Received: 15 April 2024 | Accepted: 30 May 2024

DOI: 10.1111/ahe.13073

ORIGINAL ARTICLE

Anatomia Histologia Embryologia WILEY

Histological tissue classification with a novel statistical filter-based convolutional neural network

Nejat Ünlükal[1] | Erkan Ülker[2] | Merve Solmaz[1] | Kübra Uyar[3] | Şakir Taşdemir[4]

[1]Department of Histology and Embryology, Selcuk University, Konya, Turkey

[2]Department of Computer Engineering, Konya Technical University, Konya, Turkey

[3]Department of Computer Engineering, Alanya Alaaddin Keykubat University, Antalya, Turkey

[4]Department of Computer Engineering, Selcuk University, Konya, Turkey

Correspondence
Nejat Ünlükal, Department of Histology and Embryology, Medicine Faculty, Selcuk University, 42130 Konya, Turkey.
Email: nejatunlukal@gmail.com

Abstract

Deep networks have been of considerable interest in literature and have enabled the solution of recent real-world applications. Due to filters that offer feature extraction, Convolutional Neural Network (CNN) is recognized as an accurate, efficient and trustworthy deep learning technique for the solution of image-based challenges. The high-performing CNNs are computationally demanding even if they produce good results in a variety of applications. This is because a large number of parameters limit their ability to be reused on central processing units with low performance. To address these limitations, we suggest a novel statistical filter-based CNN (HistStatCNN) for image classification. The convolution kernels of the designed CNN model were initialized by continuous statistical methods. The performance of the proposed filter initialization approach was evaluated on a novel histological dataset and various histopathological benchmark datasets. To prove the efficiency of statistical filters, three unique parameter sets and a mixed parameter set of statistical filters were applied to the designed CNN model for the classification task. According to the results, the accuracy of GoogleNet, ResNet18, ResNet50 and ResNet101 models were 85.56%, 85.24%, 83.59% and 83.79%, respectively. The accuracy was improved by 87.13% by HistStatCNN for the histological data classification task. Moreover, the performance of the proposed filter generation approach was proved by testing on various histopathological benchmark datasets, increasing average accuracy rates. Experimental results validate that the proposed statistical filters enhance the performance of the network with more simple CNN models.

KEYWORDS

artificial intelligence, CNN, deep learning, feature extraction, image classification, statistical filter

1 | INTRODUCTION

Deep learning models have been successfully employed a wide range of applications in variety of fields (Chen et al., 2023; Dayana et al., 2023; Qiu et al., 2023; Uyar, Solmaz, et al., 2022). Moreover, CNNs are a more sophisticated kind of artificial neural network. Weights between layers in artificial neural networks are initially generated randomly, and training of the network is provided by updating weights. In a similar manner, convolution filters of CNN are initially initialized at random, and training of the model is then performed by updating filter components. Given lots of studies, however, it remains unclear how convolution filters are initialized. Pre-trained models include pre-trained filters, but statistical filters can be used to initialize the convolution filters.

There is some research in the literature that analyse the filters that are utilized in the convolution layers of CNNs. Tang et al. (2018) designed MFNet and developed a novel detection approach based on median filtering. At the preprocessing step, the nearest neighbour interpolation method was used to increase the number of small-size images. The nonlinear classification ability of the method was enhanced with mlpconv layers and convolution layers derived from the feature maps. They expressed that the proposed method obtained significant improvement in detection performance. Özbulak and Ekenel (2018) constructed a sample CNN model and in the first convolution layer Gabor-based filters were applied. The performance of the designed model was tested using CIFAR10, CIFAR100 and MNIST datasets. The initialization of the first layer of a CNN model with Gabor-based convolution filters eliminated transfer learning and the use of the pre-trained networks. The computation performance optimization method developed by Liu et al. (2019) was presented to decrease the computational complexity in CNN by removing the unnecessary convolution filters. Due to pruning redundant convolution kernels, they obtained CNN models that are more feasible on embedded systems. Zadeh et al. (2019) aimed at a DL-based framework using Gabor filters that provide edge detection and feature extraction for the task of human emotion recognition. They supported the idea that the proposed subfeatures extracted by Gabor filters increased the speed and accuracy of training the neural network. He et al. (2019) aimed at a pedestrian flow statistics algorithm to detect pedestrian targets using real-scene videos and public datasets. In their study, the Kalman filter was used. They commented that the algorithm has good accuracy and high application value in real-time applications. Xie et al. (2019) suggested a filter-in-filter method to decrease the number of parameters and the computational cost. They described a filter's subfilters and identified numerous patterns with varying scales and orientations. They verified the efficiency of the proposed approach on Tiny ImageNet, CIFAR100 and ImageNet image classification benchmark datasets. A Gabor-based filter initialization method was presented by Molaei and Abadi (2020) to capture the variation in image content dependent on the parameters. They demonstrated how well their suggested strategy performed on classification and segmentation tasks. They came to the conclusion that the Gabor filter bank initialized network was better able to recover from noise and learn the input more precisely. Ma et al. (2020) presented a novel convolutional model named PCFNet and the filters named Sobel, Gabor and Schmid were used as predefined convolutional filters. The kernels in the first convolution layer were swapped out for kernels produced using image filters. Pan et al. (2020) proposed DropFilterR regularization method, and classification was improved by removing several filter components, which diminishes the number of learnable parameters and relaxes the weight-sharing rule of CNNs. Jeong et al. (2021) proposed a novel method for filter synthesis utilizing linear combinations of just a few basic filters that are given as input characteristics. They investigated whether filters with a limited number of input basis filters might be expressed by linear combinations and compressed the model to decrease learnable parameters. Uyar, Taşdemir, et al. (2022) searched for a novel filter initialization approach using reference images from different perspectives for the classification of small datasets. These three-dimensional reference images identified patterns at different scales and orientations. Finally, filters that extract image features more effectively provided better classification results of CNN models. Verma et al. (2022) suggested a unique Eigenvalue-based Framework (EVF) to recognize and eliminate unnecessary convolution filters to reduce the computational cost. Filter generation approach reduces the weights and biases of convolution filters, resulting in a decrease in the number of fully connected layer inputs. Experimenting with AlexNet and Vgg-16 models, they found that reduced versions of the VGG-16 and AlexNet models were comparable. Wang et al. (2023) proposed a novel residual Gabor convolutional network to recognize finger veins. Good direction and scaleability of the Gabor convolution layer effectively improved the features that describe vein patterns.

As seen in the literature, although considerable research has been devoted to convolution filters, rather less attention has been paid to the statistical side of filter generation. It is thought that the feature extraction success of CNN models can be increased if statistical filters whose generalization ability increases according to the data presented to them to initiate convolution filters instead of pre-trained filters. This paper recommends statistical filter-based CNN for image classification, and the following are the main contributions of this paper:

- Thanks to the statistical-based filter generation approach used in this study to initialize convolution kernels of CNN, a novel perspective and contribution to the field are presented.
- Initial values of the filters in the convolution layers were determined based on statistical distributions with various coefficients of variation.
- Feature extraction has been made more consistent and goal-oriented with statistical filters. The features produced were more successful in presenting the data.
- Performance tests of the proposed statistical filters were performed in tissue analysis. There are studies on tissue categorization and analysis for specific tissues; however, to our knowledge, there are no studies on general tissue analysis.
- The success of the proposed method has been tested on an originally produced texture dataset and integrated into a CNN architecture, also named HistStatCNN. At the same time, its usefulness has been demonstrated on different datasets selected from the literature.

This paper is set up as follows: The study's materials and methods are presented in Section 2. The details of the suggested strategy are described in Section 3. Section 4 presents the experimental findings and discussions. The key findings of this study are finally summarized in Section 5.

2 | MATERIALS AND METHODS

2.1 | Image classification

Image classification is one of the most popular and well-studied applications, and the primary goal is to automatically categorize images into predefined classes or categories. CNN-based image classification has achieved remarkable success across various domains, including object recognition, medical image analysis and natural language processing, making it a crucial component in many real-world applications.

