IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received April 6, 2020, accepted April 21, 2020, date of publication April 27, 2020, date of current version May 12, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2990700

Multi-Lane Capsule Network for Classifying Images With Complex Background

SIWEI CHANG AND JIN LIU, (Member, IEEE)
College of Information Engineering, Shanghai Maritime University, Shanghai 201306, China
Corresponding author: Jin Liu (jinliu@shmtu.edu.cn)

This work was supported by the National Natural Science Foundation of China under Grant 61872231 and Grant 61701297.

ABSTRACT Capsule Network (CapsNet) is a novel structure for deep neural network, mapping the region of target instance to vectors and matrices rather than scalars. This process is enabled by dynamic routing algorithm which help CapsNet to achieve more robust capacity with fewer parameters than traditional CNNs. However, one drawback of Capsule is that it turns to account for everything in the image, which leads to a poor performance when the backgrounds are too varied to model with a reasonable sized net. We proposed a multi-lane capsule network with strict-squash (MLSCN) to solve this problem. In MLSCN, we designed a novel Capsule based network structure, replaced the Squash function and optimized the implementation of dropout. To validate modification proposed in the paper, we conducted extensive experiments on three public datasets which include MNIST, affNIST and CIFAR10. Ablation experiments were also conducted to analyze contributions of each component of MLSCN. Experimental results show that MLSCN outperforms the original CapsNet in multiple benchmarks. Our model boosts the classification accuracy of CIFAR10 about 17% with negligible parameter increase compare to original CapsNet. Besides, the proposed model achieves an accuracy of 65.37%, while original CapsNet is only 41.62% on MNIST-BC.

INDEX TERMS Capsule network, multi-lane, strict-squash, drop-circuit, lane-filter, complex background.

I. INTRODUCTION

During the last decade, deep learning has achieved remarkable progress on various tasks [1], [2]. Convolutional neural networks [3]–[5] in particular, have been firmly established as state-of-the-art approaches in computer vision tasks such as classification, object detection and instance segmentation. One of the basic principles of CNN is 'induction bias', which means that if a local feature is useful in one image location, the same feature is likely to be useful in other locations. This in turn leads to a series of characters called translation invariance, rotation invariance and scale invariance. Invariance is achieved through local receptive fields, weight sharing and pooling operations since the advent of CNNs. Numerous efforts have since continued to push the boundaries of convolutional neural networks [6]–[9]. Although CNN achieved some invariance, and did benefit from it, it lost a lot of information, such as the spatial relationship between features [10]–[12]. Considering the large number of parameters in CNN, pooling layer is essential. Therefore, discarding information is inevitable. To alleviate this problem, we have to design complex structures specifically for different tasks and datasets.

However, such solutions are only palliatives. Most of these targeted models lack generalization ability and can only mitigate the effects of information loss to a certain extent. Capsule Network is a novel structure proposed by Sabour [13], [14] which discards pooling operations and uses a dynamic routing algorithm instead. It achieved the same accuracy compared to traditional CNN networks with dynamic routing technique, and even better performance in the tasks with fewer data. Besides, the inference process can be more interpretable through reconstruction operations. Through the extraction of low-level features and clusters by this dynamic routing algorithm, CapsNet obtains equivariance rather than invariance. Equivariance ensures that capsule network has the ability to identify features that are slightly deformed and have the capability of getting the spatial relationship between different target instances.

However, there are many obstacles on the road which prevent CapsNet from being applied on large-scales application. The major deficiencies are the requirements of massive, contiguous memory when dealing with complex tasks as well as the incapacity to classify images in a complex background.

The associate editor coordinating the review of this manuscript and approving it for publication was Hiram Ponce.

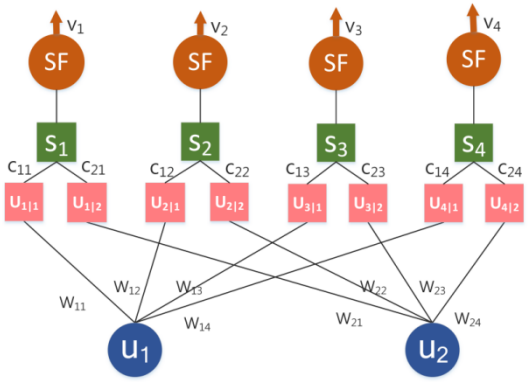

FIGURE 1. The process of dynamic routing algorithm.

In this article, the main contributions of this paper are as follows:

1) A novel strict-squash algorithm is proposed to restrain the sensitivity of the dynamic routing algorithm and limit the impact of useless features on capsule initialization.
2) To focus on those lanes which extract powerful features in the multi-lane module and augment their impact on the final probabilities, we design an efficient dynamic weighting assignment strategy.
3) By combining these two methods, we propose a novel model, MLSCN, on the basis of MLCN [31]. MLSCN achieves optimized parallelism based on adaptive lane selection, significantly improving the classification accuracy for images with complex background.

The rest of the paper is organized as follows. Section 2 introduces related works. In Section 3, we illuminate our methods in detail. Section 4 shows the results of experiments. Finally, Section 5 concludes the paper.

II. RELATED WORK

Before deep learning, machine learning was the mainstream method when addressing the issue of computer vision and natural language processing. However, machine learning is based on feature engineering which require extensive professional research in corresponding field. In 1994, Yan LeCun introduced LeNet [15] for handwritten character recognition and classification and achieved 98% accuracy. Based on LeNet, Alex Krizhevsky proposed AlexNet [16] in 2012 and be the state-of-the-art on the ImageNet dataset. Oxford Visual Geometry Group proposed VGGNet [17] which performed well in both classification tasks and positioning tasks in 2014. Kaiming He presented ResNet [18] in 2015 which makes it possible to train a deep neural network and avoid the accuracy of saturation. Some other models such as R-CNN [19], Faster R-CNN [20] which dominate target detection field, Mask-RCNN [21] the most powerful model in the segmentation, UNet [22] and FCN [23] are also very popular models. Although CNNs can already address most of issues in computer vision, CNNs have two major drawbacks. Firstly, due to the sub-sampling using in pooling steps to transfer more important features to the next layer, some viewpoint changes and spatial relationships between higher level components will be lost. Secondly, CNNs are not robust against new viewpoints because they cannot extrapolate their geometric information.

