Received October 22, 2019, accepted November 24, 2019, date of publication November 28, 2019,
date of current version December 12, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2956508

A Survey on the New Generation of Deep Learning in Image Processing

LICHENG JIAO, (Fellow, IEEE), AND JIN ZHAO

Key Laboratory of Intelligent Perception and Image Understanding of the Ministry of Education, School of Artificial Intelligence,
Xidian University, Xi'an 710071, China

Corresponding author: Licheng Jiao (lchjiao@mail.xidian.edu.cn)

This work was supported in part by National Key R&D Program of China (No. 2018YFC0825305, No. 2018YFC0825303), the State Key Program of National Natural Science of China (No. 61836009), the National Natural Science Foundation of China (No. 61501353, No. 61573267, No. 61473215), the Natural Science Pre-Research Fund of Shaanxi University of Science and Technology (No. 2019BJ-11), in part by the Program for Cheung Kong Scholars and Innovative Research Team in University (No. IRT_15R53), in part by The Fund for Foreign Scholars in University Research and Teaching Programs (the 111 Project) (No. B07048).

ABSTRACT During the past decade, deep learning is one of the essential breakthroughs made in artificial intelligence. In particular, it has achieved great success in image processing. Correspondingly, various applications related to image processing are also promoting the rapid development of deep learning in all aspects of network structure, layer designing, and training tricks. However, the deeper structure makes the back-propagation algorithm more difficult. At the same time, the scale of training images without labels is also rapidly increasing, and class imbalance severely affects the performance of deep learning, these urgently require more novelty deep models and new parallel computing system to more effectively interpret the content of the image and form a suitable analysis mechanism. In this context, this survey provides four deep learning model series, which includes CNN series, GAN series, ELM-RVFL series, and other series, for comprehensive understanding towards the analytical techniques of image processing field, clarify the most important advancements and shed some light on future studies. By further studying the relationship between deep learning and image processing tasks, which can not only help us understand the reasons for the success of deep learning but also inspires new deep models and training methods. More importantly, this survey aims to improve or arouse other researchers to catch a glimpse of the state-of-the-art deep learning methods in the field of image processing and facilitate the applications of these deep learning technologies in their research tasks. Besides, we discuss the open issues and the promising directions of future research in image processing using the new generation of deep learning.

INDEX TERMS Image processing, deep learning, convolutional neural network, generative adversarial network, extreme learning machine, deep forest, capsule networks, ADMM-Net, image classification, style transfer, object detection, super-resolution.

I. INTRODUCTION

Since images play an essential role in our daily life, and as the advances in computer information collection systems, one can obtain more and more image sets, but most of them cannot be processed manually [1]–[3]. Hence image processing becomes attractive since much of this image information can be represented and processed digitally. With the fast computers and signal processors available in the 2000s, image processing has become the most common processing technique to be used in medical images, remote sensing image and nature image [4] (see Fig.1), and generally, is used because it is not only the most versatile method but also the cheapest [5]. Image processing has been playing a more vital and essential role in various information access systems to enhance the cognition level and facilitate decision-making process [6]. In pattern recognition and machine learning, the commonly used image processing includes image generation, image compression and encoding, image deblurring, super-resolution, image segmentation, classification and object recognition, change detection, image annotation, and image retrieval, etc. In particular, machine learning

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

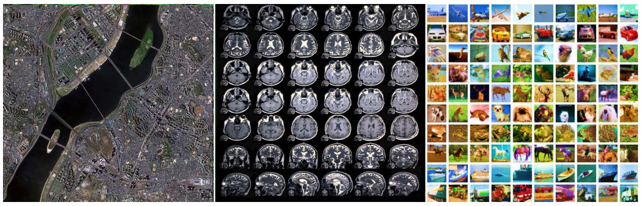


FIGURE 1. Three typical sample type of image processing. Left: remote sensing image [8], Middle: medical images [9], Right: nature image [10].

techniques has been widely and successfully applied to image processing research [7].

Compared with the nonlearning-based methods that might not precisely translate domain knowledge into rules or features, machine learning acquires its knowledge from data representations [11]. In other words, while the low-level features can be hand-crafted with great success for some specific data and tasks, designing useful features for new data and tasks requires new domain knowledge since most hand-crafted features cannot adapt to new conditions [12]. Besides, conventional machine learning techniques usually do not directly deal with raw data but heavily rely on the data representations, which further to require considerable domain expertise and sophisticated engineering [13]–[15].

Learning higher-level features from the data of interest is considered as a plausible way to remedy the limitation of hand-crafted features [16]–[19]. A successful example of such methods is learning through the framework of deep networks, which draws significant attention recently. Compared with hand-crafted features, learned multiple levels of representations require less human interventions and provide much better performance. There is no doubt that deep learning techniques have made significant advances in image processing. The core idea of deep learning is to discover multiple levels of representation and automatically learns the representations, with the hope that higher-level features represent more abstract semantics of the data [20]. Meanwhile, such abstract representations learned from deep networks are expected to provide more invariance to intra-class variability [21], [22].

To further improve the readability of this survey, we present the phrases commonly and explanations that appeared in deep learning in TABLE 1. As is known to all, one key ingredient in the success of deep learning in image processing is the use of CNNs [23], which includes a convolutional flow module stacked on top of each other. Moreover, each convolutional flow module comprises four parts— convolution filter bank layer, feature maps pooling layer, nonlinear processing layer, and BN. Many classical variations on deep CNNs have been proposed to different tasks in image processing, such as LeNet, AlexNet, GoogleNet, VGGNet, R-CNN, YOLO, SSD, SqueezeNet, ResNet, DenseNet [24], SegNet, and DCGAN, etc. and their success is usually justified empirically [25]–[27]. With the development of computing power and data scale, many classical deep learning algorithms, which based on the SGD method, have verified,

TABLE 1. An explanation of phrases commonly.

Initialism	Full Spelling of Initialism
CNNs	Convolutional Neural Networks
R-CNN	Region-based Convolutional Neural Network
YOLO	You Only Look Once
SSD	Single Shot MultiBox Detector
ResNet	Residual Networks
DenseNet	Densely connected Convolutional Neural Network
SegNet	Segmentation Networks
GAN	Generative Adversarial Networks
DCGAN	Deep Convolutional Generative Adversarial Networks
SGD	Stochastic Gradient Descent
AE	Auto-Encoder networks
DBN	Deep Belief Networks
RNN	Recurrent /Recursive Neural Networks
BP	Back-Propagation
CV-CNN	Complex Value Convolutional Neural Network
SRCNN	Super-Resolution Convolutional Neural Network
Mask R-CNN	Mask Region-based Convolutional Neural Network
FCN	Fully Convolutional Networks
FractalNet	Fractal Networks
C-GAN	Conditional Generative Adversarial Networks
PTGAN	Person Transfer Generative Adversarial Networks
LAPGAN	Laplacian Generative Adversarial Network
TAC-GAN	Text conditioned Auxiliary Classifier GAN
SegAN	Segmentation Generative Adversarial Network
CoGAN	Coupled Generative Adversarial Network
SRGAN	Super-Resolution Generative Adversarial Network
ELM	Extreme learning machine
RVFL	Random Vector Functional Link
C-ELM	Complex Extreme Learning Machine
H-ELM	ELM for Multilayer Perceptron
BLS	Broad Learning System
F-BLS	Fuzzy Broad Learning System
ADMM	Alternating Direction Method of Multipliers
ADMM-Net	ADMM Networks
CapsuleNet	Capsule Networks
ML-CSC	Multi-Layer Convolutional Sparse Coding
CSC	Convolutional Sparse Coding
VAE	variational Auto-Encoder
DDL	Deep Dictionary Learning
FPN	Feature Pyramid Network
RoI	Region of Interest
RPN	Region Proposal Networks
PCA	Principal Component Analysis
PSPNet	Pyramid Scene Parsing Network
BN	Batch Normalization
DnCNNs	Denoising Convolutional Neural Networks
CBIR	Content Based Image Retrieval

but only under an apparent theoretical premise can we know what the worst is. At present, The uncertainty of deep learning is mainly reflected in three aspects [28],

- The BP algorithm for updating weights and bias causes gradient diffusion or explosion.
- The initialization method of network weights/bias affects the solution to non-convex optimization problems.
- The regularization methods affect the generalization performance of the deep learning model.