2.2 | Pre-trained CNN architectures

The adaption of a previously trained model to a different design problem using its knowledge can be explained as transfer learning, and this approach can be used in case of insufficient data. However, looking at pre-trained architectures, the typical trend is designing deeper architectures (Gu et al., 2018). In this study, GoogleNet, ResNet18, ResNet50 and ResNet101 pre-trained deep models and a simple CNN model were evaluated. The number of layers and the number of learnable parameters of the HistStatCNN and pre-trained CNN models are listed in Table 1.

There is a huge number of parameters to be learned in pre-trained CNN models because of the high number of layers, while the proposed HistStatCNN model with 18 layers has far fewer parameters. The main purpose of creating a new simple CNN model is to demonstrate that simpler CNNs with the proposed statistical filters produce satisfactory results without the need to use pre-trained CNN models.

2.3 | HistStatCNN

HistStatCNN consists of an input layer, four convolution layers (first and second convolution layers (C1, C2) with 256 filters each, third and fourth convolution layers (C3, C4) with 128 filters each, 768 in total), three max-pooling layers (MP1, MP2, MP3), four ReLu layer, three fully connected layers (FC1, FC2, FC3), dropout, softmax and classification layers. The detailed properties of each layer on the HiststatCNN model are illustrated in Figure 1-4. Moreover, the simulation of HistStatCNN with layers is shown in Figure 2. The filter values obtained using the statistical approach were assigned as initial values to convolution layer filters.

2.4 | Filters in CNNs

Convolution filters play a fundamental role in feature extraction in CNN. A convolution filter is a small matrix, typically of size ($k \times k$). The filter is applied to the input image in a sliding window manner, and at each position, element-wise multiplication is performed between the filter and the corresponding region of the input. The result is summed up to obtain a single value, which becomes a pixel in the output feature map. The process of sliding the filter across the input and computing these convolutions allows the network to capture localized patterns and features, such as edges, corners and textures in the input image. By stacking multiple filters in a CNN layer, the network learns and detects different features, leading to increasingly abstract representations in deeper layers.

2.5 | Feature extraction in CNN

Convolution filters are used by CNN to the inputs to produce feature maps, which list the presence of any features that were recognized in the input image. We visualize the intermediate feature representations across different CNN layers to show the feature extraction ability of statistical filters. Figure 3 illustrates some feature maps after applying statistical filters to the HistStatCNN model.

2.6 | Datasets

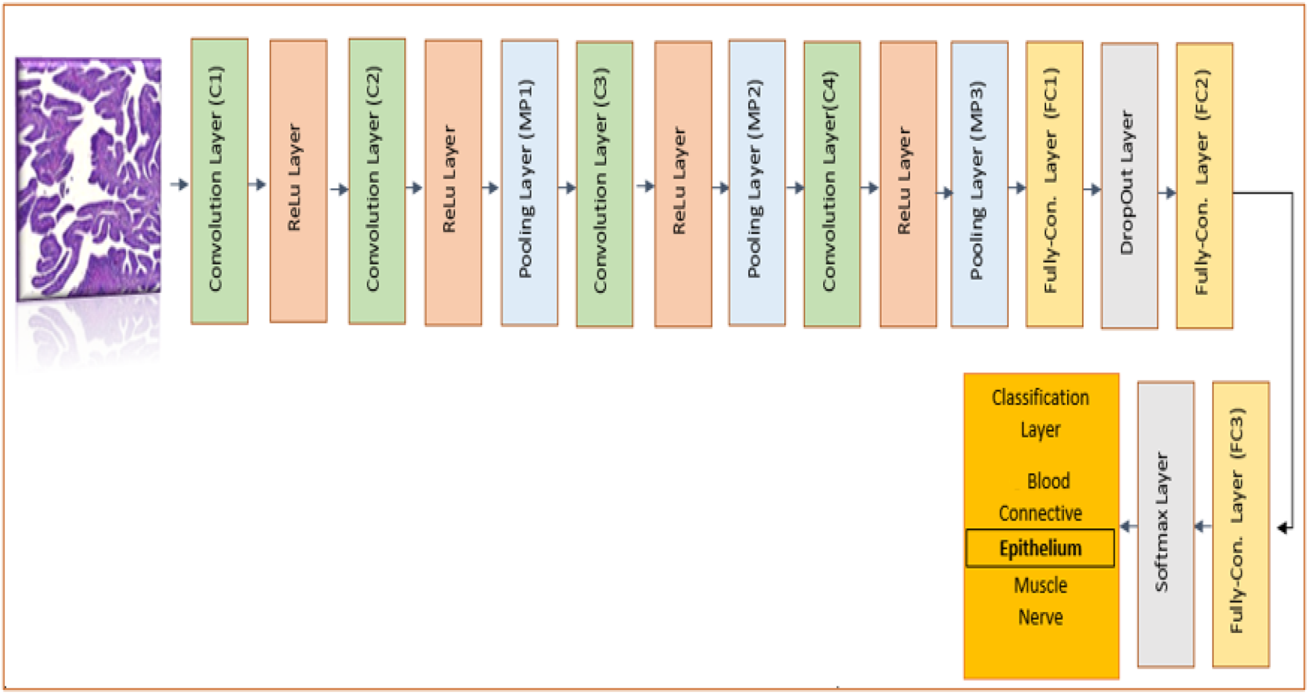
The performance of the proposed approach was tested on various medical image datasets that contain both histological and histopathological tissues. The details of the datasets can be explained as follows:

Histology dataset is a novel real-world dataset that includes images of blood, connective, epithelium, muscle and nerve at various magnification ratios (×4, ×10, ×40 and ×100), as shown in Figure 4. The dataset includes 2030 blood, 2024 connective, 2201 epithelium, 2013 muscle tissue and 2243 nerve tissue images. For each tissue,

TABLE 1 Properties of CNN models (Mathworks).

Model	# of layers (depth)	# of learnable parameters (millions)	Input size
HistStatCNN	18	0.027	64×64
GoogleNet	22	7.0	224×224
ResNet18	18	11.7	224×224
ResNet50	50	25.6	224×224
ResNet101	101	44.6	224×224

Abbreviation: CNN, convolutional neural network.



Layer No	Name of the Layer	Properties of the Layer		
1	Input Layer	Size: 64x64x3		
2	Convolution Layer(C1)	Number of Filters: 256	Filter Size: 3x3	Stride:1x1
3	ReLu Layer	-	-	-
4	Convolution Layer(C2)	Number of Filters: 256	Filter Size: 3x3	Stride:1x1
5	ReLu Layer	-	-	-
6	Pooling Layer (MP1)	Max Pooling	Filter Size: 2x2	Stride: 2x2
7	Convolution Layer(C3)	Number of Filters: 128	Filter Size: 3x3	Stride:1x1
8	ReLu Layer	-	-	-
9	Pooling Layer (MP2)	Max Pooling	Filter Size: 2x2	Stride: 2x2
10	Convolution Layer(C4)	Number of Filters: 128	Filter Size: 3x3	Stride:1x1
11	ReLu Layer	-	-	-
12	Pooling Layer (MP3)	Max Pooling	Filter Size: 2x2	Stride: 2x2
13	Fully-Connected Layer(FC1)	Output Size:120	-	-
14	DropOut Layer	Probability: 0.3	-	-
15	Fully-Connected Layer(FC2)	Output Size: 64	-	-
16	Fully-Connected Layer(FC3)	Output Size: 5	-	-
17	Softmax Layer	-	-	-
18	Classification Layer	-	-	-

FIGURE 1 Detailed layer properties of HistStatCNN model.

2000 images were taken for the experimental studies. All procedures were conducted after the approval of the Ethic Committee for Medicine Faculty, Selcuk University (2019/175).

Animal diagnostics lab (ADL) (Vu et al., 2016) has provided access to a histopathology dataset that contains images of the kidney, lung and spleen from mammals with normal and inflamed, as illustrated in Figure 5. Table 2 lists the details of the datasets.