These two problems hampered the further development of CNNs, and researchers had to start looking for novel network structures. Hinton, Krizhevsky and Sida presented the design of capsules network in 2011. Later publication of the dynamic routing algorithm, which allows capsules in the low-level cluster onto high-level to obtain the characteristics of objects while retaining spatial information. Capsule network could achieve a good generalization using relatively fewer parameters than CNNs (only 8.2M parameters for the MNIST model with reconstruction).

Since the publication of the dynamic routing and the capsule network, several works have emerged. Some of them improve the algorithm or modify the architecture and the others used CapsNet in other applications. Shahroudnejad *et al.* [24] illustrated potential intrinsic explainability properties of Capsule network, by analyzing its behavior and structure. Pöpperl *et al.* [25] show how to apply capsule network to ultrasonic data from automotive ultrasonic sensors to classify object height. Jaiswal *et al.* [26] presented guidelines for designing CapsuleGANs as well as updated objective function for training CapsuleGANs and outperform convolutional-GANs on the generative adversarial metric and at semi-supervised classification with a large number of unlabeled generated images and a small number of real labeled ones, on MNIST and CIFAR-10 datasets. Jimenez-Sanchez *et al.* [27] tested the capsule network in Medical Imaging Data Challenges and found that it achieved remarkable performance even with fewer trainable parameters than the tested counterpart CNNs. Mobiny *et al.* [28] tested capsule network's performance in lung cancer screening and showed that CapsNet can outperform CNN especially when the training set is small. Mukhometzianov *et al.* [29] ran the capsule network with multiple image datasets and they showed with outstanding results, capsule network still needs much more time to train than other CNNs. Wang *et al.* [30] prove that the routing mechanism proposed in Sabour *et al.* [13] is similar to minimize a standard clustering loss with a KL regularization on the coupling probabilities and discuss few possible ways to improve the performance of capsule networks.

Most of the work is interpretative or just port the capsule network and insert it into an existing backbone network as a block. They do not solve the key issue: CapsNet is highly susceptive to noise. CapsNet tends to account for all the elements in the image which is attribute to the dynamic routing algorithm. Besides, it is hard to maintain equivalence while suppress this susceptibility in single-line capsule network. Therefore, in this paper we propose to use multi-lane structure and propose a novel squash function to solve these problems.

III. METHOD

Image classification is a key step in several computer visual tasks. Capsule network, as an improvement on convolutional neural networks, uses a dynamic routing algorithm and discards the pooling operation. These modifications enable CapsNet address issues that CNNs cannot solve, such as instance occlusion and multi-view recognition. Although this emerging model is now efficient enough to qualify complete most of tasks, it is unable to classify images which have complex backgrounds. which can be formalized as follows. The input of this issue can be formalized as follows:

$$G_{in}^{i\times j} = \left(\left(g_{11},\ldots,g_{1j}\right),\left(g_{21},\ldots,g_{2j}\right),\ldots,\left(g_{i1},\ldots,g_{ij}\right)\right)$$

where g_{ij} is the value of pixel in the location (i, j) of input. After processed by convolutional networks, we can get a feature map F:

$$F_{out}^{i\times j} = \left(\left(\hat{g}_{11},\ldots,\hat{g}_{1j}\right),\left(\hat{g}_{21},\ldots,\hat{g}_{2j}\right),\ldots,\left(\hat{g}_{i1},\ldots,\hat{g}_{ij}\right)\right)$$

where $\hat{g}_{ij}$ is the value of pixel in the location (i, j) of output. Then, $F_{out}^{i\times j}$ will be used as the input of capsule layer to finish the classify step. The output of capsule layer is defined as:

$$P = \left\{p_1, p_2, \ldots p_j\right\}$$

where p_i is the probability of each category. In general, most regions of an image are background which means most of $\hat{g}_{ij}$ are useless. However, the original capsule network pays too much attention to them. This is the primary cause for the poor performance of the CapsNet when classifying images in complex background.

In this paper, a novel capsule network combined with the original capsule network and multi-lane structures is proposed. With three major improvements which called strict-squash function, drop-circuit and lanes filter, the new model outperforms original capsule network on a variety of tasks. The details of experiment can be seen in the Section 4.

A. CAPSULE NETWORK

Capsule Network is shallow with only two convolutional layers and one fully connected layer. The architecture firstly uses 256, 9 × 9 convolution kernels with a stride of 1 and ReLU activation on the input image to get the first layer of feature map. Then, another convolution of 9 × 9 convolution kernel is adopted to get the second layer of the feature map. The convolution kernel will regularly sweep through the input features. The matrix elements will be multiplied the matrix elements by the input features in the receptive field and superimpose the deviation. The process can be summarized as Eq.(1):

$$\begin{aligned} Z^{l+1}(i,j) &= \left[Z^l \otimes \omega^l\right](i,j) + b \\ &= \sum_{k=1}^{K_l}\sum_{x=1}^{f}\sum_{y=1}^{f}\left[Z_k^l\left(s_0 i + x, s_0 j + y\right)\omega_k^{l+1}(x,y)\right] \\ L_{l+1} &= \frac{L_l + 2p - f}{s_0} + 1 \end{aligned} \tag{1}$$

where $(i,j) \in \{0,1,\ldots,L_{l+1}\}$ and b is the bias. Z^l and Z^{l+1} represent the input and the output of layer l+1. l+1 is the size of Z^{l+1}. Z(i, j) correspond to the pixel of the feature map and K is the number of channels in feature map. f, s_0 and p represent the parameters of convolutional layer, the convolution kernel size and the padding layer respectively. CapsNet uses the second layer of feature map to generate the number of vectors called capsules and initialize an activation to each of these capsules independently. In the next layer, these capsules would be cluster according to the dynamic routing algorithm

Let u_i as the activation vector of capsule i in the previous layer. U_j is defined to represent the activation vector of capsule j in the next layer. $V_{j|i}$ could be comprehended as the inclination that capsule i is trending to be clustered into capsule j. The definition can be formalized as Eq.(2):

$$V_{j|i} = w_{j|i} \times u_i \tag{2}$$

The coupling coefficients between capsule i and all the capsules in the layer above sum to 1 are determined by a 'routing SoftMax' whose initial logits b_{ij} are the prior probabilities that capsule i should be coupled to capsule j which can be shown in Eq.(3), Eq. (4):

$$c_{ij} = \frac{\exp(b_{ij})}{\sum_k b_{ik}} \tag{3}$$

$$s_j = \sum_i \left(c_{j|i} \times v_{j|i}\right) \tag{4}$$

Squash function could be seen as a normalized operation on the weighted sum of votes from former layers which can be shown in Eq.(5):

$$u_j = Squash\left(S_j\right) = \frac{\left|\left|S_j\right|\right|^2}{1 + \left|\left|S_j\right|\right|^2}\frac{S_j}{\left|\left|S_j\right|\right|} \tag{5}$$

