reduces training and testing errors. The stochastic pooling selects the random values; it may select 0.1 as the pooling result, which is inappropriate for the network. In the process of pooling, we should ignore the relatively small values. In order to select the values which are representative and take more values into account, [62] proposes restricted stochastic pooling. The restricted stochastic pooling [62] is the blend of max and stochastic pooling which randomly selects value from the first  $n$  larger values in each pooling region. Initially, all values of pooling region are sorted pick out the first  $n$  larger values. This scheme selects the random value from these selected values. The restricted stochastic pooling is represented as

$$y_{\text{restricted}} = \text{random}(S_n) \quad (13)$$

In the above Eq. 13 random is the process which randomly selects the first  $n$ th larger values of  $S_n$ . If  $n$  is three then three values are selected shown in gray color have same probability of selection (see Fig. 17). The selected value of  $n$  affects the accuracy and processing time. The optimum value found by [62].

*Stochastic spatial sampling pooling (S3Pool)* [63] is two step method that uses stochastic down sampling. A pooling window ( $2 \times 2$ ) glides over the feature map with unit stride value tailed by the down sampling. The down sampling picks single value from non-overlapping pooling region in uniform and deterministic manner (see Fig. 18). From signal processing point of view this is not the optimal way of reconstructing the signal. The S3Pool method replaces the general down sampling step by stochastic spatial sampling (S3Pool). The blend of stochastic and S3Pool work as a strong regularization and data augmentation step by introducing distortions in the feature maps.

0.1	1	8.1	9
0.6	8.7	1.5	8.5
0.5	3	0.2	0.2
1.7	2.1	1.1	3

Fig. 17 Sample Image showing the selection criteria of restricted stochastic pooling [62]

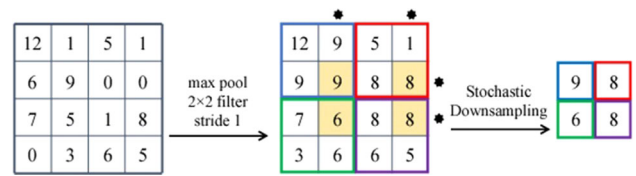


Fig. 18 S3Pool, pooling window  $k = 2$ , stride  $s = 2$ , grid size  $g = 2$  [63]

The S3Pool partitions the feature map of  $h \times w$  into  $p = (h/g)$  vertical and  $q = (w/g)$  horizontal strips, here  $g$  represents grid size. It selects arbitrarily  $(g/s)$  rows and  $(g/s)$  columns to acquire the down sampled feature map of size  $(h/s) \times (w/s)$ . This down sampling is stochastic in nature; hence, it produces different feature maps for training which amounts to perform a type of data augmentation at intermediate layers which yields unlike feature maps at each pass for the same training examples, which amounts to implicitly performing a sort of data augmentation, but at intermediate layers. The grid size controls the distortion for adapting the CNN with designs and datasets. This is useful to house the CNN with multiple pooling, which ultimately controls the trade-off between regularization strength and converging speed. The S3Pool performs “virtual” data augmentation and hence acts a strong regularizer. The S3Pool is fast to compute during training phase and does not require additional parameters. It introduces little computational overhead over the general max pooling.

*Max pooling dropout* [10] uses the combination of max pooling and drop out technique (see Fig. 19). During the training phase activations are randomly preference using multinomial distribution. During the testing phase probabilistic weighted pooling is used, which acts a model averaging. The probabilistic weighted pooling fits training data in well manner as well as generalizes the testing data better than the max and scaled max pooling. The probabilistic pooling performs very well on small retain probability as compared to the max and scaled max probability.

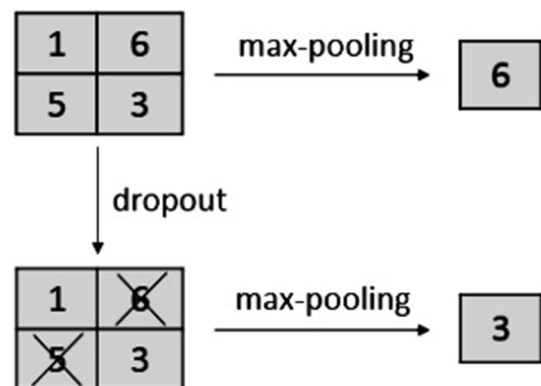
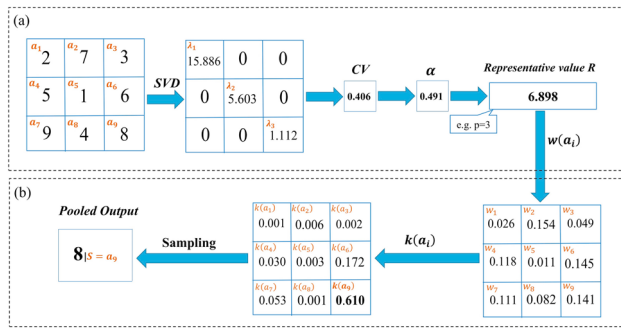


Fig. 19 Procedure of max pooling dropout [10]



**Fig. 20** Framework of sparsity-based stochastic pooling [64]

This performance gap reduces as retaining probability increases. Max pooling dropout with typical retaining probabilities (around 0.5) often outperforms stochastic pooling with greater margin. Experimental evidence endorses the advantage of using max pooling dropout, and confirms the dominance of probabilistic weighted pooling over max and scaled max pooling.

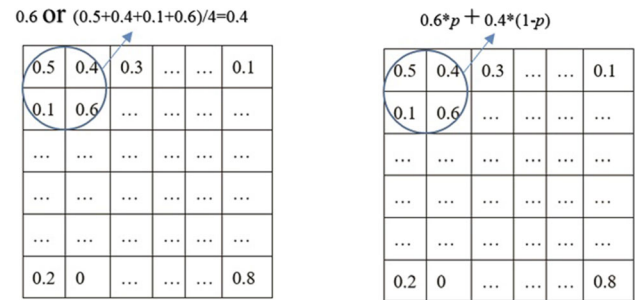
*Sparsity-based stochastic pooling* [64] incorporates the leads of max, average and stochastic pooling. A degree of sparsity is introduced for acquiring the optimized feature in pooling region (see Fig. 20). The feature value is stretched from average to maximum value. This optimized feature value is engaged for probability weights assignment of activations in normal distribution. This method uses weighted random sampling in order to preserve the advantages of stochastic pooling. The non-stationary nature of image feature and stochastic nature of pooling regions improves the performance of pooling [65]. This method is tested on benchmark datasets like MNIST [19], CIFAR-10, CIFAR-100 [42], and SVHN [15] that reflects the improvement in recognition accuracy.