DL methods need enough data to train the designed model. Therefore, the image dataset can be expanded using data augmentation techniques that improve the performance of deep models

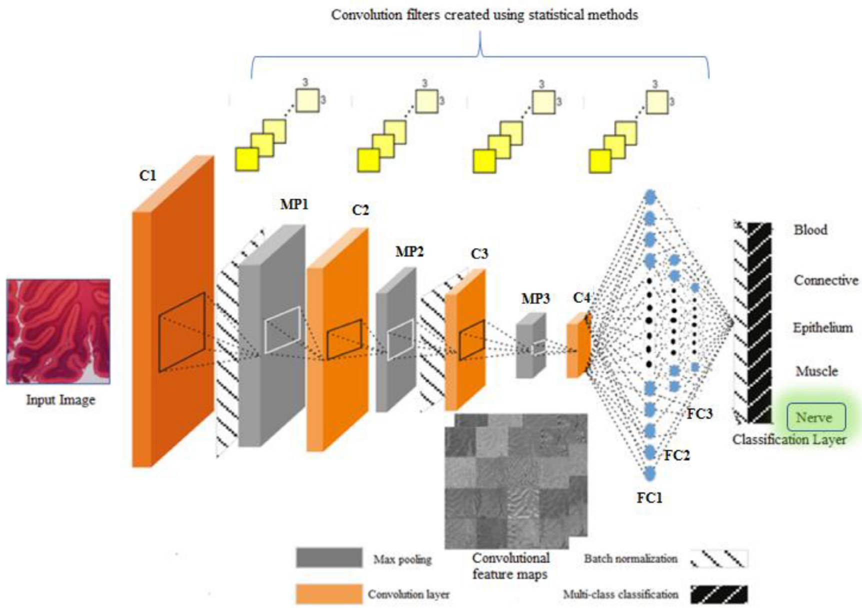


FIGURE 2 Simulation of HistStatCNN model with layer.

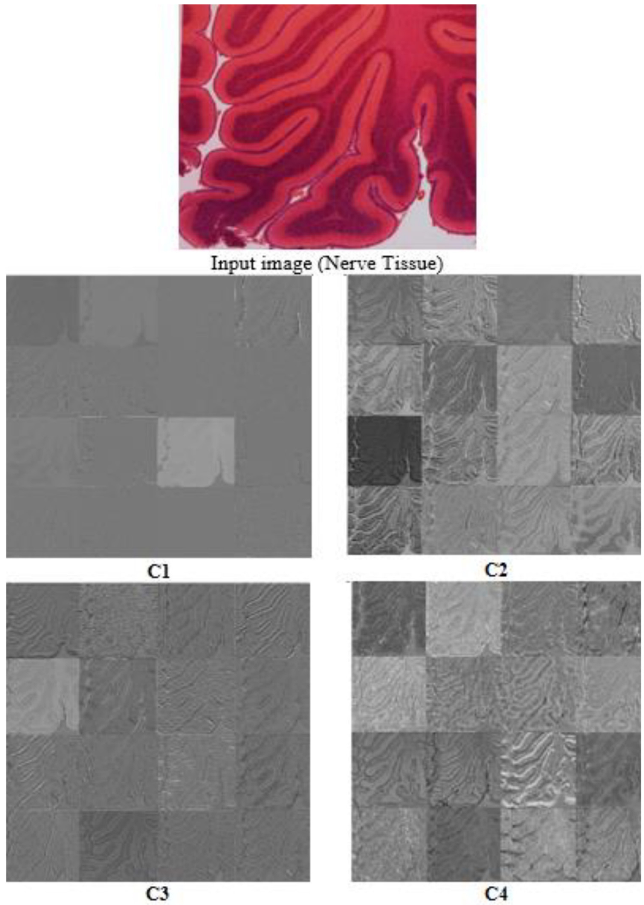


FIGURE 3 Feature extraction ability of statistical filters that are applied as convolution filters to HistStatCNN.

(Perez & Wang, 2017; Wong et al., 2016; Xu et al., 2016). In the experimental studies, reflection (vertical) and rotation in different angles (90°, 180° and 270°) were used to increase the dataset.

2.7 | Performance measurement metrics

The classification performances of models were compared depending on the measurement metrics. The mathematical definitions of these metrics are expressed in Equations (1, 2, 3, 4). In addition to these metrics, standard deviation (STD) was calculated to compare the results. Finally, the system was validated with a 10-fold cross-validation method.

$$\text{Accuracy (Acc)} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FN} + \text{FP} + \text{TN}} \quad (1)$$

$$\text{Precision (Specificity)(Pre)} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (2)$$

$$\text{Recall (Sensitivity)(Rec)} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (3)$$

$$F - \text{measure} (F - \text{m}) = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (4)$$

where FN refers to false negative, FP is false positive, TN corresponds to true negative and TP is true positive.

3 | PROPOSED METHOD

3.1 | Filter generation for convolution layers of CNN using statistical methods

Deep and complex CNN models have more learnable parameters and convolutional processes. As a result, these CNN models have a huge number of parameters and floating-point computations.

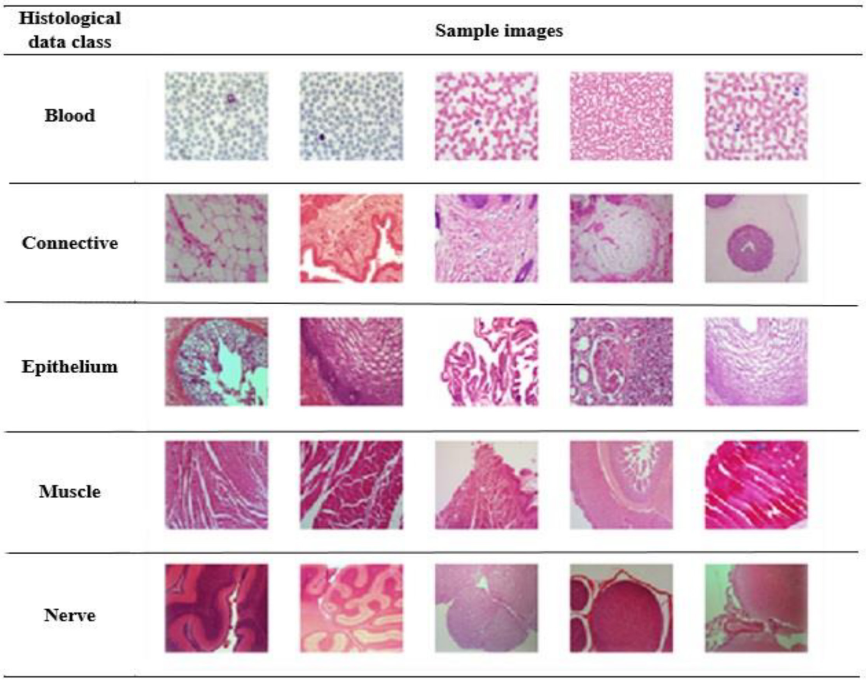


FIGURE 4 Histological image samples.

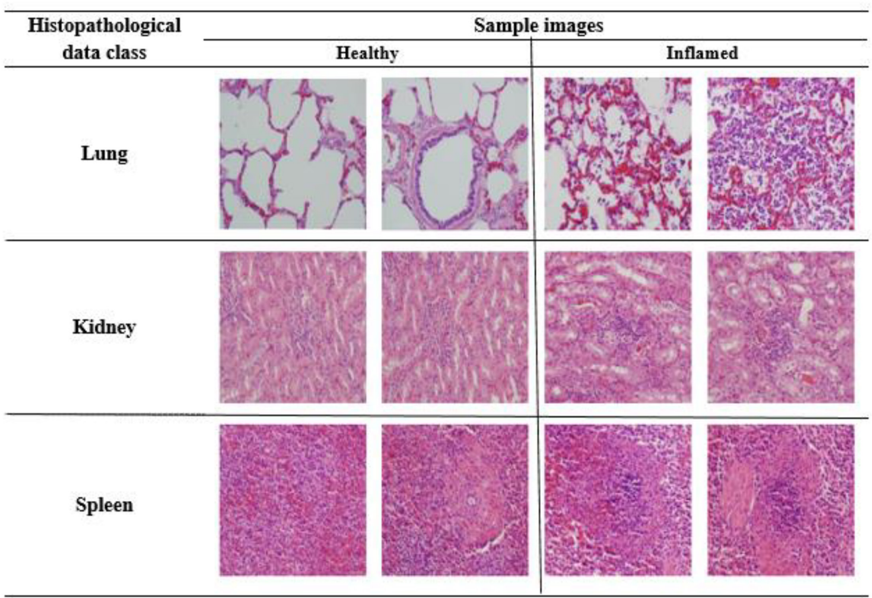


FIGURE 5 Sample images of animal diagnostics lab (ADL) dataset.