Finally, we just need to compute $c_{j|i}$ rather than update u_i or $v_{j|i}$ which could be represented by Eq.(6) while U_j is just the result after one round.

$$b_{j|i} = b_{j|i} + v_{j|i} \times u_j \tag{6}$$

The visualization of dynamic routing algorithm can be shown as Fig 1in which SF represent the Squash function:

B. MLSCN

Our novel capsule network, multi-lane capsule network with strict squash algorithm (MLSCN), as depicted in Fig 2, is based on the MLCN [31]. The components before lane-filter block can be seen as an improved MLCN with strict squash algorithm. The lane-filter block is utilized to achieve dynamic weighting assignment. In our model, each lane contains 3 convolution layer and 1 capsule layer. We adopt a stride of 1 for sliding kernels. Padding is used to ensure the size of feature map. The raw image is sent to the convolution block which have first one 1 × 1 convolution layer with 16× kernel_size kernels, followed by two convolution layers with

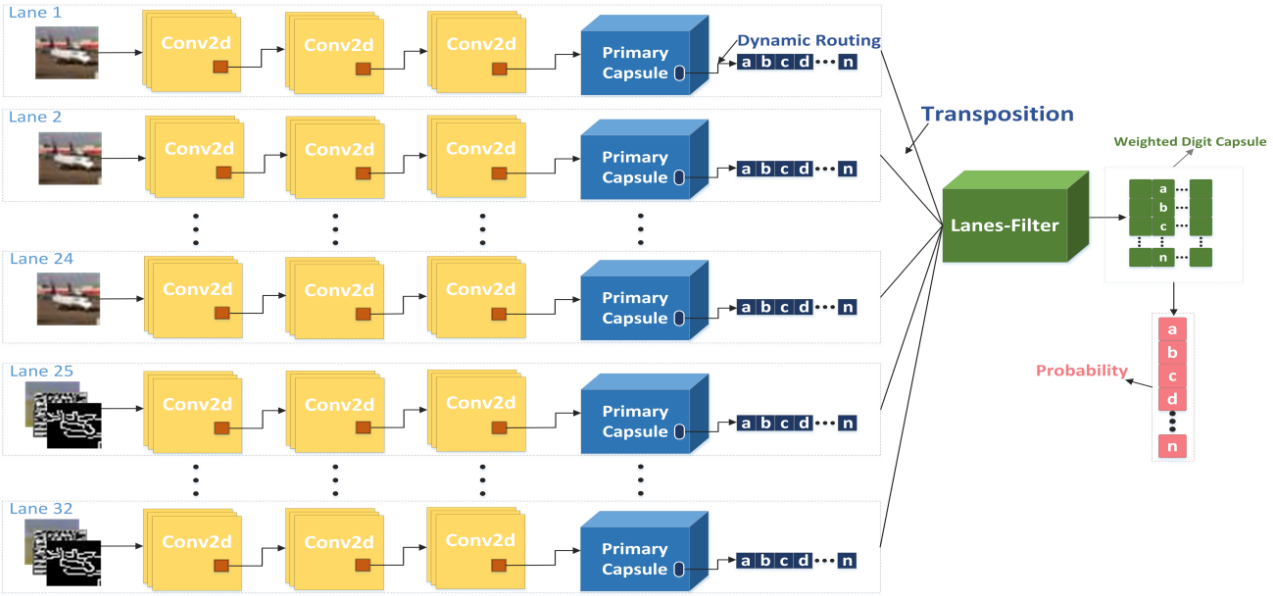

FIGURE 2. The architecture of MLSCN.

kernel size 9. Finally, the output of convolution block needs to be reshaped to align with the input of the primary capsule layer.

The output of the capsule network is a set of vectors called digitCaps. Each vector represents one of the classes in the classification problem, and we can reconstruct the input image by decoding them. When it is reconstructed, it was found that not only could the entire vector be reconstructed, but each dimension of the vector could also be reconstructed separately. This characteristic of reconstruction show that each dimension represents a distinctive feature of the image. By utilizing this assumption, the original capsule network can be split into lanes. The output of each lane is a 1×N vector of which each dimension is mapped to one of the dimensions in the digitCaps. Each lane can be calculated and adjusted separately. Such structure provides parallelism with negligible extra time overhead.

The output of each lane represents a dimension of digitCaps which can be seen as a feature of the input image. We find that the output of each lane has a different effect on the classification result. Due to the multi-lane structure, and by using approximately 32 lanes, the output is a 10×32 matrix. The original Capsule Network output is a matrix of 10×16, which means that the number of lanes is redundant. But such redundancy is necessary because the number of lanes required will vary from task to task. The number of lanes should be proportional to the complexity of the images to be classified. In order to guarantee the generalization of the model, the number of channels should always be set to a higher value. Intuitively, we think that each lane has extracted different features and each feature is of different importance in the final classification. Hence, we have a reason to assign weights to each lane to be able to change their contribution to the final classification results. Following the dynamic routing process, the output of the primary capsule layer is a 1×N vector which represents the result of classification that each lane predicted. The output of each lane is assembled in the Lanes-Filter, for which we contrive a filter function to get rid of the lanes that have a negative impact on the classification result. Simultaneously, the filter function is also responsible for assigning weights to beneficial lanes to amplify those lanes which have extracted excellent features. The function can be defined as follows:

$$\varepsilon_i = (w^T \cdot x_i + b_i)\frac{1}{||x_i||}$$
$$s.t. \sum_{i=1}^{n} \varepsilon_i = 1 \quad \varepsilon_i \geq 0 \tag{7}$$

where w is the weight matrix and x_i represent the output of lane i. b_i and ε_i represent the bias and the weight of lane i. We limit the sum of ε_i to meet the normalization requirements. Besides, the value of ε should be equal or greater than 0 to prevent the negative effect of certain lanes to the final probability generation. The output of each lane is transposed before being sent to filter block. They are then stitched into a matrix. The result of the model is the maximum value of the matrix after row wise summation. The process can be formalized as follows:

$$L_1 = [p_{11}, p_{12}, \ldots p_{1n}]^T$$
$$L_2 = [p_{21}, p_{22}, \ldots p_{2n}]^T$$
$$L_3 = [p_{31}, p_{32}, \ldots p_{3n}]^T \Longrightarrow L_{ij} = \begin{pmatrix} p_{11} & p_{21} & & p_{n1} \\ p_{12} & p_{22} & \cdots & p_{n2} \\ p_{13} & p_{23} & & p_{n3} \\ & \vdots & \ddots & \vdots \\ p_{1n} & p_{2n} & \cdots & p_{nn} \end{pmatrix}$$
$$\vdots$$

$$L_n = [p_{n1}, p_{n2}, \ldots p_{nn}]^T \tag{8}$$

$$\mathrm{Y} = \max(\sum_{\mathrm{i=1}}^{\mathrm{n}} L_{1i}) \tag{9}$$

where the L_i is the output of each lane and P_i represents the probability of each category.