*Hybrid pooling method* (HPM) [66] uses maximum and average pooling. During training stage the convolution feature map is detached to two regions for max and average pooling. A max and average pooling method is applied for the probability of  $p$  and  $1-p$ . The optimal  $p$  is around 0.75. The output of these combined method is weighted average of the two methods. It is represented as

$$y_{vw} = (p \times \max(x_{ij})) + ((1-p) \times \text{mean}(x_{ij})) \quad (14)$$

Figure 21 shows the procedure for max and average pooling while right one shows calculation of hybrid pooling method during the testing phase.

The authors [66] trained the model with mini-batch gradient descent method. The batch size, momentum and learning rate is set at 50, 0.99 and 1. It is found experimentally that the HPM produces better generalization ability of CNN on MNIST dataset if the mixing probability is properly adjusted.



**Fig. 21** Comparison of hybrid pooling scheme and normal pooling (max or average) [66]

*Failure probability density pooling* The failure probability theory is utilized by [67] for CNN regularization. The pooling dispense failure probability density (FPD) to the activations of feature map. This feature map image contains eigenvectors from high to low dimensions to sustain the association of high-dimensional image features. The weights are updated by assigning FPD to the activations of feature map. It is tested on CIFAR-10, CIFAR-100 [42], and SVHN [15] and equated with advanced dropout, maxout, and stochastic pooling methods for classification task in terms of speed and accuracy. The experimental results reflects the superiority of failure density probability pooling.

*Mixed gated tree pooling* is an improvement over the mixed pooling [22] where random coefficient are replaced with a real number stretching from 0 to 1. This real number assigns the weights of maximum and average values. This mixing proportion mechanism reflects the features of max and average pooling, while the randomness of the sampling process is sacrificed. The stochastic pooling provides the weight probability of activation and picks the activation based on this probability. The gated max–average pooling is formulated as

$$f_{\text{gate}}(x) = \sigma(w^T x) f_{\text{max}}(x) + (1 - \sigma(w^T x)) f_{\text{avg}}(x) \quad (15)$$

Above Eq. 15, the values in pooling region are signified by  $x$  and the values of the gating mask are denoted by  $w$ . The first approach is further classified as responsive (mixed max–average pooling) and unresponsive (gated max–average pooling) to the characteristics of pooling region. The mixed max–average pooling approach results in some specific, unchanging blend of max and average. The gated max–average pooling approach uses a gating mask to govern a “responsive” blend of max and average pooling. A sigmoid function is applied to inner-product between the region to be pooled and gating mask. The output of this stage uses mixing balance amid max and average. Furthermore, inspired by [68], who incorporated MLPs with decision trees, [69] used a binary decision tree to learn a combination of previously learned individual



pooling filters. A particular incarnation of their approach, which combined their tree and max-average methods, achieved state-of-the-art results on several benchmarks. In particular, they outperformed several high-performing convolutional networks such as NIN [28], stochastic pooling [22], the DCNNs presented by [24], maxout networks and drop connect networks [60] on various image classification benchmarks, including the MNIST [19], CIFAR-10 and CIFAR-100 [42], and SVHN [15] datasets. Notably, despite their successes, RCNNs outperformed them on the CIFAR-100 [42] dataset. Furthermore, for future DCNNs to readily incorporate decision analysis tools such as decision trees into their architectures, further work on reducing the computational costs and exorbitant number of model parameters required by such models is still required.

### 2.3 Rank-based pooling methods

In these types of pooling, the activations of feature map in pooling region have different weights and are combined together via a weighted sum. These weights are learned during the gradient-based optimization or training. A key difference between ordinal pooling network (OPN) and a pooling operation is that while a typical pooling acts upon one feature map at a time, OPN consists of a different set of weights for each feature map, and therefore, pools feature from all the feature maps simultaneously. Figure 22 shows the comparison between value, location and rank-based pooling methods.

*Multipartite pooling* [41] based on the multipartite ranking of the features in pooling layers of deep CNN. The Fisher discrimination is used to map features into a space. This mapping is used as a measure to rank the existing features, with respect to their specific discriminant power, for each class. The multipartite ranking is used to score the separability of instances, and to aggregate one versus all scores, giving an overall distinction score for each features. Therefore, this pooling scheme projects the features to a new space and then score them by an accumulative bipartite ranking approach. The feature selection operator picks the most informative and highest scores convolutional features in a pooling window, by learning a multipartite ranking scheme from the training set. Inspired by stochastic pooling, higher ranked activations in each

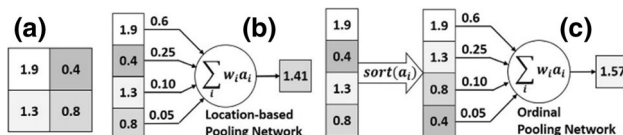
window are picked with respect to their scoring function responses. This leads to an efficient spread of responses and effective generalization for deep CNN. The performance consistently improves in all the experiments. Authors conducted experiment on four publicly available datasets (MNIST [19], CIFAR-10, CIFAR-100 [42], and SVHN [15]) and report the errors of four different pooling schemes (maximum, average, stochastic [59, 22] and multipartite). This multipartite pooling method outperforms on standard benchmark datasets all other pooling strategies (average, maximum and stochastic pooling) with identical evaluation protocols. It also provides a more efficient generalization for the deep learning architectures. The multipartite pooling considers the distribution of each class and calculates the rank of individual features. Due to the data driven process of scoring, the performance gap between training test errors is considerably closer. The conducted experiments on various benchmarks confirm that the proposed strategy of multipartite pooling consistently improves the performance of deep convolutional networks, by using better model generalization for the test time data.

*Ordinal pooling* [70] process is used to assign and arrange different weights to all activations of feature map in pooling region. These arrangements are based on their rank and order of sequence, and these are combined via weighted sum method. These weights are learned during the gradient-based optimization or training. A key difference between ordinal pooling network (OPN) and a pooling operation is that while a typical pooling acts upon one feature map at a time, OPN consists of a different set of weights for each feature map, and therefore, pools feature from all the feature maps simultaneously. Owing to this fact, OPN is referred in this work as a pooling network rather than a pooling operation.

The idea of a rank-based weight aggregation was first introduced by [71], who proposes a global weighted rank pooling (GWRP). The GWRP estimates a score associated with a segmentation class. The GWRP works on the elements of feature map to evaluate the score of segmentation class. In GWRP, all the elements of a feature map are first sorted in the descending order, depending upon their scores for a particular segmentation class. However, the weights that are assigned based on the order of the elements are determined from a hyperparameter and therefore do not change during the training. A confidence weighted pooling is proposed by [72] for color constancy. The mathematically ordinal pooling is represented by

$$s_j = \sum_{i \in R_p} w(\text{ord}(a_{ij})) a_{ij} \quad \forall j \in [1, N] \quad (16)$$

In this,  $a_{ij}$  represents all the activations within the pooling region  $R_p$  in a feature map, and  $N$  represents total number of feature maps. The order  $\text{ord}()$  determines the



**Fig. 22** Pooling operations. **a**  $2 \times 2$  pooling region, **b** location-based and **c** ordinal pooling network [70]

weights of activation. The weights of pooling region will be same if order of sorted sequence remains same. The ordinal pooling has same number of parameters as that of location-based pooling. The ordinal pooling generalizes both average and max pooling due to the nonlinearity introduced by sorting step. For example, if all the weights are equal to  $1/R_p$ , then average pooling scheme will be selected. Similarly if the maximum activation value is 1 and rest are 0 then max pooling will be selected. These sorting steps avoid the effects of under and over valuation of larger or smaller activation, hence preserve the information in most efficient manner. The ordinal pooling is experimented on MNIST [19] dataset; it found that classification accuracy is improved by 0.10%; hence, convergence is faster.