In light of this problem, continuous statistical methods for determining the initial values of the filters used in the convolution layers have been proposed to eliminate the need for a pre-trained network in the classification of small datasets. These statistical methods are Beta (McDonald & Xu, 1995), Burr (Yari & Tondpour, 2017), Chi-square (Lehoczky, 2001), Exponential (Lehoczky, 2001), *F* (Li & Martin, 2002), Gamma (Lehoczky, 2001), Generalized Extreme Value (Guida & Longo, 1988), Generalized Pareto (Hosking & Wallis, 1987), Lognormal (Tarmast, 2001), Normal (Filatov & Savyolova, 2020), Rayleigh (Al-Babtain, 2020) and Weibull (Ahmad et al., 2020). The input parameters of the statistical distributions were determined by taking into account the coefficient of variation of the distribution. The mathematical definition of the coefficient of variation, c_v is determined as indicated in Equation 5.

$$c_v = \frac{\sigma}{\mu} \tag{5}$$

where σ is the standard deviation of the distribution and μ is the mean of the distribution. The coefficient of variance is considered to form distributions in a certain standard. The input parameters of the distributions are given in Table 3. Input parameter values to generate distributions with different coefficients of variation and the steps to generate statistical filters are presented with details in Sections 3.2 and 3.3, respectively.

3.2 | Input parameters of statistical methods to obtain distributions with different coefficients of variation

A statistical approach was presented to determine the initial values of the convolution kernels. The input parameters based on the coefficient of variation of the statistical distributions are given in Table 4. *R* statistics programme (Team, 2014) was utilized to obtain input parameters of the distributions. This programme includes fields that allow entering input parameters for each distribution. When the input parameters of the distribution are entered, the coefficient of variation for that distribution is also displayed on the interface.

3.3 | Filter generation steps

The proposed statistical-based filter generation approach is described in Algorithm 1. While explaining the working principle of the algorithm, only the function content was created for the distribution with a coefficient of variation of 0.5. The working logic of the functions in the algorithm is the same for other coefficients of variation. Statistical filter generation is performed in five steps:

1. A random numerical dataset was generated using statistical distributions. A standard has been established to initialize the input parameters of the distributions. This standard is the coefficient of variation. The distribution with random input parameters creates different kinds of data. Hence, the input parameters were determined in such a way that the coefficient of variation of the distributions is the same.
2. For each distribution, input parameter values were determined considering the coefficients of variation, which are 0.5, 1.0 and 1.5. Detailed input parameter values of the distributions are given in Table 4.
3. By taking into account the relevant coefficient of variation, for each distribution, a certain amount of numbers were generated using determined input parameters. The sample size of the distribution for a specific coefficient of variation varies depending on CNN model. In our CNN model, there are four convolution layers. Seven hundred and sixty-eight different filters are needed, 256 in each first and second convolution layer and 128 in each third and fourth convolution layer. Since the size of all filters is $3 \times 3 \times 3$, a filter requires 27 values. For a distribution, 2997 random numbers were generated and 111 filters were produced. Since 12 statistical distributions are used, there are 1332 filters for a distribution with a specific coefficient of variation in the filter bank.
4. The generated numbers are not within a certain range and differ for each distribution. For example, when Beta and Chi-square distributions are created with a coefficient of variation of 0.5, the range of generated numbers for Beta is [0.0042, 0.9955], while the range of generated numbers for Chi-square is [0.5892, 27.5424] (Figure 6). The generated data were normalized in the range of [−1, 1] for each distribution because the edges and the features that define the object can be found more easily and effectively.
5. Finally, constructed statistical filters were given initial values and applied to the filters in the convolution layers of the created CNN model.

TABLE 2 ADL dataset description.

	ADL kidney	ADL lung	ADL spleen
Normal	178	155	159
Inflamed	157	153	161
Total	335	308	320

Abbreviation: ADL, animal diagnostics lab.

TABLE 3 Input parameters of the statistical methods.

Method	Input #1	Input #2	Input #3	Input #4
Beta	$\alpha 1$	$\alpha 2$	a	b
Buur	k	α	β	γ
Chi-square	υ	γ	–	–
Exponential	λ	γ	–	–
F	$\upsilon 1$	$\upsilon 2$	–	–
Gamma	α	β	γ	–
Generalized Extreme Value	k	σ	μ	–
Generalized Pareto	k	σ	μ	–
Lognormal	σ	μ	γ	–
Normal	σ	μ	–	–
Rayleigh	σ	γ	–	–
Weibull	α	β	γ	–

Note: α, k: shape parameter; a, b: boundary parameter; β, λ, σ: scale parameter; γ, μ: location parameter; υ: degree of freedom.

Algorithm 1. Statistical-based Filter Generation Algorithm

Input: Determination of statistical distributions to be used in filter production, filter size←m, C_{05}←Input parameters of statistical distributions with a coefficient of variation of 0.5, C_{10}←Input parameters of statistical distributions with a coefficient of variation of 1.0, C_{15}←Input parameters of statistical distributions with a coefficients of variation of 1.5

Output: Statistical filters with different coefficients of variation with the size of m

dataSize← The number of elements to be produced for each statistical distribution
lowerBound← lower value for the normalization of statistical filters
upperBound← upper value for the normalization of statistical filters
count← number of statistical distributions
R_{05}← GenerateRandomNumbers05(C_{05}, dataSize)
R_{10}← GenerateRandomNumbers10(C_{10}, dataSize)
R_{15}← GenerateRandomNumbers15(C_{15}, dataSize)
FB_{05}←GenerateFilterBank05(R_{05}, m, dataSize)
FB_{10}←GenerateFilterBank10(R_{10}, m, dataSize)
FB_{15}←GenerateFilterBank15(R_{15}, m, dataSize)
NF_{05}← NormalizeFilters05(FB_{05}, lowerBound, upperBound)
NF_{10}← NormalizeFilters10(FB_{10}, lowerBound, upperBound)
NF_{15}← NormalizeFilters15(FB_{15}, lowerBound, upperBound)
function *GenerateRandomNumbers05(C_{05}, dataSize)*
// Numbers are generated using input parameters of the statistical distributions and data size
//using R statistical program so generated random number array is returned as a vector.
return array of numbers produced depending on distribution with a coefficient of 0.5 variation
end the function
function *GenerateFilterBank05(R_{05}, m, dataSize)*
t←0; f=[ ] ←filter bank that contain filters with the dimension mxm
for i:0:dataSize
    f_t←$R_{05}(i+1 : i+$[1]multiple (m,m))
    i← i+ [1]multiple (m,m)
    t←t+1
return filter bank (f) that contains filters with mxm dimension
end the function
function *NormalizeFilters05(FB_{05}, lowerBound, upperBound)*
// each filter in the filter bank " FB_{05}" is normalized between [lowerBound, upperBound]
return normalized filter bank
end the function
[1] multiple: multiplication of two elements

4 | EXPERIMENTAL RESULTS

This section compares how well HistStatCNN model and existing CNN models performed on classification tasks using benchmark datasets from both histological and histopathological sources.