Similar to the original capsule network, our model also has the problem of overfitting. Instead of using any dropout strategy, original capsule uses reconstitution as a way to prevent overfitting. Dropout [33], as a popular method to prevent overfitting, refers to the temporary discarding of neural network units according to a certain probability during the training phrase. Instead of using any dropout strategy, original capsule uses reconstitution as a way to prevent overfitting. However, traditional dropout and reconstitution do not work for our model because each lane should be seen as an independent neural network. Considering the parallelism of our network, a proper regularization method should target regular object to the lane. Inspired by [34], we adopt drop-circuit and add early stop mechanism to solve the problem. Drop-circuit is introduced to prevent lanes from adapting together. The method can be shown as Fig 3.

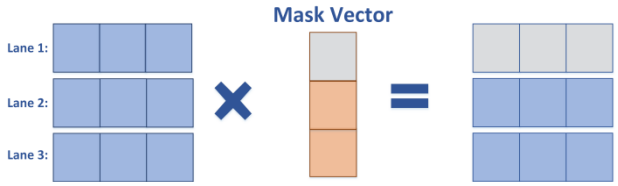

FIGURE 3. In this method, mask matrix changes into mask vector and each lane is treated as an integrated whole. The discarding rate is set as 0.33 and one lane is discarded in the result.

Our model needs a large number of channels to ensure that the network is convergent when adopt drop-circuit mechanism. The impact of discarding is excessive when the number of lanes is low, so as to model be unable to get optimum to solve. Besides, it takes several rounds of training to regain useful information when drop-circuit happened. Therefore, if discards occur in the last few rounds of training, the accuracy of the model will be affected. Early stop is adopted to solve this problem.

C. STRICT-SQUASH

A view that is worth considering for the capsule network is that the activation value of a capsule represents the probability that a specific type of entity. Since there are multiple instance in an image, there should be a lot of bad capsules. These bad capsules have no contribution to the classification of target instances, and even reduce the accuracy. From this point of view, the poor performance of the capsule on CIFAR10 can be attributed to the fact that capsule networks are not able to successfully distinguish between powerful and powerless features.

In original capsule network, only 20 of $8 \times 12 \times 12$ primary capsules on average have largest weight coefficient max(chi) of j with value that is more than 0.15. The distribution of

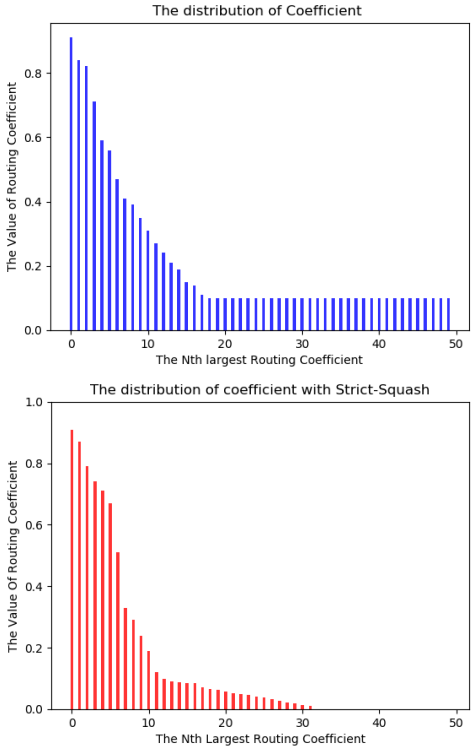

FIGURE 4. Visualization of the distribution of coefficient in the dynamic routing algorithm.

coefficient has an obvious characteristic of tailing which can be seen in Fig (4). We ascribe such distribution to two reasons: the rapid growth at the beginning of the original squash function and the sensitiveness of routing mechanism. The original squash function prone to generation a large activation value even if a capsule only has small $\|s_j\|$. In addition, the replacement of pooling layer with routing mechanism make CapsNet has to account for everything in the image. The result is that information transmitted between the layers become more and more confuse with the increasing of the training cycle. Briefly, the probability of certain instance will be amplified. This is attribute to the large number of capsules that encode them and the character of routing mechanism, not because of the importance of these instances.

To solve this problem, one reasonable modification is to introduce sparsity to restrain capsules from getting a high activation value easily. In order to active useful high-level capsule, it is necessary to highlight low-level capsules that are helpful to classification. We propose a novel squash function based on [32] named strict-squash function. The function grows slowly when $\|s_j\|$ is small and accelerate quickly when $\|s_j\|$ exceeds a certain value. Then, the function drops after reaching the peak value. Finally, function value will converge to 0.6 when $\|s_j\|$ reach the threshold. The function can be formalized as Eq. (10):

$$S(x) = \begin{cases} \alpha \|s_j\|^2 \times 2^{(-\beta\|s_j\|+1)} \times \dfrac{s_j}{\|s_j\|} & \|s_j\| \leq threshold \\ 0.6 & \|s_j\| > threshold \end{cases} \tag{10}$$

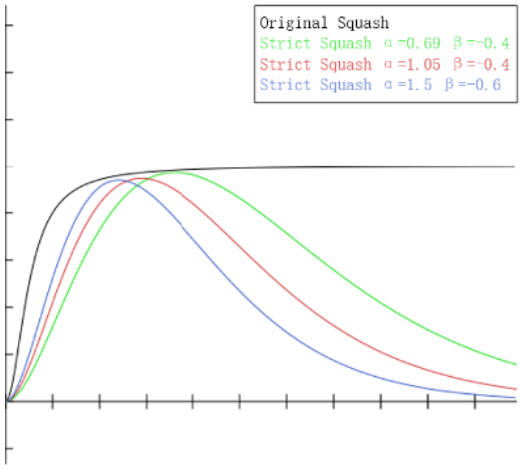

FIGURE 5. Comparison between original squash function and strict-squash with different hyperparameter.

where the parameter α and β are hyperparameters. The effect of these two hyperparameters on the function can be seen in Fig 5. s_j is a vector and $\|s_j\|$ is the module of s_j. $\frac{s_j}{\|s_j\|}$ is used to restrict the function value between 0 and 1.