*Global weighted rank pooling (GWRP)* [71] estimates a score associated with a segmentation class. The GWRP works on the elements of feature map to evaluate the score of segmentation class. In GWRP, all the elements of a feature map are first sorted in the descending order, depending upon their scores for a particular segmentation class, which is similar to our case. However, the weights that are assigned based on the order of the elements are determined from a hyperparameter and therefore do not change during the training. A rank-based pooling mechanism generates the pooling output based on the weighted sum of activations [65]. It identifies the important activation using evaluated rank, hence achieves better performance. There are 3 different weighting strategies namely, rank-based average pooling (RAP), rank-based weighted pooling (RWP) and rank-based stochastic pooling (RSP) as proposed by [65] and as shown in Fig. 23. In rank-based

pooling initially, all activations are sorted within the pooling region in descending order and ranks  $a(i)$  are assigned based on their position. In this, higher ranks are assigned to lower activations

$$a(i) > a(j) \Rightarrow r(i) < r(j) \quad (17)$$

When two activations have same value, then above equation is modified as

$$a(i) = a(j) \wedge i < j \Rightarrow r(i) < r(j) \quad (18)$$

The RAP is regarded as the trade-off between the max and average pooling. A defined rank threshold eliminates the near zero activations. The weights of nominated activations are set to be  $1/t$ , while rests are set as 0. The output is calculated as

$$S_j = \frac{1}{t} \sum_{i \in R_j, r_i \leq t} a_i \quad (19)$$

In above equation,  $R_j$  is the pooling region selected in  $j$ th feature map with  $i$  as index of activation. The  $r_i$  and  $a_i$  are the activations rank. The changes in the value of  $t$  are from unity to pooling size which corresponds to the selection of max to average pooling. The RAP filters out the negative activations considering only high responses, hence preserves the important information and improves the discriminating capabilities.

In RWP, each activation is weighted by a coefficient in the pooling region. It assigns larger weights to more important and higher activation which significantly improves the performance. The ranking can be linear as well as nonlinear function as given by Eq. 20

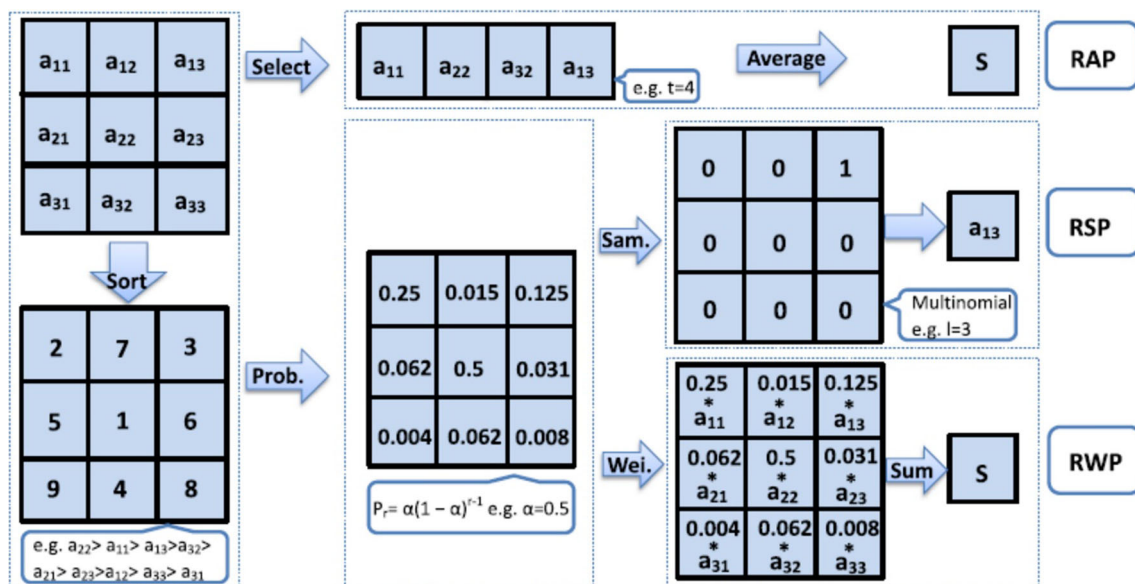


Fig. 23 Rank-based pooling scheme [65]

$$p_r = \alpha(1 - \alpha)^{r-1}, \quad r = 1, \dots, n \quad (20)$$

In above equation,  $r$  and  $n$  represent the hyperparameter, rank of activations and size of pooling region. The activations in each region are weighted by the probability  $p_i$

$$p_i = \frac{a_i}{\sum_{k \in R_j} a_k} \quad (21)$$

The RSP substitutes the conventional pooling operations by stochastic procedure [73]. A multinomial distribution produces the probabilities  $p$  and provides the weight to the activation. The value of  $\alpha$  (alpha) reflects the two cases of RSP, when the reflected value is unity then max pooling will be selected, otherwise the reflected value less than unity, stochastic pooling will be selected. The backward propagation propagates the gradients through the nominated activations and updates the parameters of nominated activations, while others are frozen. Experimental results shows the effectiveness of these pooling on MNIST [19], CIFAR-10, CIFAR-100 [42] and NORB [16] datasets.

## 2.4 Transformed domain-based pooling methods

Pooling operation reduces the spectral variance on the input features maps. The max pooling is the utmost common pooling strategy. It helps in reducing spectral variance by eliminating the variability in the time frequency space. This variability occurs due speaking charms, channel distortions, etc. The padding in time and frequency pooling is effective for deep CNN [74]. The speech processing is mostly explored in frequency dimensions [75, 76], though [77] did investigate CNNs with pooling in time, but not frequency. However, most CNN work in vision performs pooling in both space and time (i.e.,  $x$  and  $y$  dimensions) [7]. Heterogeneous pooling [78] provides constrained frequency shift invariance with minimal speech class confusion in the speech spectrogram. It is compared with the fixed size pooling. The larger pooling size imposes large invariance in frequency shift, not differentiated with similar formant frequencies among the different speech sounds. The fixed pooling size increases the confusion the close major formant frequencies.

The pooling size relates a trade-off between the desired invariance over a range of frequency shift and the undesirable phonetic confusion which appears due to the distinct phones' formants within the range. The heterogeneous pooling uses different pooling size to numerous subclasses of the full feature maps.