Training parameters were determined to evaluate the performances of pre-trained CNN models and the proposed HistStatCNN model. After 9 epochs, corresponding to 3159 iterations, the training was terminated. Additional training settings are listed in Table 5 as follows: minibatch size = 128, drop factor = 0.2, drop period = 5

TABLE 4 Input parameters of the methods to obtain distributions with the coefficient of variation of 0.5, 1.0 and 1.5.

Coefficient of variation	Method	Input #1	Input #2	Input #3	Input #4
0.5	Beta	1.5	1.5	0	1
	Buur	3	2	0.52	0.1
	Chi-square	7	0.35	–	–
	Exponential	0.39	2.5	–	–
	F	70	13	–	–
	Gamma	2.75	2	1.1	–
	Generalized Extreme Value	0.1	0.7	1.58	–
	Generalized Pareto	0.09	2	2.6	–
	Lognormal	0.48	1	0	–
	Normal	1	2	–	–
	Rayleigh	2	0.1	–	–
	Weibull	2.1	2	0	–
1.0	Beta	0.7	4.1	0	1
	Buur	1.5	1.995	1.4	0
	Chi-square	1	0.35	–	–
	Exponential	1	0	–	–
	F	5	8	–	–
	Gamma	1	1.8	0	–
	Generalized Extreme Value	0.38	0.7	1.58	–
	Generalized Pareto	0.2	0.9	0.325	–
	Lognormal	0.834	1	0	–
	Normal	1	1	–	–
	Rayleigh	1	−0.6	–	–
	Weibull	1	1	0	–
1.5	Beta	0.271	1.99	0	1
	Buur	1.21	1.977	1	0
	Chi-square	1	−0.0575	–	–
	Exponential	1	−0.3335	–	–
	F	24	5	–	–
	Gamma	0.34	2.07	0.1	–
	Generalized Extreme Value	0.1	1	0.308	–
	Generalized Pareto	0.278	1	0	–
	Lognormal	1.087	1	0	–
	Normal	1.5	1	–	–
	Rayleigh	1	−0.817	–	–
	Weibull	0.684	1	0	–

and initial learning rate = 0.01. The network was optimized using the Stochastic Gradient Descent with Momentum (SGDM).

4.1 | Results on pre-trained CNN models

The classification success of pre-trained CNN models that contain residual blocks, which are ResNet18, ResNet50 and ResNet101 and inception modules, which is GoogleNet have been evaluated. The minibatch accuracy and loss graph during training are illustrated in Figure 7, and performance metrics of the pre-trained CNN models are given in Table 6.

Among all models, the highest accuracy, as listed in Table 6, 85.56%, was obtained with GoogleNet. However, these pre-trained CNN models cause computational complexity and memory usage because of their deepness. The proposed CNN model is a more simple CNN model, and values of the convolution filters were initialized using statistical distributions.

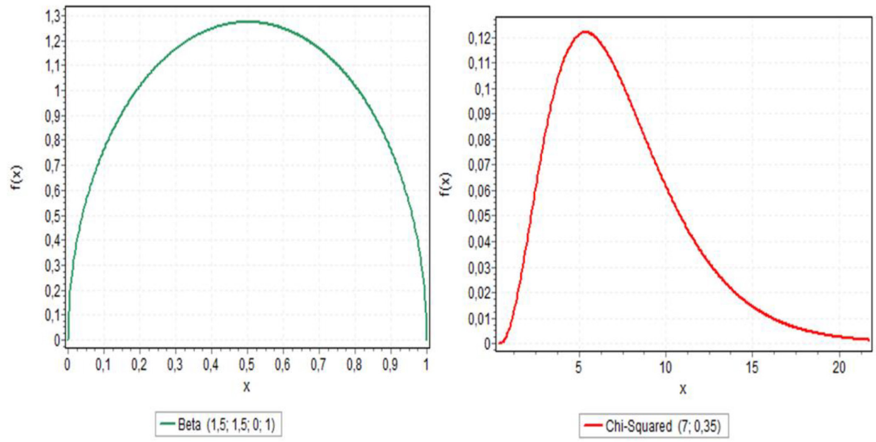


FIGURE 6 Beta and Chi-square probability density function graphics with the coefficient of variation 05.

TABLE 5 Parameters for training of CNN models.

Parameter	Drop factor	Drop period	Execution environment	Initial learning rate	Maximum epoch	Minibatch size	Optimizer
Value	0.2	5	Multi-GPU	0.01	9	128	SGDM

Abbreviations: GPU, graphics processing unit; SGDM, Stochastic Gradient Descent with Momentum.

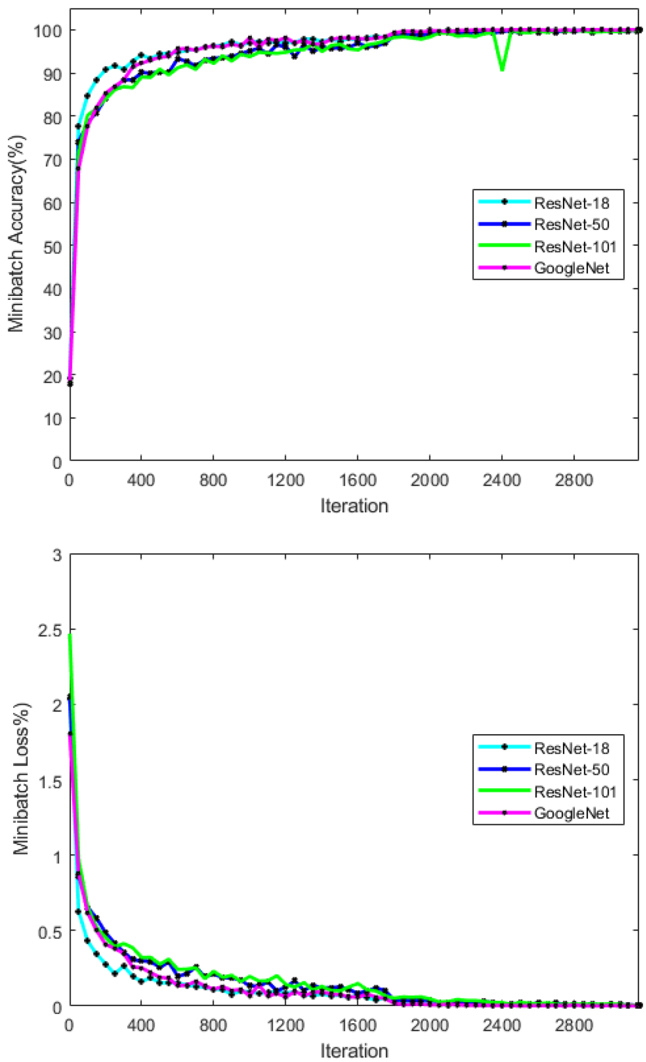


FIGURE 7 Minibatch accuracy and loss graph of convolutional neural network (CNN) models during training.

TABLE 6 Test results of the pre-trained CNN models for the classification task.

	Measurement metrics(%)				
CNN model	Top 1 Acc	Acc	Pre	Rec	F-m
GoogleNet	96.74	85.56	87.57	85.56	86.54
ResNet18	97.36	85.24	86.87	85.24	86.03
ResNet50	95.52	83.59	85.22	83.59	84.37
ResNet101	92.86	83.79	85.33	83.79	84.54

Abbreviations: Acc, accuracy; CNN, convolutional neural network; *F*-m, *F*-measure; Pre, precision; Rec, recall.

4.2 | HistStatCNN results on generated statistical convolution filters

The filters in the filter bank generated from statistical methods with different coefficients of variation were used as convolution layer filters.

4.2.1 | Experiment #1: Analysis of HistStatCNN classification performance employing filters with 0.5 coefficient of variation

First, filters with a 0.5 coefficient of variation were assigned to all convolution layer filters of HistStatCNN. Table 7 gives the test result using statistical filters with a 0.5 coefficient of variation to all convolution layers on HistStatCNN.

Average precision, recall and *F*-measure values at the end of the experiments were 86.86%, 84.85% and 85.58%, respectively. The classification accuracy was achieved at 84.85%, and a close accuracy rate compared to the best CNN model was achieved using filters with a 0.5 coefficient of variation.