As we can see in Fig 5, strict-squash function has a gentle trend at the beginning. In order to highlight those powerful low-level capsules, the function should decrease at a slower rate after reaching the peak. Because of these two requirements, we set $\alpha = 0.69$ and $\beta = -0.6$ as the final solution.

IV. EXPERIMENTAL RESULTS

In order to evaluate the effectiveness of our model, we conduct experiments on three public datasets, i.e., MNIST, CIFAR10, affNIST and our own dataset.

A. DATASETS

The datasets used for the experiments in this paper are briefly introduced below:

MNIST MNIST has a training set of 60,000 examples and a test set of 10,000 examples. Each example is a monochrome of handwritten digit. The digits have been size-normalized and centered in a fixed-size image.

CIFAR10 The CIFAR-10 dataset consists of 60000 32×32 color images in 10 classes, with 6000 images per class. There are 50000 training images and 10000 test images. The test batch contains exactly 1000 randomly selected images from each class. The training batches contain the remaining images in random order. We use some tricks on the training set, such as data augment for those data with certain label to enrich the number of training sets.

affNIST affNIST is modified from MNIST which applies various reasonable affine transformation. The examples in the affNIST become 40×40 pixels large, with significant translations involved. The dataset is split into training, validation, and test data, which contain 50,000, 10,000, and 10,000 cases respectively.

MNIST-CB We randomly generate 70,000 colored backgrounds and use them to replace the pure black background in MNIST. In the process, the image become 40×40 pixels large with slight deformations involved.

B. IMPLEMENTATION DETAILS

All the experiments are performed on a standard workstation with the following configuration, CPU: Intel®Xeon(R) CPU E5-2630 v3 @ 2.40GHz × 32; GPU: Quadro K620 & Tesla K40c; RAM: 64 GB. The model trained for a total of 7.6 hours and about 5000 iterations tended to be stable.

C. QUANTITATIVE EVALUATION

In this experiment, the evaluation metric is classification accuracy. The experiment results can be shown in Table 1. From the results, we observe that our model achieves best results on 3 out of 4 benchmarks, which verifies the effectiveness of our model. In the experiment, we set the number of channels of model to 32, adopting the drop-circuit mechanism and strict-squash. Drop-circuit has a rate of 5%. The hyperparameters of strict-squash are set to 0.69 and -0.6, respectively. In particular, we pre-processed Cifar10 and set its input ratio to 12:2:1.

TABLE 1. Comparisons of our capsule networks and baselines on four image classification benchmarks.

Model	MNIST	MNIST-CB	affNIST	CIFAR-10
CapsNet	99.75	41.62	79.16	59.8
CapsNet-R	99.65	42.21	81.63	57.61
P-Capsule	99.62	41.26	66.03	63.71
MLCN	98.16	58.73	74.62	75.18
MLSCN	99.73	65.37	81.71	76.79

In particular, we found that the accuracy of this model is not stable. In the average precision obtained after ten training phases, there is about a plus or minus 3% floating compare to the mean which is a large fluctuation. We speculate that the reason for this phenomenon is that the features of each lane fit are different, so that the final result of the model will fluctuate. Besides, many of the hyperparameters involved in the experiment were obtained through experiments. The results list in Table 1 are not the best performances that MLSCN can achieve.

D. ABLATION STUDY

In order to analyze the impact of different components on the classification results, we do experiments on the MLSCN with different setups. The experimental results are shown in Table 2. Generally, all three proposed methods contribute to the effectiveness of MLSCN. The number of lanes is positively related to classification accuracy.

We list all possible combinations of our improvement in Fig 6 and compare them with MLSCN. Drop-circuit will cause a wide range of fluctuation in the experiment because it may discard the lane that is important to the classification.

TABLE 2. Comparison with different setting on CIFAR10.

No	Lanes	Dropout	Squash	Lanes-Filter	Accuracy
1	6	Normal	Original	Yes	34.19
2	16	No	Strict-Squash	Yes	53.73
3	32	No	Original	Yes	69.484
4	32	No	CI-Squash	No	67.198
5	32	No	PA(n=6)	No	67.836
6	32	No	Strict-squash	No	69.774
7	32	Drop-circuit	Original	No	72.614
8	32	Normal	Original	No	70.731
9	32	Drop-circuit	Strict-squash	No	74.541
10	32	Drop-circuit	Original	Yes	73.472
11	32	No	Strict-squash	Yes	75.653
12	32	Drop-circuit	Strict-squash	Yes	76.791