Besides, many classical deep learning algorithms rely on a massive amount of datasets because the singularity of the sample does not effectively remove, and the spatial logic or structure relationship of the target for the sample is ignored, etc [29]. According to the different requirements of the application task, at the same time, to solve the existing problems of classical deep learning (such as AE, CNN, DBN, RNN, and GAN, etc.), a variety of new generation algorithms and frameworks of deep learning have proposed. These latest frameworks not only improved the network generalization performance and significantly improved the efficiency of optimization but also enriched the research system of deep learning [30], [31].

A. MOTIVATIONS

In recent years, researchers have published many reviews/ surveys papers related to deep learning technology in the field of image processing, such as the works in the fields of microscopic image analysis [32], hyperspectral image analysis [33] and medical image analysis [34] etc. Besides, the work of Tian et al. summarize the related processing ideas of deep learning in specific image denoising tasks [35]. Deng et al. summarize the deep stacked model under the semi-supervised learning paradigm from the design and application points of view [36]. These works have positive enlightenment for the improvement of existing methods and model selection of specific application background. However, the types of deep models mentioned in these works are not abundant, none of them can provide a mathematical principle of the models, and they lack some of the most recent deep learning models. Compared with the above reviews/surveys, an initial motivation of this survey is trying to summarize the application of these new-generation deep learning algorithms and frameworks in image processing and to evaluate the effectiveness of these new generations of deep network framework and acquire more inspiration for deep network's design and optimization tricks [37].

In particular, the summary of the new generation of deep learning models will follow along the four main lines, 1) Deep CNN and its improved deep network structure [38], the classic framework include CV-CNN, SRCNN, Mask R-CNN, FCN, U-Net, DenseNet, FractalNet, etc. 2) GAN [39] and its improved deep network structure, the classic framework includes DCGAN, PTGAN, InfoGAN, LAPGAN, TAC-GAN, SegAN, CoGAN etc. 3) ELM [40] or RVFL [41] (denote as ELM-RVFL) and its improved deep network structure, the classic framework includes C-ELM, H-ELM, BLS, F-BLS, etc. 4) Other typical new generation deep network structure includes Deep Forest, ADMM-Net, CapsuleNet, ML-CSC [42], VAE, PCANet, DDL, etc. More concretely, these can be summarized as TABLE 2 and TABLE 3.

In this survey, our main intention is to focus on a new generation of deep learning models and their mathematical principles. It is worth pointing out that we have three criterions for choosing several representative models. One is that this network has a novel topological structure. Second, excellent generalization performance can achieve in specific image application tasks. Third, the research results generally accepted by scientific researchers, and there are further development and improvement for a deep model. Most of the other deep learning models can be variants of these four deep architectures' main lines. In the following parts, we review the four typical deep learning series models in detail. Also, we will mainly describe the generalization performance of the corresponding new generation of deep learning algorithm from the perspective of the application task, which covers various topics, such as image classification, style transfer, object recognition, super-resolution, image compression, image segmentation, change detection, image denoising. Specifically, the framework structure flow diagram of this survey shown in Fig.2.

B. CONTRIBUTIONS

Nowadays, deep learning is the dominant method of more excellent solutions to many tasks in image processing. The survey is not only to provide a systematic overview of deep learning in image processing, but also presents a dedicated discussion on open challenges, unsolved problems, and potential future trends. Specifically, we introduce the contributions of deep learning to different image processing tasks and present the current effort devoted to addressing these issues by the new generation of deep learning. Furthermore, we point out several potential future research trends of deep learning in image processing. Meanwhile, we analysis the fundamental theoretical insights about the new generation deep networks in detail, it seems the pressing need for deep learning nowadays. Finally, This survey aims to improve or arouse other researchers to catch a glimpse of the state-of-the-art deep learning methods in the field of image processing and facilitate the applications of these deep learning technologies in their research tasks. It is particularly worth pointing out that, as far as many neurocognitive mechanisms are still further developed and perfected, the work of network modeling based on biological neural enlightenment is still a long way off.

The rest of this survey as structured as followed. In Section II we introduce the new generation of deep learning techniques that have used for image processing and that referred to throughout the survey. Section III describes the contributions of deep learning to canonical tasks in image processing: image classification, style transfer, object

TABLE 2. Some new generation of deep learning models for image processing tasks.

Series	Network	Characteristics	Representational Image Processing Tasks
CNN Series	CV-CNN (2016)	Compared with CNN, CV-CNN prefers to process complex-valued images, but the computational cost is relatively higher. Supervised learning	Polarimetric SAR Image Classification [49] Object Detection in PolSAR data [51] Enhanced Radar Imaging [52]
	SRCNN (2016)	SRCNN can restore high-resolution images from low-resolution images, but it is not ideal for texture detail restoration. Supervised learning	Image super resolution [53] Enhancing Image Resolution in Chest CT [54]
	Mask R-CNN (2017)	Compared with Fast R-CNN, Mask RCNN can achieve alignment correction for each region of interest. Supervised learning	object instance segmentation [55] Scene Text Detection [56] Building Extraction from Satellite Images [57]
	ResNet (2015)	ResNet can solve the bottleneck problem that the hierarchical information is constantly divergent as the number of layers deepens. Supervised learning	Image Classification Image Segmentation [58] Image Denoising [59]
	FCN (2015)	FCN has excellent performance in pixel level recognition tasks. In addition, the FCN is not limited by the input size. Supervised learning	Object Detection Image Segmentation
	U-Net (2015)	U-Net network can be regarded as an improvement of FCN. Its advantage is that U-Net can be used in the case of fewer samples, especially for medical images with fewer samples.	Biomedical Image Segmentation [60] Glaucoma Detection [61]
	DenseNet (2017)	Compared with ResNet, DenseNet proposes a dense connection mechanism: each layer accepts all the layers in front of it as its additional input. Supervised learning	Brain Tumor Segmentation [62] Image Classification
	FractalNet (2016)	FractalNet shows that path length is essential for training ultra-deep neural networks; the effect of residuals is accidental. Supervised learning	Image Classification Segmentation for Cardiovascular MRI [63] remote sensing image segmentation [64]
GAN Series	DCGAN (2014)	DCGAN is a architecture which combines CNN and GAN, but it still has some shortcomings such as unstable training. Unsupervised learning	Image Generation Image Inpainting [66]
	Info GAN (2016)	Info GAN can generate images with certain characteristics, which can solve the problem of interpretability of hidden variables. Unsupervised learning	Image Synthesis [67] Image Generation
	PTGAN (2017)	The advantage of PTGAN is to realize the migration between different background domains on the premise of keeping the foreground (pedestrian) unchanged. Unsupervised learning	Style Transfer [68] Image Generation
	LAPGAN (2015)	The innovation of LAPGAN lies in the use of Laplacian pyramid structure to generate pictures in CGAN in a rough to detailed way. Unsupervised learning	Super Resolution Image Generation
	TAC-GAN (2017)	A text to image GAN for synthesizing images from their text descriptions. Unsupervised learning	Synthesizing Images from Text Descriptions [69]
	SegAN (2018)	The main purpose of SegAN is medical image segmentation tasks. Semi-supervised learning	Medical Image Segmentation [70]
	CoGAN (2016)	Two GAN networks are coupled by weight sharing to generate cross-domain samples. Unsupervised learning	Style Transfer [71]
ELM-RVFL Series	C-ELM (2016)	Compared with ELM, C-ELM prefers to process complex data, but has poor performance in relatively complex classification tasks. Supervised learning	Image Classification Human Action Recognition [72]
	H-ELM (2017)	H-ELM is an effective improvement of ELM, whose core module is ELM sparse auto-encoder network. Semi-supervised learning	Image Classification Human Posture Detection [73]
	BLS (2018)	BLS is an effective improvement of RVFL and ELM. The network generalization performance can be improved by increasing the number of nodes. Semi-supervised learning	Image Classification Body Gesture Recognition [74] Face Recognition
	F-BLS (2018)	The fuzzy BLS replaces the feature nodes of BLS with a group of TS fuzzy subsystems, and the input data are processed by each of them. Semi-supervised learning	Image Classification [75]

TABLE 3. Some new generation of deep learning models for image processing.