TABLE 7 Test result of the filters with 0.5 coefficient of variation applied to HistStatCNN model.

	Measurement metrics							
Model	Top 1 Acc	Acc	Pre	Rec	*F*-m	STD of Pre	STD of Rec	STD of *F*-m
Experiment#1	0.9470	**0.8485**	0.8686	0.8485	0.8558	0.0737	0.0925	0.0835

Abbreviations: Acc, accuracy; *F*-m, *F*-measure; Pre, precision; Rec, recall; STD of *F*-m, standard deviation of *F*-measure; STD of Pre, standard deviation of precision; STD of Rec, standard deviation of recall.

TABLE 8 Test result of the filters with 1.0 coefficient of variation applied to HistStatCNN model.

	Measurement metrics							
Model	Top 1 Acc	Acc	Pre	Rec	*F*-m	STD of Pre	STD of Rec	STD of *F*-m
Experiment #2	0.9504	**0.8483**	0.8599	0.8483	0.8540	0.0876	0.0991	0.0935

Abbreviations: Acc, accuracy; *F*-m, *F*-measure; Pre, precision; Rec, recall; STD of *F*-m, standard deviation of *F*-measure; STD of Pre, standard deviation of precision; STD of Rec, standard deviation of recall.

TABLE 9 Test result of the filters with 1.5 coefficient of variation applied to HistStatCNN model.

	Measurement metrics							
Model	Top 1 Acc	Acc	Pre	Rec	*F*-m	STD of pre	STD of Rec	STD of *F*-m
Experiment #3	0.9546	**0.8572**	0.8682	0.8572	0.8625	0.0799	0.0910	0.0855

Abbreviations: Acc, accuracy; *F*-m, *F*-measure; Pre, precision; Rec, recall; STD of *F*-m, standard deviation of *F*-measure; STD of Pre, standard deviation of precision; STD of Rec, standard deviation of recall.

4.2.2 | Experiment #2: Analysis of HistStatCNN classification performance employing filters with a 1.0 coefficient of variation

After Experiment #1 experiment, filters with a coefficient of variation of 1.0 were assigned to all convolution layer filters of HistStatCNN. Table 8 shows the test result using statistical filters with a 1.0 coefficient of variation for all convolution layers on HistStatCNN.

Average precision, recall and *F*-measure values at the end of the experiments were 85.99%, 84.83% and 85.40%, respectively. As listed in Table 8, the achieved classification accuracy is 84.83%. Compared to Experiment #1, very close classification accuracy was achieved using filters with a 1.0 coefficient of variation.

4.2.3 | Experiment #3: Analysis of HistStatCNN classification performance employing filters with a 1.5 coefficient of variation

Filters with a coefficient of variation of 1.5 were assigned to all convolution layer filters of HistStatCNN. Table 9 shows the test results using statistical filters with a 1.5 coefficient of variation for all convolution layers on HistStatCNN. Figure 8 demonstrates minibatch accuracy and loss graphs of HistStatCNN with different statistical filters during training.

At the conclusion of the experiment, the average precision, recall and F-measure were 86.82%, 85.72% and 86.25%, respectively. As stated in Table 9, the achieved classification accuracy is 85.72%.

Filters with a coefficient of variation of 1.5 (Table 9) yield more successful results than the other two filters (Tables 7 and 8). To increase the classification success, the idea of using filters with different variation coefficients in the convolution layers of HistStatCNN has emerged and a mixed parameters set of filters was applied to HistStatCNN.

4.2.4 | Experiment #4: Analysis of HistStatCNN classification performance employing filters with the mixture of the 0.5, 1.0 and 1.5 coefficients of variation

When we observe test results from Experiment #1 to Experiment #3, the performance metrics are very close to each other. To extract basic visual elements like oriented edges or corners, local receptive fields are used. To find more complicated higher-order characteristics, the extracted features are then integrated with the subsequent layers (Mrazova & Kukacka, 2012). Considering the working logic of CNNs, filter values with a high coefficient of variation have been assigned to the convolution layers in the inner layers. Because of that, the mixture of the filters with the coefficient of variation of 0.5 to the first convolution layer, 1.0 to the second and third convolution layer and

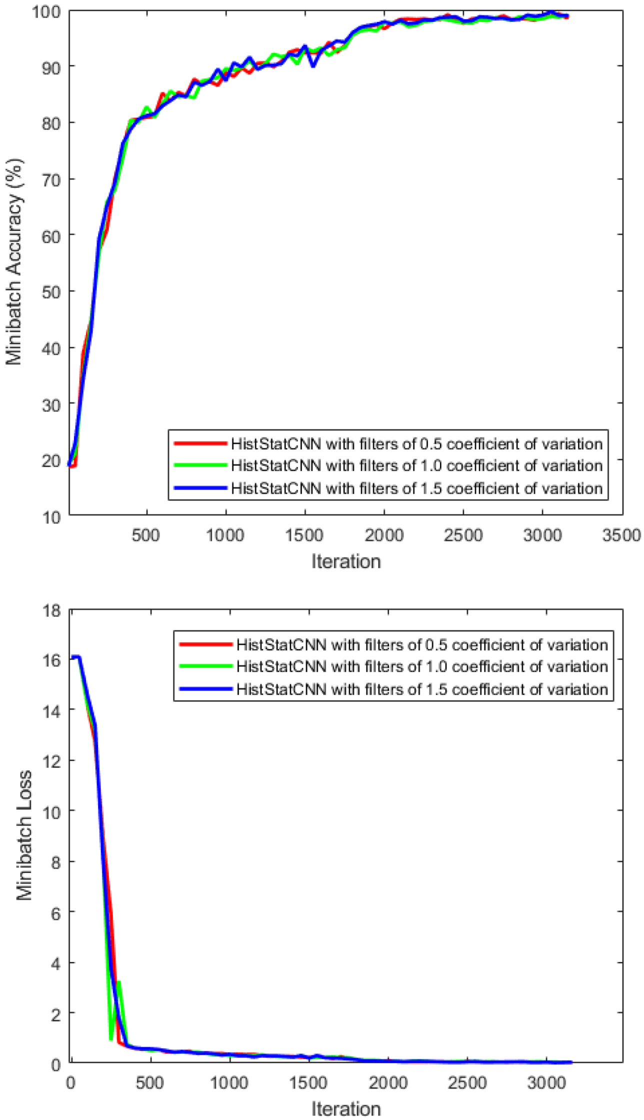


FIGURE 8 Minibatch accuracy and loss graphs of HistStatCNN with three unique parameters set during training.

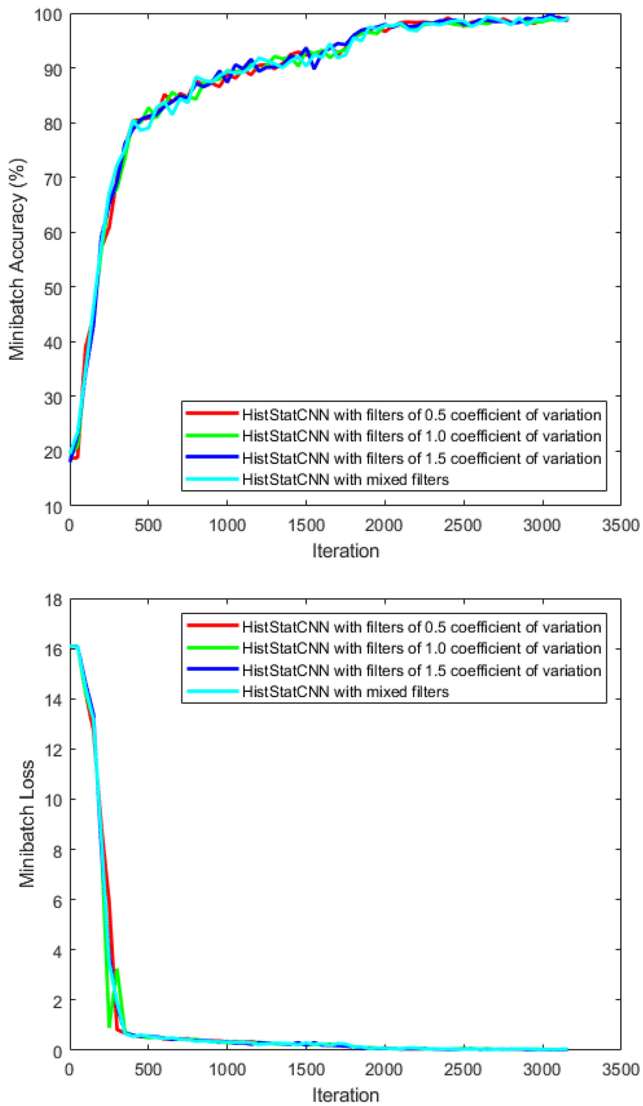


FIGURE 9 Minibatch accuracy and loss graphs of HistStatCNN with four statistical filter cases during training.