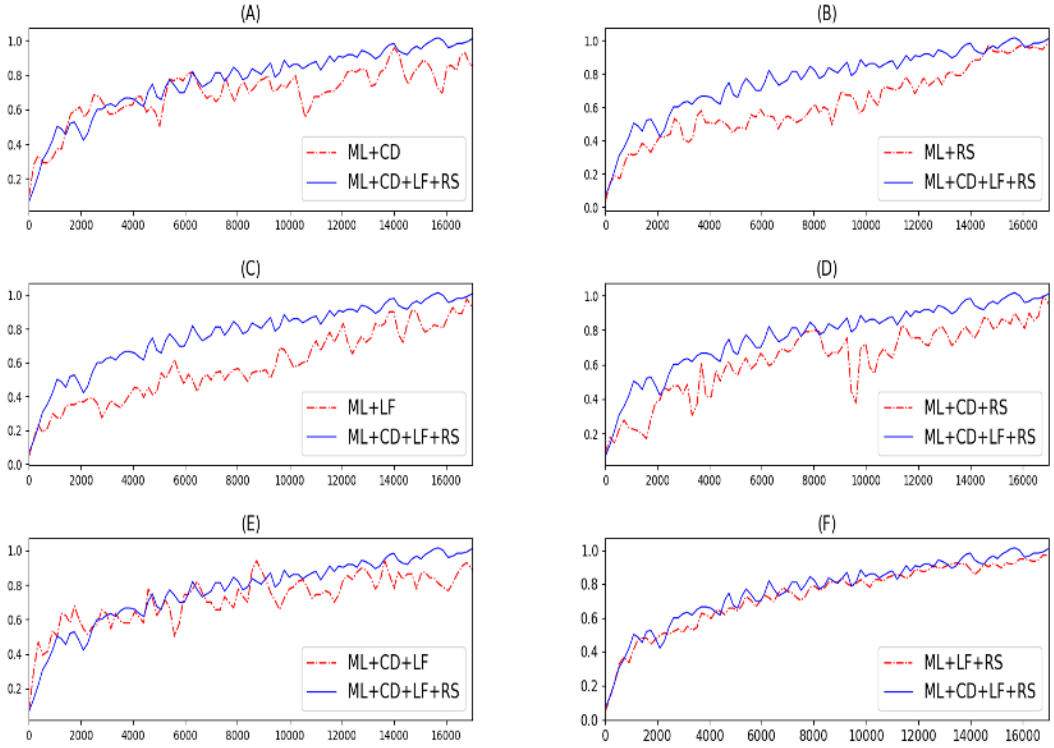

FIGURE 6. (A) Comparison between multi-lane combine with drop-circuit and ensemble; (B) Comparison between multi-lane combine with strict-squash and ensemble; (C) Comparison between multi-lane combine with lanes-filter and ensemble; (D) Comparison between multi-lane combine with drop-circuit, strict-squash and ensemble; (E) Comparison between multi-lane combine with drop-circuit, lanes-filter and ensemble; (F) Comparison between multi-lane combine with lanes-filter, strict-squash and ensemble.

Even if we limit it and stop it early, it is hardly to fit the data in the final stage of the experiment which can be seen in Fig 6-A. Combine with Lane-Filter, the lane will not be discarded if its weight is greater than a certain threshold, which can be seen in Fig 6-B. Moreover, we find that the combination of drop-circuit and strict-squash appears to oscillate over time as be seen in Fig 6-D. This oscillation can be attributed to the discarding of certain lanes which contain powerful features. Considering that drop-circuit is completely random, this situation is possible without the protection mechanism of Lane-filter. Compare with Fig 6-B, Fig 6-F converges faster about 2500 steps faster than Fig 6-B when achieve the same accuracy and it had a smoother ascent process.

E. SQUASH FUNCTION COMPARISON

We have compared strict-squash with some common regularization method and the other two squash improvement called CI-squash and PA. The CI-squash can be shown as:

$$u_j = \{\frac{-ReLU\left(-s_j + bar\right) + bar}{bar} * \frac{s_j}{\}}|\,|s_j|\,| \qquad (11)$$

where bar is hyperparameter that represents threshold of the capsule length. When the input length of capsules is greater than its threshold, the activation value will be set to 1.

The PA can be defined as

$$Power_n(x) = ||x||^n * \frac{x}{||x||}$$
$$new\_u_j = Power_n(u_j) \quad (12)$$

where the s_j is the same as previously defined and bar is a threshold. In particular, the activation will be set to 1 when $||s_j||$ exceeds this threshold.

TABLE 3. Comparisons of our capsule networks and baselines on four image classification benchmarks.

Method	CIFAR-10
MLSCN	73.472
MLSCN + CI	72.874
MLSCN + PA (n=6)	74.053
MLSCN + Strict-Squash	76.791

The experimental results can be seen in Table 3. We notice that CI-squash and PA-squash do improve our model classification results. This shows that the modification of squash function is a general way to enhance the performance of the CapsNet. Besides, strict-squash achieves the best result in this experiment, which enhance accuracy by 2.7% compared to other squash function.

F. FEATURE SELECTION

The original Capsule Network lacks the ability to extract features directly from rich background images. To solve this problem, we enrich the inputs of the model. In our experiments, we use three types of preprocessing LAB, Texture and Canny to enrich the input.

Such data augments are reasonable because some features of the image are enhanced after pre-processing. For example, we often adopt canny algorithm to enhance the edge feature, which can be seen in the 2th and 10th columns of Fig 7. However, this method will bring negative effects because it does discard all color information in the picture. Besides, this operation is not suitable to all kinds of pictures. For example, in the 8th columns of Fig 7, it brings a lot of noise.

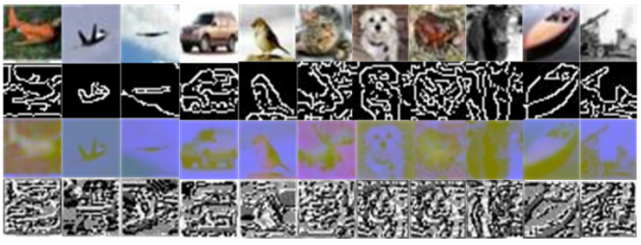

FIGURE 7. From top to bottom, each line corresponds to a type of image (Raw, Canny, Lab, Texture) and contains 10 different labels.

Based on previous work [35], we find that the number of lanes is one of the most important factors to the result of classification in the parallel network. Therefore, we assume that different input and mixing ratios will have a great impact on the experimental results.

In the experiment, we found that increasing the number of lanes does not guarantee the rise of accuracy of the classification results. The accuracy of the 48 lanes architecture is almost the same as that of the 32 lanes. We configure the model with more lanes than 48, but the accuracy reduces instead. We believe this is because too many lanes are mixed with large amount of useless information and the model could hardly distinguish between them, although we have already added filter and regularization to the network.

We notice that in the final classification results, there are two types of labels were not satisfactory. Hence, we augment the data for these two tags separately to enrich the input for higher validation accuracy. The effect of adjusting the proportion of different types of images on the experimental results can be seen in Table 4.

G. LANES FILTER

In this experiment, we designed a visual approach to reflect the effects of Lanes-Filter. We choose the best setup in the ablation experiment as the target of visualization. We record the weight of each lane separately. These weights are accumulated and averaged according to the image type which can be seen in Fig 8, 9.

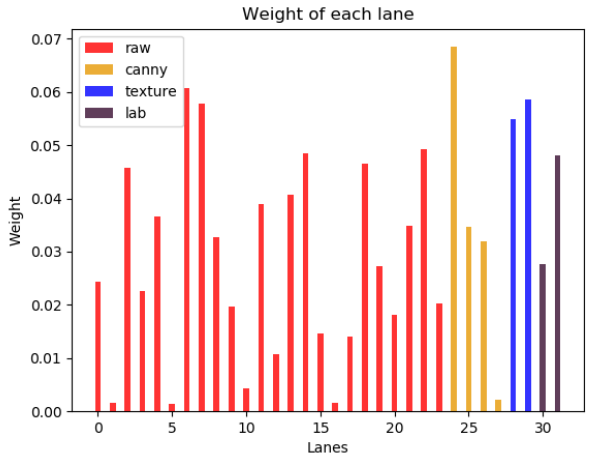

FIGURE 8. Weight of each lane.