Other Series	Deep Forest (2017)	Deep Forest is a novel decision tree integration method. Compared with DNN, Deep Forest has fewer parameters. Semi-supervised learning	Image Classification Hyper-spectral Image Classification [76]
	ADMM-Net (2016)	ADMM-Net is a model designed to solve the general compression sensing problem by alternating iteration algorithm. It has the characteristics of fast iteration update and good convergence. Unsupervised learning	Image Compression MRI Image Compressive Sensing [77]
	CapsuleNet (2017)	CapsuleNet is a vectorized CNN network. Its main advantage is that training requires relatively few training samples. Supervised learning	Image Classification Action Detection [78]
	ML-CSC (2018)	ML-CSC is a deep dictionary model whose core module is CSC. The advantage of ML-CSC model lies in the hierarchical sparse dictionary representation. Unsupervised learning	Image Compression
	VAE (2013)	VAE is a generative model. Compared with GAN, VAE training is relatively simple, but the quality of generated image is not high. Unsupervised learning	Image Generation Image Denoising [79]
	PCANet (2015)	PCANet is a very simple deep learning network for image classification, which includes cascaded principal component analysis (PCA), binary hashing and blockwise histograms. Supervised learning	Image Classification [80]
	DDL (2016)	Deep dictionary learning seeks multiple dictionaries at different image scales to capture complementary coherent characteristics. Supervised learning	Image Classification [81]

detection, super-resolution, image compression, semantic segmentation, and image denoising. Section IV discusses obtained results and open issues in application areas; meanwhile, we also objectively give a critical discussion and an outlook for further research.

II. RESEARCH PROGRESS OF NEW GENERATION OF DEEP LEARNING

Deep learning is often used as classifiers or feature extractors for various tasks in image processing [43], [44]. In the following parts, we mainly introduce several new generations of deep learning techniques that have used for image processing, and that referred to throughout the survey.

A. CNN SERIES MODELS FOR IMAGE PROCESSING

1) CNN

In recent years, CNN has also made great success in image processing and object recognition. The strength of CNN lies in their shared weights. Weight sharing dramatically reduces the number of free parameters learned, thus to lower the memory requirements for running the network and allowing the training of more extensive, more powerful networks [45].

A CNN consists of convolutional layers, pooling layers, normalization layers, and fully connected layers. At each layer, the input image $\boldsymbol{X} \in \mathbb{R}^{n\times m}$ is convolved with a set of K kernels $\{\boldsymbol{W}_k \in \mathbb{R}^{v\times v}, k = 1, 2, \cdots, K\}$ and subsequently biases $\{b_k \in \mathbb{R}, k = 1, 2, \cdots, K\}$ are added, each generating a new feature map $\boldsymbol{X}_k$ by an element-wise non-linear transform $\sigma(\cdot)$. The same process is repeated for convolutional layer l,

$$\boldsymbol{X}_k^l = \sigma\left(\boldsymbol{W}_k^l \otimes \boldsymbol{X}^{l-1} + b_k^l\right) \tag{1}$$

where symbol '$\otimes$' denotes the discrete convolution operator, and its specific type of operation has a variety of forms, such as 'valid' convolution, 'same' convolution, 'extra' convolution, strided convolution, fractional-strided convolution, etc.

Another essential layer of CNN is pooling, which is a form of non-linear down-sampling. Convolutional layers are typically alternated with pooling layers where pixel values of neighborhoods are aggregated using some permutation invariant function, usually the max or average operations, which provides another form of translation invariance [46]–[48].

$$\boldsymbol{S}_k^{(l)} = Pooling\left(\boldsymbol{X}_k^{(l)}\right) \tag{2}$$

Finally, after several convolutional and max-pooling layers, the high-level reasoning in the neural network is done via fully connected layers, where weights are no longer shared. CNN is typically trained end-to-end in an entirely supervised manner. The significant reduction in the number of weights parameters and the translational invariance of the learned features contributes to the ability of CNN to be trained end-to-end.

2) CV-CNN

To fully explore the rich information embedded in complex-valued (CV) images, CV-CNN has to be developed [49], [50], [82]. The architecture of a CV-CNN can regard

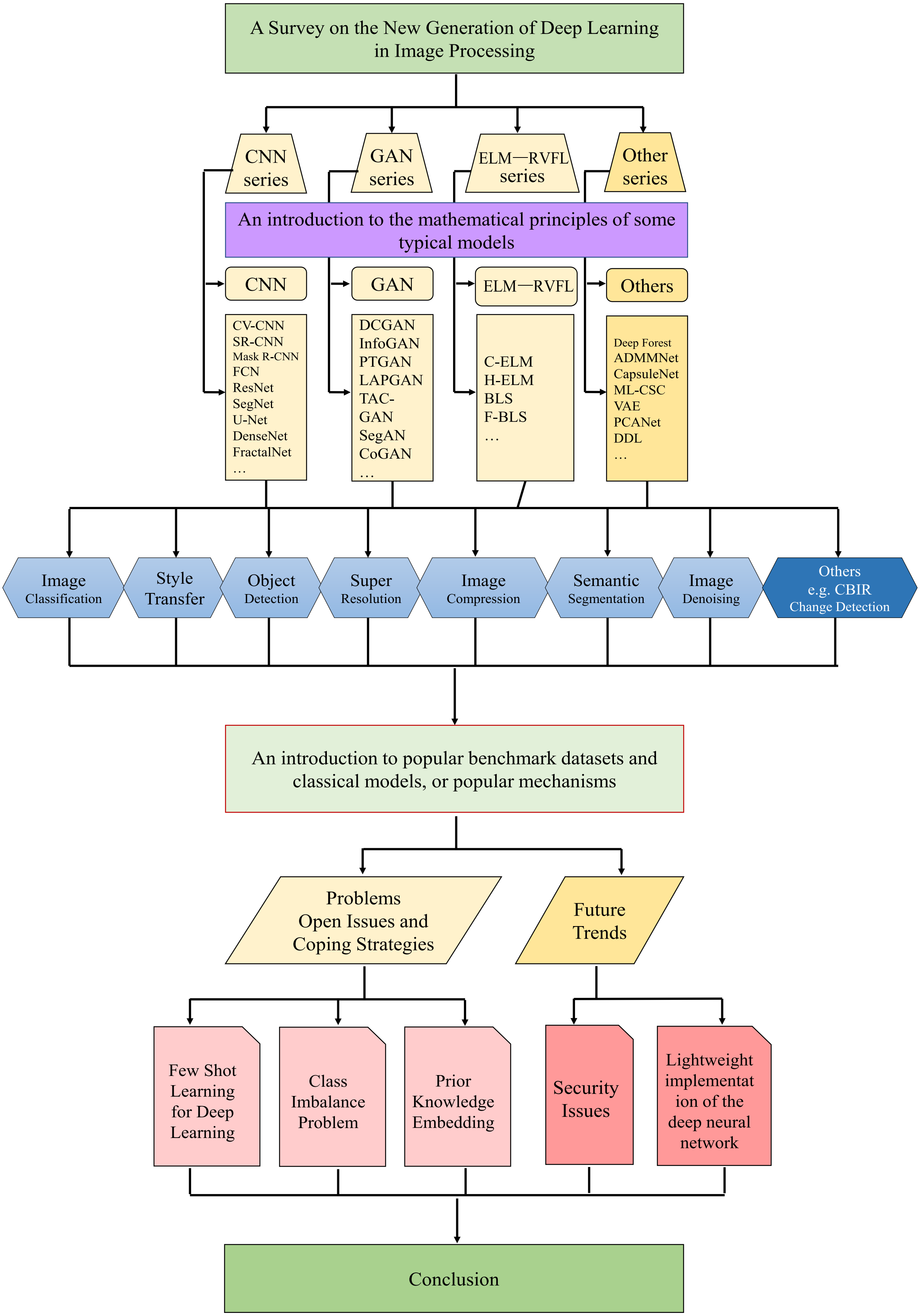


FIGURE 2. The framework structure of survey.

as a variant of the deep neural networks, which not only takes complex data as input but also propagates the phase information through all layers. For CV-CNN, all the layers of the networks, including the convolutional layer, pooling layer, fully connected layer, should be fully CV. The details of each layer in its CV version are presented in the following.