**Wavelet pooling** [79] The general form of pooling operations employs a neighborhood approach to subsampling, reminiscent of nearest neighbor interpolation in image processing. Neighborhood interpolation techniques

perform fast, with simplicity and efficiency, but introduce artifacts such as edge halos, blurring and aliasing. Minimizing discontinuities in the data are critical to aiding in network regularization and increasing classification accuracy. The wavelet pooling algorithm [79] uses second-level wavelet decomposition and discards the first-level subbands for reducing the feature dimensions. This approach forgoes the nearest neighbor interpolation method in favor of an organic, subband method that more accurately represents the feature contents with less artifacts. This method addresses the overfitting issue of max pooling, while reducing features in a more structurally compact manner than pooling via neighborhood regions. This method is compared with max, average, mixed and stochastic pooling on benchmark image classification datasets such as MNIST [19], CIFAR-10 [42], Street House View Numbers (SHVN) and Karolinska Directed Emotional Faces (KDEF). Experimental results on four benchmark classification datasets demonstrate that proposed method outperforms or performs comparatively with methods like max, mean, mixed and stochastic pooling. After performing the second-order decomposition, the image feature is extracted using second-order wavelet subbands. The pooling is done using inverse DWT.

**Spectral pooling** [80] reduces the dimensions in frequency domain by truncating the lower frequencies in power spectrum. It preserves more information per parameter with same number of parameters [80] than other pooling methods. The main idea of spectral pooling is that, it truncates the concentrated lower frequencies in power spectrum while filtering the high frequencies that acts as noise, hence lowers the information loss with any arbitrary output map dimensions. This reduces the feature map dimensions in slow manner as a function of network depth. It is achieved by applying discrete Fourier transform (DFT) on input feature map. It is assumed that the DC component is present at the center of the domain and truncates the frequency representation of central  $H \times W$  submatrix of frequencies. This central submatrix is governed by the dimensions of output feature map. Finally, the inverse DFT maps the truncated representation back to the spatial domain. This can be implemented at a negligible additional computational cost in CNN that employ Fast Fourier Transform (FFT); spectral pooling offers the advantage like it maintains more information and does not suffer from the sharp reduction in output dimensionality exhibited by other pooling techniques. It allows flexibility in reducing the map size gradually as a function of layer. The pooling approach adopted by [81] uses spectral axis. The spectral pooling suffers with large amount of computational consumption [82]. The author [82] uses FFT-based convolution with spectral pooling. The pooling method proposed by [83] and [84] uses the Hartley transform and discrete

cosine transform. In all convolutional neural network (All-CNN) [85], the pooling is replaced with another convolutional layer of equivalent stride and filter size. Another approach, RNN-based pooling [86], replaces the pooling operation with a long short-term memory (LSTM) unit, which is a variant of recurrent neural network (RNN). In this case, after all the activations in each pooling region have been scanned sequentially, the final output from RNN layer is returned as the pooled value.

Recently, a novel architecture is introduced by [87] as Caps Net overcomes the issue of information loss. The Caps Net has group of neurons called as capsules. The outputs from the capsules in one layer are routed to the capsules in the subsequent layer based upon the assignment coefficients which are determined from the expectation maximization algorithm

A novel feature pooling method utilizes the region structure information adaptively based on different exemplars, referred as adaptive region pooling [88]. The adaptive region pooling method extracts features that account for the structure of object parts, which facilitates handling the large variation of objects. The various pooling methods used in action recognition are listed in [89–97]. Attentive pooling approach [98] is compared with single max pooling for drug–drug interaction extraction.

### 3 Discussion

The convolution operation in CNN detects the different level features like low, mid and high levels. The pooling operation reduces the dimensions of such features. The feature maps of early layers preserve more important information as compared to later one. The early layer feature map captures local features, while later layer captures global features. A larger feature map in the early layer produces higher accuracy, but requires more memory usage and computation time. The max pooling in the early stage and average pooling in later stage would be a better choice for better results. The rapid early down sampling causes more information loss; hence, it must be avoided.

The max pooling and average pooling are the two choices of pooling, but these pooling suffer in certain situation; hence, variants are evolved by numerous researcher based on different criteria. These criteria are based on pooling hyperparameters (stride, filter size), input output feature map, location of pooling, number of stages, way of down sampling and random selection based on probability value. The pooling process in this article reviewed is based on the value, rank, probability and transformed domain approach. The down sampling is either sudden or gradual, and it may use multisampling approach. The sudden way of down sampling rejects about 75% information, which

vanishes most of important features. The FMP [13] avoids sudden rejection of spatial information in more gradual way by introducing a parameters, while its variant BW-FMP [43] improves the memory consumption and processing time by 18% and 13%, at the cost of classification accuracy.

Another way of down sampling to preserve the features information are proposed by [54] and [55] as like multi-sampling process. The multisampling process scales the spatial resolution of the output feature with benefits of traditional subsampling layers such as increasing receptive field and reducing computational costs. The parallel grid pooling [55] down samples the feature map without discarding any intermediate feature, and these features are processed in parallel operation.

The average pooling considers all activation with equal contribution in pooling region. This process downplays the higher activation values and considers all other activation also. If average pooling is used at the earlier layers, it will significantly reduce the accuracy of system, while in upper layers difference is less severe. The  $K$  support spatial pooling sorts all activation in pooling region in ascending order and retains first  $k$  larger activations. The final degree of activation is the mean value of the retained  $k$  values.

The max and average pooling produce better results for certain state. Therefore, a proper choice between the max and average pooling results in better performance under such state. The mixed pooling [9] defines a parameter which selects either methods, but it fails in reflecting the simultaneous advantages of these two methods. The dynamic correlation pooling selects max, average and mixed pooling based on the Mahalanobis distance. The Lp pooling selects between these two pooling method based on the probability value with improvement in error rate. A multinomial distribution is used for selecting the non-maximal activation in stochastic pooling [59, 22]. It offers the benefits of max pooling, while its stochastic nature helps in error rate improvement and prevents overfitting.

A probability value decides the average and max pooling by hybrid pooling [66] and mixed gated tree pooling [69]. The dynamic correlation pooling [26] method uses the correlation between the adjacent value to decide between the two methods. A combination of average, max and average pooling is used by [46]; it is found experimentally that average–average–max combination achieves faster and smaller convergence speed and lowermost error rate.

The scale of input may affects the performance, but SP pooling [8] is designed for such situations, which produces the fixed size output irrespective of input size. The pooling such as SP pooling [8], GA pooling [28], concentric circle pooling [47] and multi-pooling [26] are connected at the top layer. The GA pooling [28] replaces the FC layer,



hence reduce the number of parameters, since most of the parameters of CNN are related to FC layer.