1.5 to the last convolution layer was assigned for better classification performance. Figure 9 illustrates minibatch accuracy and loss graphs of HistStatCNN with four statistical filter cases during training.

Table 10 shows the test results using statistical filters with different coefficients of variation for all convolution layers. At the conclusion of the studies, average values for precision, recall and F-measure were 87.74%, 87.13% and 87.42%, respectively.

For a deeper analysis of the obtained results (Table 10), the confusion matrix obtained by the proposed HistStatCNN model on the histological dataset after 10-fold cross-validation is illustrated in Figure 10. The confusion matrix shows that there are few FP and FN when it comes to blood class. Blood tissue is the simplest tissue to classify in terms of its structure. While the nervous tissue is in second place with a success rate of 84.36%, the connective, epithelium and muscle tissues are the most difficult tissues to classify correctly since they are structurally very similar to each other.

The summary of test results of three unique parameter sets (Experiment #1, Experiment #2 and Experiment #3) and a mixed set (Experiment #4) of statistical filters are shown in Table 11. As can be seen from the experimental results indicated in Table 11, Experiment #4 represents better classification results with the filter order. In experimental studies carried out under the same training conditions, the statistical-based CNN model both contains fewer learnable parameters and gives more successful results in smaller input image sizes.

4.3 | Performance analysis of HistStatCNN with histopathological benchmark datasets

To illustrate the efficiency of HistStatCNN, an extra set of experiments was conducted. The effectiveness of HistStatCNN and CNN

Anatomia Histologia Embryologia — WILEY

TABLE 10 Test result of the filters with a mixture of 0.5, 1.0 and 1.5 coefficients of variation applied to HistStatCNN model.

	Measurement metrics							
Model	Top 1 Acc	Acc	Pre	Rec	*F*-m	STD of pre	STD of Rec	STD of *F*-m
Experiment #4	0.9610	**0.8713**	0.8774	0.8713	0.8742	0.0557	0.0619	0.0587

Abbreviations: Acc, accuracy; *F*-m, *F*-measure; Pre, precision; Rec, recall; STD of *F*-m, standard deviation of *F*-measure; STD of Pre, standard deviation of precision; STD of Rec, standard deviation of recall.

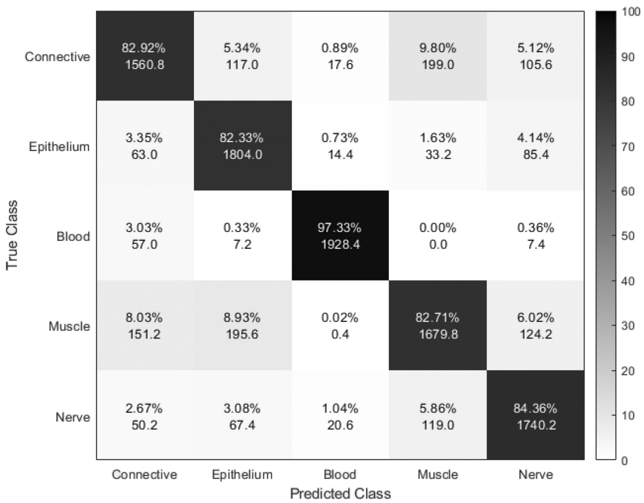


FIGURE 10 Confusion matrix obtained by the proposed HistStatCNN model on the histological dataset.

was evaluated on three different histopathological benchmark datasets. The network was trained using the same training parameters as mentioned in Table 5. Data augmentation was also utilized to reduce overfitting. The results of the experiments using benchmark datasets are given in Table 12.

As a result of classification of histopathological data task, it is revealed that the average improvement in accuracy, precision, recall and F-measure is 3.03%, 2.69%, 3.06% and 2.88%, respectively. It is obvious that the proposed approach also gives successful classification results on different histopathological benchmark datasets.

4.4 | Discussion

To demonstrate the performance of the statistical-based filter generation approach, five different experiment sets were designed. In the first experiment set, some pre-trained CNN models were used, and the performance increase was measured. In the second, third, fourth and fifth experiment sets, three different parameter sets and mixed parameter set, respectively, were adapted to the HistStatCNN model. In the last experiment set, the statistical-based filter generation approach was analysed with some histopathological benchmark datasets.

GoogleNet enhanced the accuracy value by up to 85.56% in the first experiment set. In the second experiment set, all filters were

TABLE 11 Classification performance summary using HistStatCNN with three unique parameter sets and a mixed set of statistical filters.

Model	Accuracy (%)
Experiment #1	84.85
Experiment #2	84.83
Experiment #3	85.72
Experiment #4	**87.13**

The accuracy value of best performing experiment (Experiment) is written in bold.

obtained using distributions with 0.5 coefficients of variation, and the accuracy was 84.85%. In the third experiment set, all filters were obtained using distributions with 1.0 coefficients of variation, and the accuracy was 84.83%. In the fourth experiment set, all filters were obtained using distributions with 1.5 coefficients of variation, and the accuracy was 85.72%. In the fifth experiment set, a mixed set of parameters was used, and the accuracy was 87.13%. Finally, in the last experiment, the performance of the HistStatCNN was proved by testing on various histopathological benchmark datasets. The outcomes showed an average improvement in accuracy, precision, recall and F-measure of 3.03%, 2.69%, 3.06% and 2.88%, respectively. Various experimental studies demonstrate the effectiveness and adaptability of the statistical filter to the initialization method for the convolution filters of the designed CNN models.

In addition to these results, medical practitioners will be able to concentrate on urgent patients as their burden will be reduced, and histology data will be automatically classified by using the suggested artificial intelligence system. Moreover, for undergraduate and graduate students studying medicine and health sciences, it is a valuable teaching resource (Uyar, Solmaz, et al., 2022).

5 | CONCLUSIONS

In this paper, we proposed a novel statistical filter-based CNN to initialize convolution kernels in CNN. Distributions with different coefficients of variation were produced using 12 different continuous statistical methods. Using the values in this distribution, filters of different sizes were produced. We have applied our proposed method to the designed CNN model, which is simple with far fewer learnable parameters. Three unique parameter sets and a mixed parameter set

TABLE 12 Classificiation performances of statistical filters-based CNN and CNN.

	Statistical filters-based CNN					CNN				
	Measurement metrics(%)									
ADL Dataset	Acc	Pre	Rec	F-m	Top 1 Acc	Acc	Pre	Rec	F-m	Top 1 Acc
Kidney	**93.43**	93.54	93.48	93.51	95.82	**89.43**	89.69	89.36	89.52	94.33
Lung	**94.74**	95.05	94.74	94.89	99.35	**91.10**	92.20	91.12	91.65	98.38
Spleen	**95.75**	95.80	95.75	95.77	96.56	**94.31**	94.42	94.31	94.37	96.56

The numerical values for the accuracy comparison between the proposed and CNN methods are indicated in bold. Abbreviations: Acc, accuracy; ADL, animal diagnostics lab; CNN, convolutional neural network; *F*-m, *F*-measure; Pre, precision; Rec, recall.

of statistical filters were applied to HistStatCNN. The success of the proposed method has been tested on an originally produced texture dataset and integrated into HistStatCNN. At the same time, its usefulness has been demonstrated on different datasets selected from the literature. According to studies, the resulting reduced model outperforms pre-trained models, GoogleNet and ResNet, in terms of computational cost while preserving comparable accuracy. HistStatCNN obtains satisfactory results and reuses on lower end central processing units. The overall results indicate that statistical filters can be used as a convolution kernel initialization method in CNNs.