Firstly, for the convlutional layer, the complex output feature maps $\boldsymbol{O}_k^{(l)}$ are computed by the convolution between all the previous layers input feature maps $\boldsymbol{O}^{(l-1)} \in \mathbb{C}^{n\times m}$ and a set of filters $\{\boldsymbol{\Omega}_k^{(l)} \in \mathbb{C}^{v\times v}, k = 1, 2, \cdots, K\}$, and then add a bias $\{\boldsymbol{\gamma}_k^{(l)} \in \mathbb{C}, k = 1, 2, \cdots, K\}$. where '$\mathbb{C}$' denotes the complex domain and the superscript is its dimension.

$$\boldsymbol{O}_k^{(l)} = \sigma\left(\boldsymbol{\Omega}_k^{(l)} \otimes \boldsymbol{O}^{(l-1)} + \gamma_k^{(l)}\right) \tag{3}$$

where $\sigma(\cdot)$ is an element-wise non-linear function, and $\forall z \in \mathbb{C}$. We have

$$\sigma(z) = \sigma\left(\mathcal{R}(z)\right) + j\sigma\left(\mathcal{I}(z)\right) \tag{4}$$

where $z = \mathcal{R}(z) + j\mathcal{I}(z)$. And this CV-convolution operation can be calculated by

$$\begin{aligned}
&\boldsymbol{\Omega}_k^{(l)} \otimes \boldsymbol{O}^{l-1} \\
&= \left(\mathcal{R}(\boldsymbol{\Omega}_k^{(l)}) + j\mathcal{I}(\boldsymbol{\Omega}_k^{(l)})\right) \otimes \left(\mathcal{R}(\boldsymbol{O}_k^{(l-1)}) + j\mathcal{I}(\boldsymbol{O}_k^{(l-1)})\right) \\
&= \left(\mathcal{R}(\boldsymbol{\Omega}_k^{(l)}) \otimes \mathcal{R}(\boldsymbol{O}_k^{(l-1)}) - \mathcal{I}(\boldsymbol{\Omega}_k^{(l)}) \otimes \mathcal{I}(\boldsymbol{O}_k^{(l-1)})\right) \\
&\quad + j\left(\mathcal{R}(\boldsymbol{\Omega}_k^{(l)}) \otimes \mathcal{I}(\boldsymbol{O}_k^{(l-1)}) + \mathcal{I}(\boldsymbol{\Omega}_k^{(l)}) \otimes \mathcal{R}(\boldsymbol{O}_k^{(l-1)})\right)
\end{aligned} \tag{5}$$

Secondly, for the pooling layer,

$$\begin{aligned}
\boldsymbol{V}_k^{(l)} &= Pooling\left(\boldsymbol{O}_k^{(l)}\right) \\
&= Pooling\left(\mathcal{R}(\boldsymbol{O}_k^{(l)})\right) + jPooling\left(\mathcal{I}(\boldsymbol{O}_k^{(l)})\right)
\end{aligned} \tag{6}$$

In other words, pooling layers can regard as sub-sampling layers, and pooling helps to make the representation invariant to small shifts and distortions of the input.

Finally, after several CV convolutional and CV pooling layers, CV fully-connected layers are usually added to act as classification layers. The expected output $\tilde{\boldsymbol{y}}$ and predictive output $\boldsymbol{t}$ can be written as,

$$\begin{cases}
\tilde{\boldsymbol{y}} \triangleq \boldsymbol{y} + j\boldsymbol{y} \in \mathbb{C}^c \\
\boldsymbol{t} = f(\boldsymbol{\mathcal{W}}\boldsymbol{v} + \beta) \in \mathbb{C}^c
\end{cases} \tag{7}$$

where f is the non-linear function, and real-valued vector $\boldsymbol{y} \in \mathbb{R}^c$ is the label of input sample, and CV vector $\boldsymbol{v}$ can be obtained by flattening the CV matrix $\boldsymbol{V}^{(l)}$. If the softmax is applied to CV-CNN, the result is not a probability due to its CV input. Therefore, the final output is the classifier, and the least-squares loss function is adopted in CV-CNN. Experiments show that the classification error can further reduce if employing CV-CNN instead of conventional real-valued CNN with the same degrees of freedom [50], [83].

3) SRCNN

There have been a few studies of using deep learning techniques for image super resolution [84], [85]. Especially, the SRCNN can directly learn an end-to-end mapping between the low-resolution image and high-resolution image [53]. The mapping represents a deep CNN model that consists of three operations: patch extraction and representation, non-linear mapping, reconstruction. At the beginning of detailing each operation, here we only consider a single low-resolution image, and first upscale it to the desired size using bicubic interpolation; and then denote the interpolated image as $\boldsymbol{y} \in \mathbb{R}^{m\times m\times c}$; finally, we expect to recover from $\boldsymbol{y}$ an image $f(\boldsymbol{y})$ that is as similar as possible to the ground truth high-resolution image $\boldsymbol{x} \in \mathbb{R}^{m\times m\times c}$. To keep the following description simple and understandable, we denote $\boldsymbol{y}$ a low-resolution image, and $\boldsymbol{x}$ is the high-resolution image.

Firstly, it can be expressed as an operation for the patch extraction and representation:

$$f_1(\boldsymbol{y}) = max(0, \boldsymbol{W}_1 \otimes \boldsymbol{y} + \boldsymbol{b}_1) \tag{8}$$

where $\boldsymbol{W}_1$ and $\boldsymbol{b}_1$ represent the filters and biases respectively. Specifically, $\boldsymbol{W}_1$ corresponds to n_1 filters of support $p_1 \times p_1 \times c$, where c is the number of channels in the input low-resolution image $\boldsymbol{y} \in \mathbb{R}^{m\times m\times c}$ and p_1 is the spatial size of a filter. The biases $\boldsymbol{b}_1 \in \mathbb{R}^{n_1}$. After applying the ReLU $\left(max(0, \cdot)\right)$ on the filter responses, then the output $f_1(\boldsymbol{y})$ is composed of n_1 feature maps, that is, $f_1(\boldsymbol{y}) \in \mathbb{R}^{m\times m\times n_1}$.

Secondly, for the non-linear mapping, the operation is,

$$f_2(\boldsymbol{y}) = max(0, \boldsymbol{W}_2 \otimes f_1(\boldsymbol{y}) + \boldsymbol{b}_2) \tag{9}$$

where $\boldsymbol{W}_2$ contains n_2 filters of size $p_2 \times p_2 \times n_1$ and $\boldsymbol{b}_2 \in \mathbb{R}^{n_2}$. Without loss of generality, it is possible to add more convolutional layers to increase the non-linearity, but the cost is more training time to increase the complexity of the model. The same with previous operation, that is, $f_2(\boldsymbol{y}) \in \mathbb{R}^{m\times m\times n_2}$.

Thirdly, for the reconstruction, the final high-resolution image can be generated by the following operation,

$$f_3(\boldsymbol{y}) = \boldsymbol{W}_3 \otimes f_2(\boldsymbol{y}) + \boldsymbol{b}_3 \tag{10}$$

where $\boldsymbol{W}_3$ comprise c filters of size $p_3 \times p_3 \times n_2$ and $\boldsymbol{b}_3 \in \mathbb{R}^c$, The same with previous operation, that is, $f_3(\boldsymbol{y}) \in \mathbb{R}^{m\times m\times c}$.

Although the above three operations motivate by different intuitions, and they all lead to the same form as a convolutional layer, so three operations together and form a deep CNN architecture, namely SRCNN. Further, the filtering weights and biases are to be optimized by the following loss function,

$$L(\boldsymbol{\Theta}) = \frac{1}{N}\sum_{n=1}^{N} \|F(\boldsymbol{y}_n, \boldsymbol{\Theta}) - \boldsymbol{x}_n\|_2^2 \tag{11}$$

where N is the number of training samples, and parameters $\boldsymbol{\Theta} = \{\boldsymbol{W}_1, \boldsymbol{W}_2, \boldsymbol{W}_3, \boldsymbol{b}_1, \boldsymbol{b}_2, \boldsymbol{b}_3\}$. We have $F(y, \boldsymbol{\Theta}) \triangleq f_3(\boldsymbol{y})$. Also, using MSE (mean squared error) as the loss function favors a higher PSNR (peak signal-to-noise ratio), and the

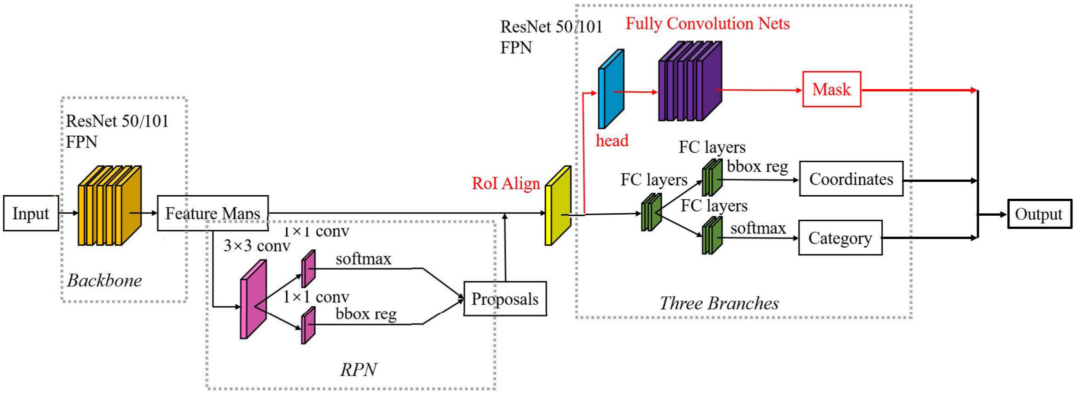


FIGURE 3. The framework of mask R-CNN [55].

loss function is minimized using SGD with the standard BP algorithm [86].