From the Fig 8, we can see that each lane is assigned different weight, and some channels have small weights imply that these lanes is less important than the others. We find that the raw image occupies the largest part of the sum of weights while the texture has the highest average weight.

In Figure 9, we observe that the average weights of the three pretreated images are 0.033, 0,058, and 0.037, respectively. This verifies that these images have a great impact on the classification results. After preprocessing the image, parts of information in the image is discarded (such as color information). Only a few numbers of lanes are needed to

TABLE 4. Ablation experiments of feature selection on CIFAR10.

No	Raw	Canny	Texture	Lab	Total	Accuracy
1	6	0	0	0	6	34.19
2	3	1	1	1	6	33.71
3	16	0	0	0	16	49.56
4	10	2	2	2	16	53.73
5	24	4	2	2	32	**76.79**
6	32	0	0	0	32	74.34
7	34	6	4	4	48	75.87
8	40	6	4	4	54	73.18

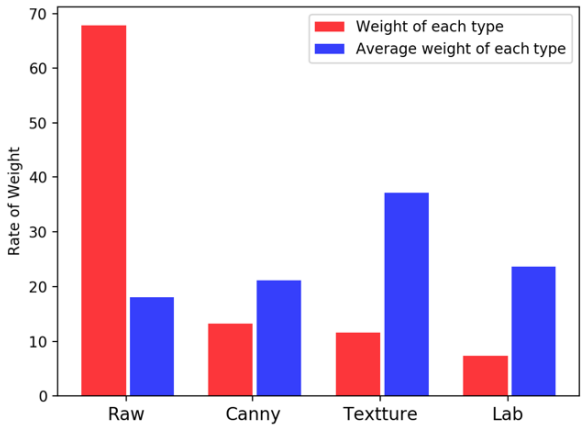

FIGURE 9. Weight of each lane and its average weigth.

H. ROBUSTNESS OF MLSCN

We added a complex background to the MNIST dataset to verify the robustness of our model. The sample of our modified dataset can be seen in Fig 10.

FIGURE 10. The right column shows the original MNIST digit and the other 12 columns show complex background versions.

We replace the black background of the original MNIST with some other colors to produce complex background. In addition, we carried out edge corrosion on the object to increase the difficulty of classification. Some image in the modified dataset become illegible because the corrosion and background replacement are random.

Our model achieved the similar accuracy with capsule network and got a much higher classification result in complex backgrounds than capsule network. The validation accuracy curve of our model rises faster than the capsule network and eventually maintain stable. The process is shown in Fig 11. As a result, our model achieved a classification accuracy of 65.37, which is 23.75% higher than the CapsNet.

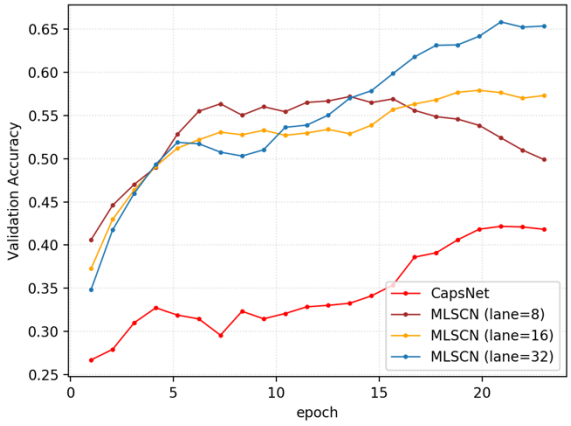

FIGURE 11. Comparisons of MLSCN and CapsNet on MNIST-BC.

The experiment proves that the architecture of our model can handle different types of complex backgrounds whether it is natural or artificial. Besides, this also proves that Our models can directly apply to different datasets without modifying.

I. ROBUSTNESS TO AFFINE TRANSFORMATIONS

Capsule network makes a remarkable performance with a simple architecture which could be attributed to the ability of using equivariance instead of invariance during inference. Benefit from the equivariance, the capsule network is more robust than traditional convolution network.

To test the robustness of capsule network to affine transformations, we trained a MLSCN and a traditional convolutional network on a padded and translated MNIST training set. We then tested this network on the affNIST dataset, in which

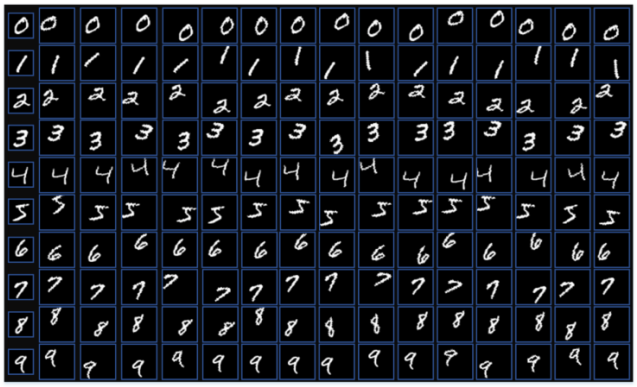

FIGURE 12. MNIST digit with a random small transformation. The left column shows the original MNIST digit, and the other 16 columns show transformed versions.

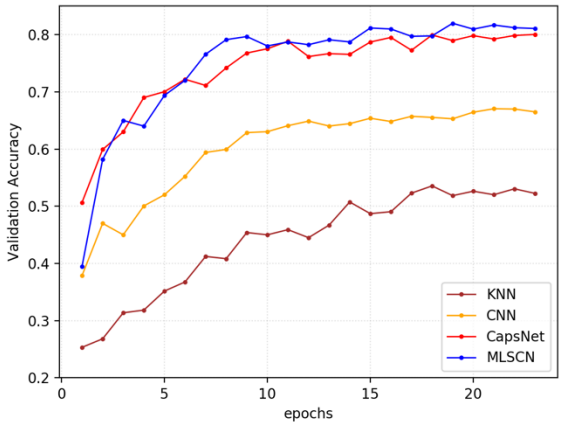

FIGURE 13. Comparison between different method.

each example is an MNIST digit with a random small affine transformation which can be seen in Fig 12. Our model which achieved 98.42% accuracy on the MNIST test set achieved 77% accuracy on the affNIST test set while a traditional convolutional model which achieved similar accuracy on the standard MNIST test set only achieved 66% on the affNIST test set. The process is shown in Fig 13 and the results are list in Table 5.

TABLE 5. Comparison ot the capability for transferring from MNIST TO affNIST.