The TI pooling [36] is applied on the top layer before FC layer. It generates new feature from a predefined set of possible transformations, which is independent of rotation and scale of input. TI pooling [36] passes multiple transformed versions of the input separately through the network for transformed instance aggregation with max pooling operation to achieve transform invariances. It is computationally efficient but unable to capture higher-order feature interactions. The higher-order features are captured by Kernel pooling [40] using Gaussian radial basis function. The subclass pooling [25] preserves the high-level spatial information and suppresses occlusions and other noises. It uses fuzzy max pooling and average pooling in order to conquer the erratic local features from obstructed areas. It is found robust to various occlusions in random patterns.

The use of filtering is another way of preserving the important features. The LEAP [31] pooling avoids the use of separate filter for each feature, instead uses shared filter to reduce number of parameters, while detail preserving pooling [30] uses inverse bilateral filter for preserving the important structure details.

The size of pooling region affects the performance, in few situations small size while larger size does not introduce the larger depth of feature map. The multi-activation pooling [45] uses the larger pooling kernels (such as  $4 \times 4$ ,  $8 \times 8$ ,  $16 \times 16$ ) with max pooling for accurate classification without increasing depth and trainable parameters.

Partial mean pooling [50] and S3Pool [63] use two stages for pooling in order to improve the discrimination ability. The PMP [50] method uses intra- and inter-patch pooling. Intra-patch step captures and filters the position variance on feature maps, while inter-patch step transforms it into low dimensions. These features are sorted in descending order and evaluate the average value of top- $K$  responses to get the pooling feature. The S3Pool [63] glides the pooling window over the feature map tailed by the down sampling which selects single feature.

The rank-based pooling applies the weight to the activations of feature map and is combined together via a weighted sum. Multipartite pooling [41] projects the features to a new space and then score them by an accumulative bipartite ranking. It selects most informative and highest scores features using multipartite ranking. The different rank-based pooling listed in literature are GWRP [71], RAP, RWP, RSP [65]

The transformed domain approaches use different domains such as time [77], space [7], frequency [80] and wavelet domain [79]. The frequency transformed domain-based pooling filters the higher frequency by truncating the lower frequency in power spectrum. This transformation is

achieved by using the transforms like DFT [80], FFT [82], Hartley transform [84] and discrete cosine transform [27]. The wavelet pooling [79] uses second-level wavelet decomposition and discards the first-level subbands for reducing the feature dimensions in a more structurally compact manner.

## 4 Future scope

In CNN, pooling layer transforms the pooling kernel feature into most prominent feature which preserves the crucial information. Most of the traditional pooling do not consider the effect of noise in the pooling kernel. It may result in undesirable output, since noise value is accumulated and propagated in the subsequent feature maps. The local features are more affected by the presence of noise. Further pooling layer ignores the activations related to the task and precise location of the object. Therefore, pooling layer should ignore the effects of noise present in the pooling kernel and extract the task-related information. This interpretation can be further investigated with due consideration of three-dimensional CNN and multi-pooling applications.

## 5 Conclusion

In CNN, pooling layer transforms the pooling kernel feature into most prominent feature which preserves the crucial information. The feature maps of early layers preserve more important information as compared to later one. The early layer feature map captures most of the local features, while later layer captures global features. A larger feature map in the early layer produces higher accuracy, but requires more memory usage and computation time. The max pooling in the early stage and average pooling in later stage would be a better choice for better results. The rapid early down sampling causes more information loss; hence, it must be avoided. The value-based pooling methods rely on the selection of an activation value based on certain criteria. Most of traditional value-based pooling methods do not consider the effect of noise in the pooling kernel. It may result in undesirable output, since noise value is accumulated and propagated in the subsequent feature maps. This presence of noise affects most of the local features in feature map. The transformed domain-based pooling can easily filter out such noise because of frequency transformation. This method filters the higher frequency (noise) by truncating the lower frequency in power spectrum. The processing time required for transform-based pooling may be quite larger as compared to other pooling methods. This enlarge time is due to the

transformation from spatial to other and original domain. Additionally, the rank-based method considers all the activation with weighted sum; hence, effect of such noise is diluted with other activations, but it requires more parameters, memory in the form of weights. The probability-based methods avoid such computation timing issues by evaluating the probability of most prominent feature for further routing. A refinement of probability values at different layer results in better performance.