ACKNOWLEDGEMENTS

None.

CONFLICT OF INTEREST STATEMENT

The authors declare that they have no conflict of interest.

DATA AVAILABILITY STATEMENT

The dataset generated during the study are not publicly available due to ethical restrictions (medical data) but are available from the corresponding author on reasonable request.

ORCID

Nejat Ünlükal https://orcid.org/0000-0002-8107-4882

REFERENCES

Ahmad, A. E. L., Baset, A., & Ghazal, M. G. M. (2020). Exponentiated additive Weibull distribution. *Reliability Engineering and System Safety*, *193*, 106663.

Al-Babtain, A. A. (2020). A new extended Rayleigh distribution. *Journal of King Saud University - Science*, *32*, 2576–2581.

Chen, Z., Chen, J., Ding, G., & Huang, H. (2023). A lightweight CNN-based algorithm and implementation on embedded system for real-time face recognition. *Multimedia Systems*, *29*, 129–138.

Dayana, A., Mary, W. R., Emmanuel, S., & Harriet Linda, C. (2023). Feature fusion and optimization integrated refined deep residual network for diabetic retinopathy severity classification using fundus image. *Multimedia Systems*, *29*, 1629–1650.

Filatov, S., & Savyolova, T. (2020). Calculation of normal distributions on the sphere S2. *Procedia Computer Science*, *169*, 860–868.

Gu, J., Wang, Z., Kuen, J., Ma, L., Shahroudy, A., Shuai, B., Liu, T., Wang, X., Wang, G., Cai, J., & Chen, T. (2018). Recent advances in convolutional neural networks. *Pattern Recognition*, *77*, 354–377.

Guida, M., & Longo, M. (1988). Estimation of probability tails based on generalized extreme value distributions. *Reliability Engineering and System Safety*, *20*, 219–242.

He, M., Luo, H., Bin, H., & Chang, Z. (2019). Pedestrian flow tracking and statistics of monocular camera based on convolutional neural network and Kalman filter. *Applied Sciences*, *9*, 1624.

Hosking, J. R. M., & Wallis, J. R. (1987). Parameter and quantile estimation for the generalized Pareto distribution. *Technometrics*, *29*, 339–349.

Jeong, J., Cho, J.-H., & Lee, J.-G. (2021). Filter combination learning for CNN model compression. *ICT Express*, *7*, 5–9.

Lehoczky, J. P. (2001). Distributions, statistical: Special and continuous. In N. J. Smelser & P. B. Baltes (Eds.), *International encyclopedia of the Social & Behavioral Sciences*. Pergamon.

Li, B., & Martin, E. B. (2002). An approximation to the F distribution using the chi-square distribution. *Computational Statistics and Data Analysis*, *40*, 21–26.

Liu, C., Lin, T., Wu, Y., Lin, Y., Lee, H., Tsao, Y., & Chien, S. (2019). Computation-performance optimization of convolutional neural networks with redundant filter removal. *IEEE Transactions on Circuits and Systems I: Regular Papers*, *66*, 1908–1921.

Ma, Y., Luo, Y., & Yang, Z. (2020). PCFNet: Deep neural network with predefined convolutional filters. *Neurocomputing*, *382*, 32–39.

Mathworks Pretrained Deep Neural Networks. Retrieved July 15, 2023, from https://www.mathworks.com/help/deeplearning/ug/pretrained-convolutional-neural-networks.html

McDonald, J. B., & Xu, J. Y. (1995). A generalization of the beta distribution with applications. *Journal of Econometrics*, *66*, 133–152.

Molaei, S., & Abadi, M. E. S. A. (2020). Maintaining filter structure: A Gabor-based convolutional neural network for image analysis. *Applied Soft Computing*, *88*, 105960.

Mrazova, I., & Kukacka, M. (2012). Can deep neural networks discover meaningful pattern features? *Procedia Computer Science*, *12*, 194–199.

Özbulak, G., & Ekenel, H. K. (2018). Initialization of convolutional neural networks by Gabor filters. In *2018 26th signal processing and communications applications conference (SIU)*, 1–4.

Pan, H., Niu, X., Li, R., Shen, S., & Dou, Y. (2020). DropFilterR: A novel regularization method for learning convolutional neural networks. *Neural Processing Letters*, *51*, 1285–1298.

Perez, L., & Wang, J. (2017). The effectiveness of data augmentation in image classification using deep learning. *arXiv Preprint arXiv*:1712.04621.

Qiu, C., Song, Y., Liu, Z., Yin, J., Han, K., & Liu, Y. (2023). CMFCUNet: Cascaded multi-scale feature calibration UNet for pancreas segmentation. *Multimedia Systems*, *29*, 871–886.

R Core Team. (2014). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing.

Tang, H., Ni, R., Zhao, Y., & Li, X. (2018). Median filtering detection of small-size image based on CNN. *Journal of Visual Communication and Image Representation*, *51*, 162–168.

Tarmast, G. (2001). Multivariate log-normal distribution. international statistical institute: Seoul 53rd session.

Uyar, K., Solmaz, M., Tasdemır, S., & Ünlükal, N. (2022). A novel histological dataset and machine learning applications. *Turkish Journal of Science and Technology*, *17*, 185–196.

Uyar, K., Taşdemir, Ş., Ülker, E., Ünlükal, N., & Solmaz, M. (2022). Improving efficiency in convolutional neural networks with 3D image filters. *Biomedical Signal Processing and Control*, *74*, 103563.

Verma, A., Meenpal, T., & Acharya, B. (2022). Computational cost reduction of convolution neural networks by insignificant filter removal. *Romanian Journal of Information Science and Technology*, *25*, 150–165.

Vu, T. H., Mousavi, H. S., Monga, V., Rao, G., & Rao, U. K. A. (2016). Histopathological image classification using discriminative feature-oriented dictionary learning. *IEEE Transactions on Medical Imaging*, *35*, 738–751.

Wang, Y., Huimin, L., Qin, X., & Guo, J. (2023). Residual Gabor convolutional network and FV-mix exponential level data augmentation strategy for finger vein recognition. *Expert Systems with Applications*, *223*, 119874.

Wong, S. C., Gatt, A., Stamatescu, V., & McDonnell, M. D. (2016). Understanding data augmentation for classification: When to warp? In *2016 International Conference on Digital Image Computing: Techniques and applications (DICTA)* (pp. 1–6). IEEE.

Xie, G., Yang, K., & Lai, J. (2019). Filter-in-filter: Low cost CNN improvement by sub-filter parameter sharing. *Pattern Recognition*, *91*, 391–403.

Xu, Y., Jia, R., Mou, L., Li, G., Chen, Y., Lu, Y., & Jin, Z. (2016). Improved relation classification by deep recurrent neural networks with data augmentation. *arXiv Preprint* arXiv:1601.03651.

Yari, G., & Tondpour, Z. (2017). The new Burr distribution and its application. *Mathematical Sciences*, *11*, 47–54.

Zadeh, M., Taghi, M., Imani, M., & Majidi, B. (2019). Fast facial emotion recognition using convolutional neural networks and gabor filters. In *2019 5th Conference on Knowledge Based Engineering and Innovation (KBEI)*, 577–581.

How to cite this article: Ünlükal, N., Ülker, E., Solmaz, M., Uyar, K., & Taşdemir, Ş. (2024). Histological tissue classification with a novel statistical filter-based convolutional neural network. *Anatomia, Histologia, Embryologia*, *53*, e13073. https://doi.org/10.1111/ahe.13073