Finally, it is worth pointing out that the ground truth high-resolution images $\{\boldsymbol{x}_n\}$ can prepare as $m \times m \times c$-pixel sub-images randomly cropped from the training images. By applying sub-images means, these samples treated as small images rather than patches. Meanwhile, to synthesize the low-resolution samples $\{\boldsymbol{y}_n\}$, one can blur a sub-image by the Gaussian kernel, sub-sample it by the up-scaling factor, and upscale it by the same factor via bicubic interpolation. We assume that all the convolutional layers have no padding for avoiding the phenomenon of border effects.

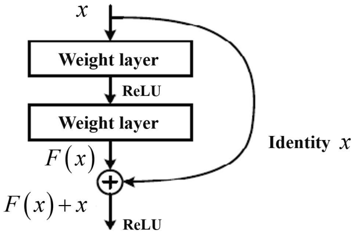


FIGURE 4. Residual learning: A building block [87].

4) RESNET

It is well-known that many other non-trivial visual recognition tasks have also greatly benefited from ultra-deep models. Therefore, a question arises that " is learning better networks as easy as stacking more layers?" With the network layer increasing, it is indisputable that an obstacle is the notorious problem of vanishing or exploding gradients, which can hamper the convergence of networks. Besides, accuracy gets saturated and then degrades rapidly. Unexpectedly, such degradation does not cause by over-fitting. Furthermore, adding more layers to a suitably deep model lead to higher training error. For the degradation problem, The classical ResNet [87] can explicitly let these stacked layers fit a residual mapping rather than a desired underlying mapping. For example, a building block in Fig.4 that has two layers,

Therefore, a building block defined as:

$$y = F(x) + x \tag{12}$$

where x and y are the input and output of the layers considered. The function $F(x)$ represents the residual mapping to be learned, that is,

$$F(x) \triangleq \boldsymbol{W}_2\sigma(\boldsymbol{W}_1x + b_1) + b_2 \tag{13}$$

where σ denotes classical active function ReLU [88].

The operation $F(x) + x$ is performed by a short-cut connection and element-wise addition. It should note that the dimensions of x and $F(x)$ must be the same. If this is not the case, the first choice is that one can perform a linear projection $\boldsymbol{W}_s$ by the short-cut connections to match the dimensions:

$$y = F(x) + \boldsymbol{W}_s x \tag{14}$$

The other choice is that one can still perform identity mapping, but with extra zero entries padded for increasing dimensions was considered. Certainly, we can also use the square matrix in Eq.(12), and it can further make the form of residual function $F(x)$ more flexible. In particular, if the residual function $F(x)$ represents multiple convolutional layers, then the element-wise addition is performed on two feature maps, channel by channel.

More importantly, ResNet structure is simple, which can solves the problem of deep convolution neural network performance degradation under ultra-deep conditions, and its classification performance is excellent. Finally, more tricks include BN right after each convolution and before activation, and one can do not use dropout or max-out.

5) MASK R-CNN

Mask R-CNN is a simple, flexible, and general framework for object instance segmentation [55]. This framework consists of two stages, the first stage, called an RPN, proposes candidate object bounding boxes. The second stage, which extends Faster R-CNN [89] (see black flowchart in Fig.3) by adding a branch for predicting binary segmentation masks on each RoI, in parallel with the current branch for softmax classification and bounding box regression. Significantly, Faster R-CNN does not design for pixel-to-pixel alignment between network inputs and outputs. RoIAlign is a simple

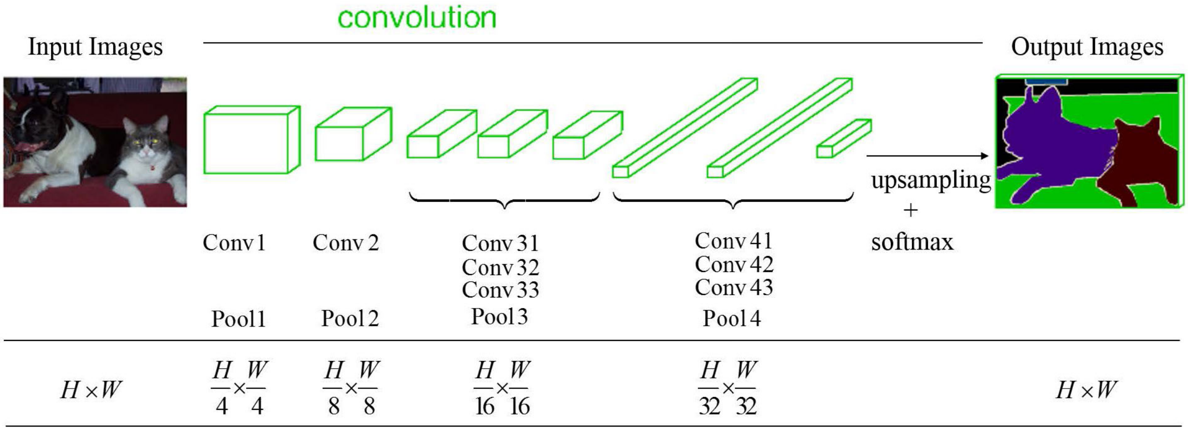


FIGURE 5. **The architecture of FCN for object segmentation [91].**

quantization-free layer for fixing the misalignment, which faithfully preserves exact spatial locations.

Further, during the training phase, a multi-task loss on each sampled RoI can be expressed as follows,

$$L = L_{cls} + L_{box} + L_{mask} \tag{15}$$

where L_{cls} and L_{box} is classification loss and bounding-box loss, respectively. Moreover, the mask branch has a Km^2-dimensional output for each RoI, which encodes K binary masks of resolution $m \times m$, one for each of the K classes. To this, we apply a per-pixel sigmoid and define L_{mask} as the average binary cross-entropy loss. Besides, for an RoI associated with ground-truth class k, L_{mask} is only defined on the k-th mask.

For the Mask R-CNN network architecture in Fig.3, we can differentiate between the convolutional Backbone architecture used for feature extraction over an entire image and the network head for bounding-box recognition (softmax classification and regression) and binary mask prediction that is applied separately to each RoI. Specifically, one can using the nomenclature network-depth-features for the Backbone architecture, such as ResNet networks of depth 50 or 101 layers, and with an FPN [90] backbone extracts RoI features from different levels of the feature pyramid according to their scale. Therefore, Using a ResNet-50/101-FPN backbone for feature extraction captures excellent gains in both accuracy and speed. For the network head, which closely follows architectures presented in previous work to which adds a fully convolutional mask prediction branch; the head part on the ResNet-50/101-FPN that uses fewer filters is more efficient. Finally, RPN can be trained separately and does not share features of Mask R-CNN unless specified.

6) FCN

FCN, a novel deep CNN architecture proposed recently, has achieved excellent performance on pixel levels recognition tasks, such as object segmentation and edge detection [91]. Typical recognition nets, including LeNet, AlexNet, ostensibly take fixed-sized inputs and produce non-spatial outputs. Undoubtedly the fully connected layers of these nets have fixed dimensions and throw away spatial coordinates. Positively, these fully connected layers can also regard as convolutions with kernels.

It is assuming that each layer of data onto a CNN architecture is a three-dimensional array $H \times W \times P$, that is, spatial dimensions $H \times W$ (height and width) and feature or channel dimension P. First, how to convert a fully connected layer to a convolutional layer? For examples, from $H \times W \times P_l$ to $1 \times 1 \times P_{l+1}$, where P_l denotes the number of channel dimension of l-th block or ConvNet layer. Actually, we need P_{l+1} filters of size $H \times W \times P_l$. Second, How to realize up-sampling? The method used is backward strided convolution (sometimes called deconvolution), which can connect coarse outputs to dense pixels via interpolation. For instance, simple bilinear interpolation computes each output from the nearest four inputs by a linear map that depends only on the relative positions of the input and output cells. Thus up-sampling is performed in-network for end-to-end learning by BP from the pixel-wise loss.