## Compliance with ethical standards

**Conflict of interest** The authors declare that they have no conflict of interest.

## References

- Scherer D, Muller A, Behnke S (2010) Evaluation of pooling operations in convolutional architectures for object recognition. In: Proceedings of the international conference on artificial neural networks, pp 92–101. [https://doi.org/10.1007/978-3-642-15825-4\\_10](https://doi.org/10.1007/978-3-642-15825-4_10)
- Zhong Z, Jin L, Feng Z (2015) Multi-font printed Chinese character recognition using multi-pooling convolutional neural network. In: Proceedings of 13th international conference on document analysis and recognition, pp 96–100. <https://doi.org/10.1109/ICDAR.2015.7333733>
- Springenberg JT, Dosovitskiy A, Brox T (2015) Martin ried-miller, striving for simplicity the all convolutional net. [arXiv: 1412.6806v3](https://arxiv.org/abs/1412.6806v3)
- Chen J, Hua Z, Wang J, Cheng S (2017) A convolutional neural network with dynamic correlation pooling. In: Proceedings of international conference on computational intelligence and security, pp 496–499. <https://doi.org/10.1109/CIS.2017.00115>
- Karpathy A (2017) Stanford University CS231n: convolutional neural networks for visual recognition. <http://cs231n.stanford.edu/syllabus.html>. Accessed 28 Nov 2018
- Li C, Yang SX, Yang Y, Gao H, Zhao J, Qu X, Wang Y, Yao D, Gao J (2018) Hyperspectral remote sensing image classification based on maximum overlap pooling convolutional neural network. *Sensors* 18:3587. <https://doi.org/10.3390/s18103587>
- Krizhevsky A, Sutskever I, Hinton G (2012) Imagenet classification with deep convolutional neural networks. *Adv Neural Inf Process Syst*. <https://doi.org/10.1145/3065386>
- He K, Zhang X, Ren S, Sun J (2015) Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Trans Pattern Anal Mach Int* 37(9):1904–1916. <https://doi.org/10.1109/TPAMI.2015.2389824>
- Yu D, Wang H, Chen P, Wei Z (2014) Mixed pooling for convolutional neural networks. In: Proceedings of the 9th international conference on rough sets and knowledge technology, pp 364–375. [https://doi.org/10.1007/978-3-319-11740-9\\_34](https://doi.org/10.1007/978-3-319-11740-9_34)
- Wu H, Gu X (2015) Max-pooling dropout for regularization of convolutional neural networks. [arXiv:1512.01400v1](https://arxiv.org/abs/1512.01400v1)
- Shi W, Loy CC, Tang X (2016) Deep specialized network for illuminant estimation. In: European conference on computer vision, pp 371–387. [https://doi.org/10.1007/978-3-319-46493-0\\_23](https://doi.org/10.1007/978-3-319-46493-0_23)
- Nagi J, Ducatelle F, Di Caro GA, Ciresan D, Meier U, Giusti A, Nagi F, Schmidhuber J, Gambardella LM (2011) Max-pooling convolutional neural networks for vision-based hand gesture recognition. *Proceedings of the IEEE international conference on signal and image processing applications*, pp 342–347. <https://doi.org/10.1109/ICSIPA.2011.6144164>
- Graham B (2015) Fractional max-pooling. [arXiv:1412.6071v4](https://arxiv.org/abs/1412.6071v4)
- Ranzato MA, Huang FJ, Boureau Y, LeCun Y (2007) Unsupervised learning of invariant feature hierarchies with applications to object recognition. In: *Proceedings. computer vision and pattern recognition*. <https://doi.org/10.1109/CVPR.2007.383157>
- Netzer Y, Wang T, Coates A, Bissacco A, Wu B, Ng AY (2011) Reading digits in natural images with unsupervised feature learning. In: *Proceedings of the neural information processing systems*
- LeCun Y, Huang FJ, Bottou L (2004) Learning methods for generic object recognition with invariance to pose and lighting. *Proc IEEE Conf Comput Vis Pattern Recognit* 2:97–104. <https://doi.org/10.1109/CVPR.2004.1315150>
- Boureau Y, Ponce J, LeCun Y (2010) A theoretical analysis of feature pooling in visual recognition. In: *Proceedings of the 27th international conference on machine learning*, pp 111–118
- Lee H, Grosse R, Ranganath R, Ng AY (2009) Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In: *Proceedings of the international conference on learning representations*, pp 609–616. <https://doi.org/10.1145/1553374.1553453>
- LeCun Y, Bottou L, Bengio Y, Haffner P (1998) Gradient-based learning applied to document recognition. *Proc IEEE* 86(11):2278–2324. <https://doi.org/10.1109/5.726791>
- Fei-Fei L, Fergus R, Perona P (2006) One-shot learning of object categories. *IEEE Trans Pattern Anal Mach Intell* 28(4):594–611. <https://doi.org/10.1109/TPAMI.2006.79>
- Ciresan DC, Meier U, Masci J, Maria Gambardella L, Schmidhuber J (2011) Flexible, high performance convolutional neural networks for image classification. *Proc Int Joint Conf Artif Intell* 1:1237–1242. <https://doi.org/10.5591/978-1-57735-516-8/IJCAI11-210>
- Zeiler MD, Fergus R (2013) Stochastic pooling for regularization of deep convolutional neural networks. [arXiv: 1301.3557v1](https://arxiv.org/abs/1301.3557v1)
- Sainath TN, Kingsbury B, Mohamed A, Dahl GE, Saon G, Soltan H, Beran T, Aravkin Aleksandr Y, Ramabhadran B (2013) Improvements to deep convolutional neural networks for LVCSR. In: *2013 IEEE workshop on automatic speech recognition and understanding*, pp 315–320. <https://doi.org/10.1109/ASRU.2013.6707749>
- Jarrett K, Kavukcuoglu K, LeCun Y (2009) What is the best multi-stage architecture for object recognition?. In: *Proceedings of the IEEE international conference on computer vision*, pp 2146–2153. <https://doi.org/10.1109/ICCV.2009.5459469>
- Long Y, Zhu F, Shao L, Han J (2018) Face recognition with a small occluded training set using spatial and statistical pooling. *Inf Sci* 430–431:634–644. <https://doi.org/10.1016/j.ins.2017.10.042>
- Wang F, Huang S, Shi L, Fan W (2017) The application of series multi-pooling convolutional neural networks for medical image segmentation. *Int J Distrib Sensor Netw* 13:12. <https://doi.org/10.1177/1550147717748899>
- Eom H, Choi H (2018) Alpha-pooling for convolutional neural networks. [arXiv:1811.03436v1](https://arxiv.org/abs/1811.03436v1)
- Lin M, Chen Q, Yan S (2013) Network in network. [arXiv:1312.4400v3](https://arxiv.org/abs/1312.4400v3)
- Zhang B, Zhao Q, Feng W, Lyu S (2018) AlphaMEX: a smarter global pooling method for convolutional neural networks. *Neurocomputing* 321:36–48. <https://doi.org/10.1016/j.neucom.2018.07.079>