Method	MNIST	affNIST
CNN	99.22	66
Capsule	99.23	79
MLSCN	99.27	81
KNN	94.65	53
SVM	98.56	44

V. CONCLUSION

In this paper, we propose a multi-lane capsule network called MLSCN. Our major innovation is the multi-lane structure and strict-squash function. Besides, we propose lanes-filter as an attention mechanism to focus on those powerful lanes. Drop-circuit is introduced to prevent overfitting. Through these modifications, we enhance the classification accuracy when handle images with complex scenes. Experimental results show that our model achieves the best performance on four image classification benchmarks. Compare to a single type of input, we prove that multiple types of input can make this multi-lane model perform better. However, drop-circuit cannot accurately identify the lanes that adapt together. This dropout mechanism introduces some randomness to experiment result. Therefore, a more controllable dropout method is worth further researching, and we will keep working on the allocation strategy of weight and the optimization of drop-circuit mechanism.

REFERENCES

[1] Y. Yang, X. Wang, and H. Zhang, "Local importance representation convolutional neural network for fine-grained image classification," *Symmetry*, vol. 10, no. 10, p. 479, Oct. 2018, doi: 10.3390/sym10100479.

[2] J. Liu, Y. Yang, S. Lv, J. Wang, and H. Chen, "Attention-based BiGRU-CNN for Chinese question classification," *J. Ambient Intell. Humanized Comput.*, pp. 1–12, 2019, doi: 10.1007/s12652-019-01344-9.

[3] K. Fukushima, "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position," *Biol. Cybern.*, vol. 36, no. 4, pp. 193–202, Apr. 1980.

[4] K. Fukushima, S. Miyake, and T. Ito, "Neocognitron: A neural network model for a mechanism of visual pattern recognition," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-13, no. 5, pp. 826–834, Sep. 1983.

[5] Y. LeCun and Y. Bengio, "Convolutional networks for images speech and time-series," in *The Handbook of Brain Theory and Neural Networks*. Cambridge, MA, USA: MIT Press, 1995.

[6] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1–9.

[7] S. Xie, R. Girshick, P. Dollar, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1492–1500.

[8] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1251–1258.

[9] Y. Chen, J. Li, and H. Xiao, "Dual path networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 4467–4475.

[10] Y. LeCun, B. E. Boser, and J. S. Denker, "Handwritten digit recognition with a back-propagation network," in *Proc. Adv. Neural Inf. Process. Syst.*, 1990, pp. 396–404.

[11] X. Zhang, Z. Li, C. C. Loy, and D. Lin, "PolyNet: A pursuit of structural diversity in very deep networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 718–726.

[12] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, p. 436, 2015.

[13] S. Sabour, N. Frosst, and G. E. Hinton, "Dynamic routing between capsules," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 3856–3866.

[14] S. Sabour, N. Frosst, and G. Hinton, "Matrix capsules with EM routing," in *Proc. 6th Int. Conf. Learn. Represent. (ICLR)*, 2018, pp. 1–15.

[15] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[16] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017.

[17] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*. [Online]. Available: http://arxiv.org/abs/1409.1556

[18] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[19] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 580–587.
[20] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.
[21] K. He, G. Gkioxari, and P. Dollár, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, Jun. 2017, pp. 2961–2969.
[22] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.* Cham, Switzerland: Springer, 2015, pp. 234–241.
[23] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3431–3440.
[24] A. Shahroudnejad, P. Afshar, K. N. Plataniotis, and A. Mohammadi, "Improved explainability of capsule networks: Relevance path by agreement," in *Proc. IEEE Global Conf. Signal Inf. Process. (GlobalSIP)*, Nov. 2018, pp. 549–553.
[25] M. Pöpperl, R. Gulagundi, S. Yogamani, and S. Milz, "Capsule neural network based height classification using low-cost automotive ultrasonic sensors," 2019, *arXiv:1902.09839*. [Online]. Available: http://arxiv.org/abs/1902.09839
[26] A. Jaiswal, W. AbdAlmageed, Y. Wu, and P. Natarajan, "CapsuleGAN: Generative adversarial capsule network," 2018, *arXiv:1802.06167*. [Online]. Available: http://arxiv.org/abs/1802.06167
[27] J.-S. Amelia, S. Albarqouni, and D. Mateus, "Capsule networks against medical imaging data challenges," *Intravascular Imaging and Computer Assisted Stenting and Large-Scale Annotation of Biomedical Data and Expert Label Synthesis*. Cham, Switzerland: Springer, 2018. pp. 150–160.
[28] A. Mobiny and H. Van Nguyen, "Fast capsnet for lung cancer screening," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.* Cham, Switzerland: Springer, 2018.
[29] R. Mukhometzianov and J. Carrillo, "CapsNet comparative performance evaluation for image classification," 2018, *arXiv:1805.11195*. [Online]. Available: http://arxiv.org/abs/1805.11195
[30] D. Wang and Q. Liu, "An optimization view on dynamic routing between capsules," in *Proc. 6th Int. Conf. Learn. Represent. (ICLR)*, 2018, pp. 1–4.
[31] V. Martins do Rosario, E. Borin, and M. Breternitz, Jr., "The multi-lane capsule network (MLCN)," 2019, *arXiv:1902.08431*. [Online]. Available: http://arxiv.org/abs/1902.08431
[32] Z. Yang and X. Wang, "Reducing the dilution: An analysis of the information sensitiveness of capsule network with a practical improvement method," 2019, *arXiv:1903.10588*. [Online]. Available: http://arxiv.org/abs/1903.10588
[33] N. Srivastava, G. Hinton, and A. Krizhevsky, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.
[34] K. T. Phan, T. H. Maul, T. T. Vu, and W. K. Lai, "DropCircuit: A modular regularizer for parallel circuit networks," *Neural Process. Lett.*, vol. 47, no. 3, pp. 841–858, Jun. 2018.
[35] M. Amer and T. Maul, "Path capsule networks," 2019, *arXiv:1902.03760*. [Online]. Available: http://arxiv.org/abs/1902.03760

SIWEI CHANG received the B.E. degree from Shanghai Maritime University, in 2014, where he is currently pursuing the M.E. degree. His current research interests include data mining, natural language processing, and machine learning.

JIN LIU (Member, IEEE) received the B.S. degree from Lanzhou University, the M.S. degree from the University of Electronic Science and Technology of China, and the Ph.D. degree from Washington State University. He is currently a Professor with Shanghai Maritime University. His research interests include deep learning, nature language processing, and computer vision. He is a member of CAAI and CCF.