Note that the deconvolution filter in such a layer need not fix (e.g., bilinear up-sampling), but one can learn it. For examples, in Fig.5, one can learn deconvolution filter from $\frac{H}{32} \times \frac{W}{32}$ to $H \times W$. Here the number of channel dimensions is neglected, then we can obtain the size of filters is

$$\left(H + 1 - \frac{H}{32}\right) \times \left(W + 1 - \frac{W}{32}\right) \tag{16}$$

In other words, the output size after deconvolution is 32 times the input feature map, which we now call FCN-32s in Fig.6. similarly, one can learn to combine high layer information with low layer information, such as FCN-16s in Fig.6.

7) DESCRIPTION OF OTHER MODELS IN CNN SERIES

U-Net is an improvement based on FCN architecture. Its network architecture consists of two parts: contraction path and extension path [60]. The shrinking path is mainly used to

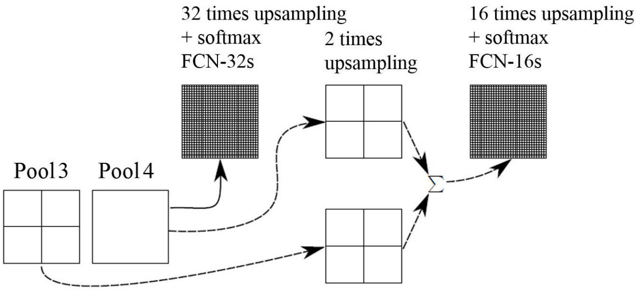


FIGURE 6. Two sampling types: FCN-32s and FCN-16s.

capture the context information about the picture, while the expanding path used to precisely locate the parts that need to the segment of the picture. Compared with FCN, the high-pixel feature extracted by U-Net in contraction path will be combined with the new feature map in the up-sampling process to maximize the retention of some important feature information in the previous down-sampling process. Besides, there is no full connected layer in the whole network, which can minimize the number of training parameters. And the U-shaped structure can better retain the information about the picture. As we all know, classical deep learning needs abundant samples and expensive computing resources; however, U-Net can be used for small sample learning. In particular, this network is suitable for medical-related image segmentation tasks.

Compared with ResNet, DenseNet's innovation lies in the outputs of each layer are connected with all successor layers in a dense block, the feature maps learned by this layer are also passed directly to all the layers behind it as input [24]. Another highlight of DenseNet is efficient for feature reuse, which dramatically reduces network parameters. It is worth pointing out that dense connection directly connects input and loss in each layer, thus alleviating the phenomenon of gradient vanishing. Finally, This new model shows state-of-the-art accuracy with a reasonable number of network parameters for the object recognition tasks.

FractalNet is an advanced and alternative architecture of the ResNet model, which is another efficient for designing large models with nominal depth [63]. Unlike ResNet, FractalNet demonstrates that path length is essential for training ultra-deep neural networks, and residual learning is not necessary for ultra-deep networks. Unlike ResNet, the performance of FractalNet trained by dropout and drop-path tricks often surpasses that of ResNet.

B. GAN SERIES MODELS FOR IMAGE PROCESSING

1) GAN

The GANs framework (see in Fig.7) includes a generative model that captures the sample distribution and a discriminative model that estimates the probability that a sample came from the training (real) sample rather than generating (fake) sample [39], [147]. Based on game theory, one can training GANs requires finding a Nash equilibrium of a non-convex game with continuous, high-dimensional parameters.

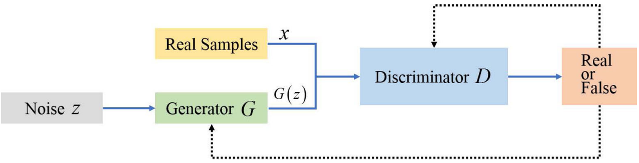


FIGURE 7. The framework of GAN [39], [147].

Generally, training GANs is a problematic issue in practice because of the instability of GANs learning. Several well-designed networks have proposed to overcoming the problem of instability. In Fig.7, We can use differentiable functions D and G to represent the discriminator and the generator, and their inputs are real sample x and random variables z, respectively. The purpose of D is to achieve the correct classification of sample source, while the purpose of G is to make the performance of generated sample $G(z)$ consistent with the performance of the real sample. Therefore, the optimization of GAN can formulate as the following minimax problem:

$$\min_G \max_D V(D, G) = \mathbb{E}_{\boldsymbol{x}\sim P_{\text{data}}}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim P_{\text{noise}}}\Big[\log\big(1 - D(G(\boldsymbol{z}))\big)\Big] \quad (17)$$

The adversarial optimization process improves the performance of D and G gradually. Eventually, when the discrimination ability of D has been improved to an excellent level but cannot discriminate against the sample source correctly, it is thought that the generator G has captured the distribution of a real sample.

Since Ian J Goodfellow proposed GAN in 2014, many GAN variants have produced so far. Undoubtedly, GANs have solved many problems in generative models and inspired other artifical intelligence (AI) methods, but there are still some limitations [148]. For example, from a mathematical point of view, GANs adopt the adversarial learning idea, but the convergence of the model and existence of the equilibrium point have not been proved yet. Besides, the same as generative models based on neural networks, GANs have the common disadvantages of neural networks, such as poor interpretability. Furthermore, although the samples generated by GANs are very different in style, there exists the mode collapse problem (that is, GANs cannot generate continuously changing samples with the change of input noise z). Scientific researchers have put forward many research directions to focus on better solving those drawbacks of GANs. From the perspective of combining GANs with other methods, how to integrate GANs with feature learning, imitation learning, and reinforcement learning to develop new AI applications and promote the development of these methods is very meaningful and hopefully. Below, we mainly introduce the performance of several classic GAN variant models in image processing.

2) DCGAN

A DCGAN firstly introduced convolutional layers to GANs architecture, which can effectively to solve instability of the

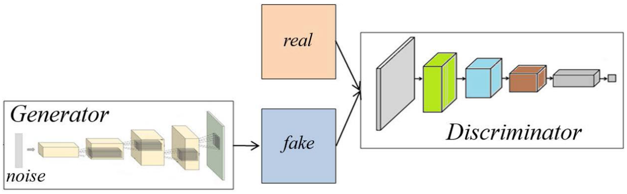


FIGURE 8. The architecture of DCGAN [92].

learning process for the classical GAN (see in Fig.8) [92]. Generally, there are some architecture guidelines for the stability of DCGAN.

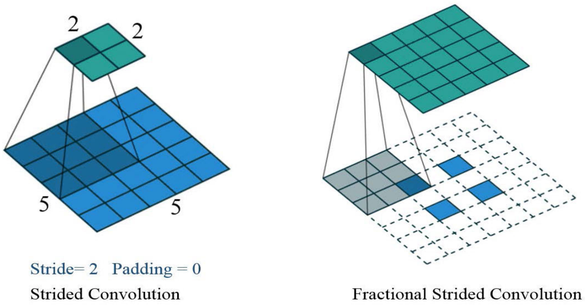


FIGURE 9. Convolution types: strided convolutions and fractional-strided convolutions.

The first is the all convolutional networks that replace deterministic spatial pooling functions (such as max pooling) with strided/fractional-strided convolutions and allowing the network to learn its spatial down-sampling or up-sampling. Concretely, pooling layers in discriminator can replace with strided convolutions, and pooling layers in the generator can replace with fractional-strided convolutions. For example, in Fig.9, we have the following relationship for strided convolutions,

$$\text{output\_size} = \left\lfloor \frac{\text{input\_size-kernel+2}\cdot\text{padding}}{\text{stride}} \right\rfloor + 1 \tag{18}$$

where kernel is the size of filters (here, kernel is 3 in Fig.9), and '$\lfloor\cdot\rfloor$' denotes *floor* function. Obviously, we have the input $x \in \mathbb{R}^{5\times5}$ and the output $y \in \mathbb{R}^{2\times2}$. For fractional-strided convolutions, which can view as the inverse process of strided convolutions. That is, how to implement from y to x ? We need to calculate the new stride and new padding for expanding y,

$$\begin{cases} \text{stride}^{(new)} = 1 \\ \text{padding}^{(new)} = \text{kernel} - 1 \end{cases} \tag{19}$$

and then we have the following relationship for fractional-strided convolutions,

$$\text{output\_size}^{(new)} = \text{stride} \times \left(\text{input\_size}^{(new)} - 1\right) + \text{kernel} \tag{20}$$

where $\text{input}^{(new)}$ is the size of y. More cases can be referred to the relevant literature.