30. Saeedan F, Weber N, Goesele M, Roth S (2018) Detail-preserving pooling in deep networks. [arXiv:1804.04076v1](https://arxiv.org/abs/1804.04076v1)
31. Sun M, Song Z, Jiang X, Pan J, Pang Y (2017) Learning pooling for convolutional neural network. *Neurocomputing* 24(8):96–104. <https://doi.org/10.1016/j.neucom.2016.10.049>
32. Grauman K, Darrell T (2005) The pyramid match kernel: discriminative classification with sets of image features. In: *Proceedings of the IEEE international conference on computer vision*, pp 1458–1465. <https://doi.org/10.1109/ICCV.2005.239>
33. Lazebnik S, Schmid C, Ponce J (2006) Beyond bags of features: spatial pyramid matching for recognizing natural scene categories. *Proceeding of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp 2169–2178. <https://doi.org/10.1109/CVPR.2006.68>
34. Jose A, Lopez RD, Heisterklaus I, Wien M (2018) Pyramid pooling of convolutional feature maps for image retrieval. *Proc Int Conf Image Process* 1:480–484. <https://doi.org/10.1109/ICIP.2018.8451361>
35. Chen L-C, Papandreou G, Kokkinos I, Murphy K, Yuille AL (2018) DeepLab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. *IEEE Trans Pattern Anal Mach Intell* 40(4):834–848. <https://doi.org/10.1109/TPAMI.2017.2699184>
36. Laptev D, Savinov N, Buhmann JM, Pollefeys M (2016) TI-POOLING: transformation-invariant pooling for feature learning in convolutional neural networks. [arXiv: 1604.06318](https://arxiv.org/abs/1604.06318)
37. Wu J, Yu Y, Huang C, Yu K (2015) Deep multiple instance learning for image classification and auto-annotation. In: *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp 3460–3469. <https://doi.org/10.1109/CVPR.2015.7298968>
38. Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A (2014) Going deeper with convolutions. [arXiv:1409.4842v1](https://arxiv.org/abs/1409.4842v1)
39. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. *Proc Int Conf Comput Vis Pattern Recognit*. <https://doi.org/10.1109/CVPR.2016.90>
40. Cui Y, Zhou F, Wang J, Liu X, Lin Y, Belongie S (2017) Kernel pooling for convolutional neural networks. *Int Conf Comput Vis Pattern Recognit* 1:3049–3058. <https://doi.org/10.1109/CVPR.2017.325>
41. Shahriari A, Porikli F (2017) Multipartite pooling for deep convolutional neural networks. [arXiv:1710.07435v1](https://arxiv.org/abs/1710.07435v1)
42. Krizhevsky A (2009) Learning multiple layers of features from tiny images. Master's Dissertation, University of Toronto, Canada
43. Hang ST, Aono M (2017) Bi-linearly weighted fractional max pooling: an extension to conventional max pooling for deep convolutional neural network. *Int J Multimed Too Appl* 76(21):22095–22117. <https://doi.org/10.1007/s11042-017-4840-5>
44. Han X-H, Lei J, Chen Y-W (2016) HEp-2 cell classification using K-support spatial pooling in deep CNNs. *LNCS* 10008:3–11. [https://doi.org/10.1007/978-3-319-46976-8\\_1](https://doi.org/10.1007/978-3-319-46976-8_1)
45. Zhao Q, Lyu S, Zhang B, Feng W (2018) Multiactivation pooling method in convolutional neural networks for image recognition. *Wirel Commun Mob Comput*. <https://doi.org/10.1155/2018/8196906>
46. Zhang J, Huang Q, Wu H, Liu Y (2017) A shallow network with combined pooling for fast traffic sign recognition. *Information* 8:45–58. <https://doi.org/10.3390/info8020045>
47. Qi K, Guan Q, Yang C, Peng F, Shen S, Huayi W (2018) Concentric circle pooling in deep convolutional networks for remote sensing scene classification. *Remote Sens* 10:934. <https://doi.org/10.3390/rs10060934>
48. Dias CA et al (2018) Using the choquet integral in the pooling layer in deep learning networks. In: Barreto G, Coelho R (eds) *Fuzzy information processing. NAFIPS 2018. Communications in computer and information science*, vol 831. Springer, Cham
49. Gong Y, Wang L, Guo R, Lazebnik S (2014) Multi-scale orderless pooling of deep convolutional activation features. [arXiv:1403.1840v3](https://arxiv.org/abs/1403.1840v3)
50. Zhi T, Duan L-Y, Wang Y, Huang T (2016) Two-stage pooling of deep convolutional features for image retrieval. *Proc Int Conf Image Process* 1:2465–2469. <https://doi.org/10.1109/ICIP.2016.7532802>
51. Gao F, Lou Y, Bai Y, Wang S, Huang T, Duan L-Y (2017) Improving object detection with region similarity learning. [arXiv:1703.00234v1](https://arxiv.org/abs/1703.00234v1)
52. Ouyang W, Wang X, Zeng X, Qiu S, Luo P, Tian Y, Li H, Yang S, Wang Z, Loy C-C, Tang X (2015) DeepID-Net: deformable deep convolutional neural networks for object detection. *Proc Comput Vis Pattern Recognit*. <https://doi.org/10.1109/CVPR.2015.7298854>
53. Yang F, Choi W, Lin Y (2016) Exploit all the layers: fast and accurate CNN object detector with scale dependent pooling and cascaded rejection classifiers. *Int Conf Comput Vis and Pattern Recognit* 1:2129–2137. <https://doi.org/10.1109/CVPR.2016.234>
54. Sadigh S, Sen P (2018) Improving the resolution of cnn feature maps efficiently with multisampling. [arXiv:1805.10766v1](https://arxiv.org/abs/1805.10766v1)
55. Takeki A, Ikami D, Irie G, Aizawa K (2018) Parallel grid pooling for data augmentation. [arXiv:1803.11370v1](https://arxiv.org/abs/1803.11370v1)
56. Hyvarinen A, Koster U (2007) Complex cell pooling and the statistics of natural images. *Netw Comput Neural Syst* 18(2):81–100. <https://doi.org/10.1080/09548980701418942>
57. Estrach JB, Szlam A, Lecun Y (2014) Signal recovery from pooling representations. In: *Proceedings of the international conference on machine learning*, pp 307–315. [arXiv:1311.4025v3](https://arxiv.org/abs/1311.4025v3)
58. Sermanet P, Chintala S, LeCun Y (2012) Convolutional neural networks applied to house numbers digit classification. In: *Proceedings of the 21st international conference on pattern recognition*, pp 3288–3291
59. Turaga SC, Murray JF, Jain V, Roth F, Helmstaedter M, Briggman K, Briggman W, Denk H, Sebastian, Seung HS (2010) Convolutional networks can learn to generate affinity graphs for image segmentation. *Neural Comput* 22(2):511–538. <https://doi.org/10.1162/neco.2009.10-08-881>
60. Wan L, Zeiler M, Zhang S, LeCun Y, Fergus R (2013) Regularization of neural networks using dropout. In: *Proceedings of the 30th international conference on machine learning*, vol 28(3), pp 1058–1066
61. Hinton GE, Srivastava N, Krizhevsky A, Sutskever I, Salakhutdinov RR (2012). Improving neural networks by preventing co-adaptation of feature detectors. [arXiv: 1207.0580](https://arxiv.org/abs/1207.0580)
62. Fei J, Fang H, Yin Q, Yang C, Wang D (2018) Restricted stochastic pooling for convolutional neural network. In: *Proceedings of the 10th international conference on internet multimedia computing and service*, Article No. 24. <https://doi.org/10.1145/3240876.3240919>
63. Zhai S, Wu H, Kumar A, Cheng Y, Lu Y, Zhang Z, Feris R (2017) S3Pool: pooling with stochastic spatial sampling. [arXiv:1611.05138v1](https://arxiv.org/abs/1611.05138v1)
64. Song Z, Liu Y, Song R, Chen Z, Yang J, Zhang C, Jiang Q (2018) A sparsity-based stochastic pooling mechanism for deep convolutional. *Neural Netw* 105:340–345. <https://doi.org/10.1016/j.neunet.2018.05.015>
65. Shi Z, Ye Y, Wu Y (2016) Rank-based pooling for deep convolutional neural networks. *Neural Netw* 83:21–31. <https://doi.org/10.1016/j.neunet.2016.07.003>