The second is BN, which stabilizes learning by normalizing the input to each unit to have zero mean and unit variance. BN can helps to deal with training problems that arise due to poor initialization and maintains gradient flow in deeper models. Besides, BN can use in other layers both of the generator and the discriminator rather than the generator output layer and the discriminator input layer.

The third is that remove fully connected hidden layers for deeper architectures. The fourth is that the ReLU activation can use in the generator except for the output layer which uses the *Tanh* function,

$$Tanh(t) = \frac{e^t - e^{-t}}{e^t + e^{-t}} \tag{21}$$

and the *Leaky_ReLU* activation function,

$$Leaky\_ReLU(t) = \begin{cases} x, & \text{if } x \geq 0 \\ \alpha x, & \text{if } x \leq 0 \end{cases} \tag{22}$$

is used in the discriminator for all layers, where α is a small constant, it means that the negative axis information will not be completely lost.

3) infoGAN

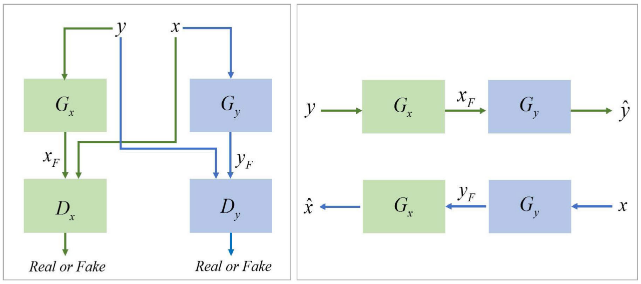
It is well-known that the goal of GAN is to learn a generator distribution that matches the real data distribution [93]. The generator network can generate samples by transforming a noise variable of z. Besides, the noise z can be used by the generator in a highly entangled way, causing the individual dimensions of z to not corresponding to semantic features of the data. In other words, z is not an interpretable expression variable. Furthermore, this is precisely the motivation of InfoGAN to find an interpretable expression, which decomposes the input noise vector into two parts: (1) z, which is treated as source of incompressible noise; (2) $\boldsymbol{c}$, which call the latent code and target the salient structured semantic features of the data distribution.

It is important to identify these latent factors without supervision. Based on information-theoretic regularization, if latent code $\boldsymbol{c}$ is interpretable for generating data $G(z, \boldsymbol{c})$ (here $G(\cdot)$ is the generative model), then there should be high mutual information. In information theory, mutual information between $\boldsymbol{c}$ and $G(z, \boldsymbol{c})$, namely $I(\boldsymbol{c}, G(z, \boldsymbol{c}))$, can be used to measures the amount of information learned from knowledge of variable $\boldsymbol{c}$ about the other variable $G(z, \boldsymbol{c})$. Further, it can be expressed as the difference between two entropy terms,

$$I(\boldsymbol{c}, G(z, \boldsymbol{c})) \triangleq H(\boldsymbol{c}) - H(\boldsymbol{c}|G(z, \boldsymbol{c})) \tag{23}$$

where $H(\boldsymbol{t})$ denotes entropy of $\boldsymbol{t}$. Obviously, $I(\boldsymbol{c}, G(z, \boldsymbol{c}))$ is the reduction of uncertainty in $\boldsymbol{c}$ when $G(z, \boldsymbol{c})$ is observed. For original GAN, the minimax game is given by the following expression:

$$\min_G \max_D V(D, G) = \mathbb{E}_{\boldsymbol{x}\sim P_{\text{data}}}[\log D(\boldsymbol{x})] + \mathbb{E}_{z\sim P_{\text{noise}}}\left[\log\left(1 - D(G(z))\right)\right] \tag{24}$$



(a) Discriminator Loss

(b) Generator Loss

FIGURE 10. The framework of cycleGAN [94].

where $D(\cdot)$ is the discriminative model, and P_{data} and P_{noise} is the real data distribution and noise distribution, respectively. For InfoGAN, we have the following information-regularized minimax optimization object:

$$\min_{G}\max_{D} V_{\text{I}}(D, G) = V(D, G) - \lambda I(\boldsymbol{c}, G(\boldsymbol{z}, \boldsymbol{c})) \tag{25}$$

where λ is the Lagrange multiplier. In practice, $I(\boldsymbol{c}, G(\boldsymbol{z}, \boldsymbol{c}))$ is hard to maximize directly as it requires access to the posterior $P(\boldsymbol{c}|\boldsymbol{x})$. Fortunately we can obtain a lower bound of it by defining an auxiliary distribution $Q(\boldsymbol{c}|\boldsymbol{x})$ to approximate $P(\boldsymbol{c}|\boldsymbol{x})$:

$$\begin{aligned}
&I(\boldsymbol{c}, G(\boldsymbol{z}, \boldsymbol{c}))\\
&\quad = H(\boldsymbol{c}) - H(\boldsymbol{c}|G(\boldsymbol{z}, \boldsymbol{c}))\\
&\quad = \mathbb{E}_{\boldsymbol{x}\sim G(z,c)}[\mathbb{E}_{c'\sim P(\boldsymbol{c}|\boldsymbol{x})}[logP(c'|\boldsymbol{x})]] + H(\boldsymbol{c})\\
&\quad = \mathbb{E}_{\boldsymbol{x}\sim G(z,c)}\Big[\mathbb{E}_{c'\sim P(\boldsymbol{c}|\boldsymbol{x})}[logQ(c'|\boldsymbol{x})]\\
&\qquad + KL(Q(\cdot|\boldsymbol{x})||P(\cdot|\boldsymbol{x}))\Big] + H(\boldsymbol{c})\\
&\quad \geq \mathbb{E}_{\boldsymbol{x}\sim G(z,c)}\Big[\mathbb{E}_{c'\sim P(\boldsymbol{c}|\boldsymbol{x})}[logQ(c'|\boldsymbol{x})]\Big] + H(\boldsymbol{c})
\end{aligned} \tag{26}$$

where $KL(\cdot)$ can measure the difference between two probability distributions. We can define a variational lower bound,

$$L_1(G, Q) \triangleq \mathbb{E}_{\boldsymbol{x}\sim G(z,c)}\Big[\mathbb{E}_{c'\sim P(\boldsymbol{c}|\boldsymbol{x})}[logQ(c'|\boldsymbol{x})]\Big] \tag{27}$$

Hence, InfoGAN is defined as the following minimax game,

$$\min_{G}\max_{D} V_{\text{InfoGAN}}(D, G) = V(D, G) - \lambda L_1(G, Q) \tag{28}$$

Finally, if $\boldsymbol{c}$ is categorical latent code, then $Q(\boldsymbol{c}|\boldsymbol{x})$ can be represented using the non-linear transmission of softmax; if $\boldsymbol{c}$ is continuous latent code, it can be represented by the Gaussian distribution.

4) PTGAN

The goal of PTGAN is to realize the migration of background style under the premise of keeping pedestrian foreground [68]. This person transfer procedure was inspired by the CycleGAN [94] (see in Fig.10). To keep the following description better understanding, we have markings,

$$\begin{cases} x \triangleq x_f + x_b \\ x_f = x \odot M(x) \end{cases} \tag{29}$$

where x_f and x_b is the person foregrounds and backgrounds for x, respectively. and $M(x)$ represents the foreground mask of x. Similarly, we have the same markings for y. How to capture person style transfer from x to y ? that is, we can obtain new image $x_f + y_b$ or $x_b + y_f$. Here, the symbol '$\odot$' denotes hadamard product.