66. Tong Z, Aihara K, Tanaka G (2016) A hybrid pooling method for convolutional neural networks, ICONIP 2016, Part II (LNCS), vol 9948pp. 454–461. [https://doi.org/10.1007/978-3-319-46672-9\\_51](https://doi.org/10.1007/978-3-319-46672-9_51)
67. Zhang Y, Shi B (2017) Improving pooling method for regularization of convolutional networks based on the failure probability density. *Optik* 145:258–265. <https://doi.org/10.1016/j.ijleo.2017.07.045>
68. Buló S, Kotschieder P (2014) Neural decision forests for semantic image labelling. *Proc IEEE Conf Comput Vis Pattern Recognit*. <https://doi.org/10.1109/CVPR.2014.18>
69. Lee C-Y, Gallagher PW, Tu Z (2015) Generalizing pooling functions in convolutional neural networks: mixed, gated, and tree. [arXiv:1509.08985](https://arxiv.org/abs/1509.08985)
70. Kumar A (2018) Ordinal pooling networks: for preserving information over shrinking feature maps. [arXiv:1804.02702](https://arxiv.org/abs/1804.02702)
71. Kolesnikov A, Lampert CH (2016) Seed, expand and constrain: three principles for weakly-supervised image segmentation. *LNCS* 9908:695–711. [https://doi.org/10.1007/978-3-319-46493-0\\_42](https://doi.org/10.1007/978-3-319-46493-0_42)
72. Hu Y, Wang B, Lin S (2017) FC<sup>4</sup> fully convolutional color constancy with confidence-weighted pooling. *Proc Comput Vis Pattern Recognit* 1:330–339. <https://doi.org/10.1109/CVPR.2017.43>
73. Zhang Y-D, Pan Ch, Chen X, Wang F (2018) Abnormal breast identification by nine-layer convolutional neural network with parametric rectified linear unit and rank-based stochastic pooling. *J Comput Sci* 27:57–68. <https://doi.org/10.1016/j.jocs.2018.05.005>
74. Qian Y, Woodland PC (2016) Very deep convolutional neural networks for robust speech recognition. [arXiv:1610.00277v1](https://arxiv.org/abs/1610.00277v1)
75. Abdel-Hamid O, Mohamed A-R, Jiang H, Penn G. (2012) Applying convolutional neural networks concepts to hybrid NN-HMM model for speech recognition. In: *Proceedings of international conference on acoustics, speech and signal processing*, pp 4277–4280. <https://doi.org/10.1109/ICASSP.2012.6288864>
76. Sainath T, Mohamed A, Kingsbury B, Ramabhadran B (2013) Deep convolutional neural networks for LVCSR. In: *Proceedings of international conference on acoustics, speech and signal processing*. <https://doi.org/10.1109/ASRU.2013.6707749>
77. Waibel A, Hanazawa T, Hinton G, Shikano K, Lang K (1989) Phoneme recognition using time-delay neural networks. *IEEE Trans Acoust Speech Signal Process* 37(3):328–339. <https://doi.org/10.1109/29.21701>
78. Deng L, Abdel-Hamid O, Yu D (2013) A deep convolutional neural network using heterogeneous pooling for trading acoustic invariance with phonetic confusion. In: *Proceedings of international conference on acoustics, speech and signal processing*, pp 6669–6673. <https://doi.org/10.1109/ICASSP.2013.6638952>
79. Williams T, Li R (2018) Wavelet pooling for convolutional neural networks. In: *Proceedings of the international conference on learning representations*, vol 6
80. Rippel O, Snoek J, Adams RP (2015) Spectral representations for convolutional neural networks. [arXiv preprint arXiv:1506.03767](https://arxiv.org/abs/1506.03767)
81. Xu Y, Kong Q, Wang W, Plumbley MD (2018) Large-scale weakly supervised audio classification using gated convolutional. *Neural Netw*. [arXiv:1710.00343v1](https://arxiv.org/abs/1710.00343v1)
82. Wang Z, Lan Q, Huang D, Wen M (2016) Combining FFT and spectral-pooling for efficient convolution neural network model. In: *Proceeding 2nd International conference on artificial intelligence and industrial engineering*, vol 133. <https://doi.org/10.2991/aiie-16.2016.47>
83. Zhang H, Ma J (2018) Hartley spectral pooling for deep learning. [arXiv:1810.04028v1](https://arxiv.org/abs/1810.04028v1)
84. Smith JS, Wilamowski BM (2018) Discrete cosine transform spectral pooling layers for convolutional neural networks, artificial intelligence and soft computing. *ICAISC 2018 (Lecture notes in computer science)*, vol. 10841. [https://doi.org/10.1007/978-3-319-91253-0\\_23](https://doi.org/10.1007/978-3-319-91253-0_23)
85. Springenberg JT, Dosovitskiy A, Brox T, Riedmiller M (2015) Striving for simplicity: the all convolutional net. In: *Proceedings of the international conference on learning representations*. [arXiv:1412.6806v3](https://arxiv.org/abs/1412.6806v3)
86. Li S, Li W, Cook C, Zhu C, Gao Y (2017) A fully trainable network with RNN-based pooling. [arXiv:1706.05157](https://arxiv.org/abs/1706.05157)
87. Sabour S, Frosst N, Hinton G (2018) Matrix capsules with EM routing. In: *Proceedings of the international conference on learning representations*. <https://doi.org/10.13140/rg.2.2.27416.44800>
88. Tsai Y-H, Hamsici OC, Yang M-H (2015) Adaptive region pooling for object detection. In: *Proceedings of 2015 IEEE conference on computer vision and pattern recognition*, pp 731–739. <https://doi.org/10.1109/CVPR.2015.7298673>
89. Cherian A, Gould S (2018) Second-order temporal pooling for action recognition. [arXiv:1704.06925](https://arxiv.org/abs/1704.06925)
90. Girdhar R, Ramanan D (2017) Attentional pooling for action recognition. [arXiv:1711.01467v3](https://arxiv.org/abs/1711.01467v3)
91. Wang P, Cao Y, Shen C, Liu L, Shen HT (2015) Temporal pyramid pooling based convolutional neural networks for action recognition. [arXiv:1503.01224](https://arxiv.org/abs/1503.01224)
92. Song S, Cheung N-M, Chandrasekhar V, Mandal B (2018) Deep adaptive temporal pooling for activity recognition. [arXiv:1808.07272](https://arxiv.org/abs/1808.07272)
93. Cherian A, Fernando B, Harandi M, Gould S (2017) Generalized rank pooling for activity recognition. *Proc Comput Vis Pattern Recognit* 1:1581–1590. <https://doi.org/10.1109/CVPR.2017.172>
94. Fernando B, Gavves E, Oramas J, Ghodrati A, Tuytelaars T (2017) Rank pooling for action recognition. *IEEE Trans Pattern Anal Mach Int* 39(4):773–787. <https://doi.org/10.1109/TPAMI.2016.2558148>
95. Fernando B, Gould S (2017) Discriminatively learned hierarchical rank pooling networks. [arXiv:1705.10420v1](https://arxiv.org/abs/1705.10420v1)
96. Wang P, Li W, Gao Z, Tang C, Ogunbona P (2018) Depth pooling based large-scale 3D action recognition with convolutional neural networks. [arXiv:1804.01194](https://arxiv.org/abs/1804.01194)
97. Kar A, Rai N, Sikka K, Sharma G (2017) AdaScan: adaptive scan pooling in deep convolutional neural networks for human action recognition in videos. *Proc IEEE Conf Comput Vis Pattern Recognit*. <https://doi.org/10.1109/CVPR.2017.604>
98. Suárez-Paniagua V, Segura-Bedmar I (2018) Evaluation of pooling operations in convolutional architectures for drug-drug interaction extraction. *BMC Bioinform*. <https://doi.org/10.1186/s12859-018-2195-1>
99. Girshick R (2015) Fast R-CNN. In: *International conference on computer vision*, pp 1440–1448. <https://doi.org/10.1109/ICCV.2015.169>
100. Gulcehre C, Cho K, Pascanu R, Bengio Y (2014) Learned-norm pooling for deep feed forward and recurrent neural networks. [arXiv:1311.1780v7](https://arxiv.org/abs/1311.1780v7)

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.