Firstly, for CycleGAN, the objective function of style transfer learning can be formulated as follows:

$$\begin{aligned}
L_{\text{style}} &= L_{GAN}(G_x, D_x, y, x)\\
&\quad + L_{GAN}(G_y, D_y, x, y) + \lambda L_{cycle}(G_x, G_y)
\end{aligned} \tag{30}$$

where $L_{cycle}(G_x, G_y)$ denotes the cycle consistency loss. and we have the following formulation,

$$\begin{aligned}
L_{\text{cycle}}(G_x, G_y) &= \mathbb{E}_{x\sim P(x)}\Big[||G_x(G_y(x)) - x||_1\Big]\\
&\quad + \mathbb{E}_{y\sim P(y)}\Big[||G_y(G_x(y)) - y||_1\Big]
\end{aligned} \tag{31}$$

Secondly, to avoid the appearance of pedestrians in style transfer may change, the objective function of identity (person or foreground) loss can be formulated as follows:

$$\begin{aligned}
L_{\text{ID}} &= \mathbb{E}_{x\sim P(x)}\Big[||(G_y(x) - x) \odot M(x)||_2\Big]\\
&\quad + \mathbb{E}_{y\sim P(y)}\Big[||(G_x(y) - y) \odot M(y)||_2\Big]
\end{aligned} \tag{32}$$

Different transferred samples of one person regard as having the same person. Based on the above discussions, PTGAN can be constructed to satisfy two constraints, i.e., the style transfer and person identity keeping. Therefore, We thus formulate the loss function of PTGAN as follows,

$$L_{\text{PTGAN}} = L_{\text{Style}} + \gamma L_{\text{ID}} \tag{33}$$

where γ is the parameter for the trade-off between two losses. In practice, Extensive experiments show that PTGAN effectively reduces the domain gap.

5) LAPGAN

LAPGAN uses a cascade of CNN within a Laplacian pyramid framework to generate images in a coarse-to-fine processing way [95]. At each level of the pyramid, a separate generative ConvNet model can be trained using the GAN approach. The motivation of LAPGAN is to generate high-quality sample images by taking random vectors as input. furthermore, LAPGAN can construct by combination CGAN (see in Fig.11) and laplacian pyramid framework. CGAN can consider that both the generator and discriminator are conditioned on some extra information $\boldsymbol{y}$, and $\boldsymbol{y}$ could be any auxiliary information, such as class labels or data from other modalities. One can perform the conditioning by feeding $\boldsymbol{y}$ into both the discriminator and generator as the additional input layer. Then the objective function of a two-player minimax game would be formulated as follows,

$$\begin{aligned}
\min_{G}\max_{D} V(D, G) &= \mathbb{E}_{\boldsymbol{x}\sim P_{\text{data}}}[\log D(\boldsymbol{x}|\boldsymbol{y})]\\
&\quad + \mathbb{E}_{\boldsymbol{z}\sim P_{\text{noise}}}\Big[\log\big(1 - D(G(\boldsymbol{z}|\boldsymbol{y}))\big)\Big]
\end{aligned} \tag{34}$$

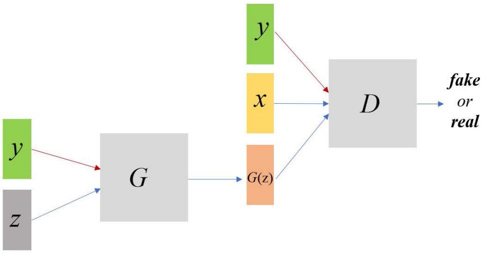


FIGURE 11. The architecture of CGAN.

In the generator, the prior input noise and ***y*** can be combined with joint hidden representation. Moreover, in the discriminator, ***x*** and ***y*** are presented as inputs and to a discriminative function.

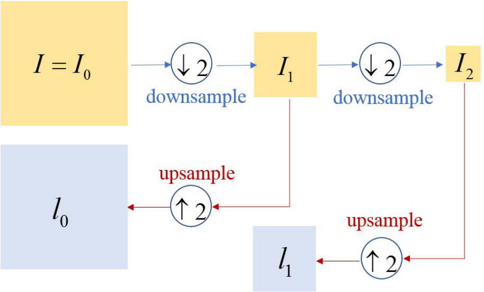


FIGURE 12. Laplacian pyramid structure: downsample and upsample for image ***I*** with Level = 2.

The Laplacian pyramid is a linear invertible image representation. For example, if the number of levels in the pyramid set to 2 and the sample factor equals 2, then we can obtain the following Fig.12 for image I. Intuitively, each level captures image structure present at a particular scale. For a better understanding of LAPGAN, we can define the difference image at each level,

$$h_k = I_k - l_k \tag{35}$$

where k is the number of levels in the pyramid, that is, $k = 0, 1, 2$. Especially, $h_2 = I_2$ is not a difference image, but a low-frequency residual equal to the final pyramid level.

Correspondingly, LAPGAN models at all levels except the final level are conditional generative models that take an upsampled version of the current image l_k as a conditioning variable, in addition to the noise vector z_k. Furthermore, the advantage lies in the independent training of each pyramid level(see in Fig.13). Finally, in the testing phase, the recurrence starts by taking $l_2 = 0$ and using the model at the final level generator G_2 to generate a difference image $\tilde{h}_2$ using noise vector z_2. Obviously, we have $I_2 \approx \tilde{h}_2$ by Eq.(35). Further, l_1 can be approximated by up-sampling $\tilde{h}_2$, that is,

$$l_1 \approx upsample(\tilde{h}_2) \tag{36}$$

where *upsampling* is the process of inserting zero-valued samples between original samples to increase the sampling rate. Similarly, one can use generator G_1 to generate a difference image $\tilde{h}_1$, and by Eq.(35),

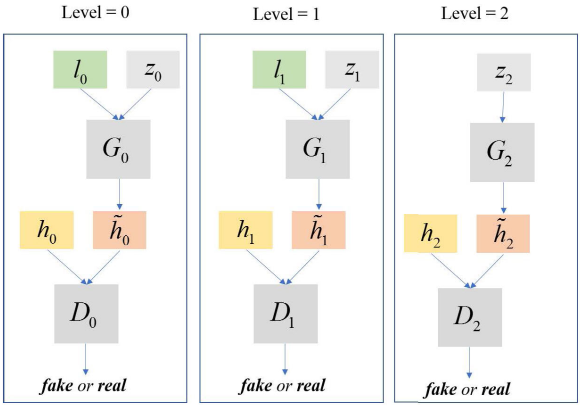


FIGURE 13. The framework of LAPGAN: each level can be viewed as CGAN separately [95].

$$l_0 \approx upsample(\tilde{h}_1 + l_1) \tag{37}$$

Finally, we can use generator G_0 to generate a difference image $\tilde{h}_0$, and the final high-quality image can be obtained by the following formulations,

$$I \approx \tilde{h}_0 + l_0 \tag{38}$$

From the above, LAPGAN can be trained by unsupervised learning.

6) DESCRIPTION OF OTHER MODELS IN GAN SERIES

TAC-GAN is a text-to-image GAN framework for synthesizing images from their text descriptions [69]. TAC-GAN builds upon the conditional auxiliary classifier GANs by conditioning the generated images on a text description instead of on a class label. For the presented TAC-GAN model, the input vector of the Generative network is built based on a noise vector and another vector containing an embedded representation of the textual description. While the Discriminator is similar to that of the TAC-GAN, it is also augmented to receive the text information as input before performing its classification.

SegAN mainly designs for the task of medical image segmentation, which includes two parts: segmentor network and critic network [70]. Unlike classical GAN architecture, they use an FCN as the segmentor to generate segmentation label maps and propose a novel adversarial critic network with a multi-scale L_1 loss function to force the critic and segmentor to learn both global and local features that capture long- and short-range spatial relationships between pixels. Finally, the segmentor and critic networks can be trained in an alternating fashion in a min-max game.

CoGAN is an improvement framework based on the GAN framework, which has established as a viable solution to image distribution learning tasks [71]. Moreover, CoGAN can extend GAN for joint image distribution learning tasks. Precisely, CoGAN consists of a tuple of GAN, each for one image domain. The CoGAN learns a product of marginal distributions rather than a joint distribution.

C. ELM-RVFL SERIES MODELS FOR IMAGE PROCESSING

ELM can provides unified learning solutions for the applications of feature learning, regression, and classification [40]. Its main advantage is the lower computational cost, which is especially important when dealing with many patterns defined in a high-dimensional space.

Furthermore, ELM theories show that hidden neurons are essential but need not iteratively tuned in many types of neural networks. However, due to its shallow architecture, feature learning using ELM may not be useful for complex natural signals, even with a large number of hidden nodes. Various improved ELM networks have proposed to overcoming these shortcomings.

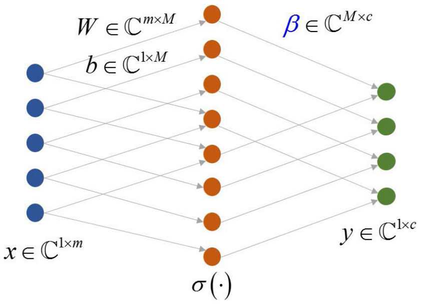

$W \in \mathbb{C}^{m\times M}$
$b \in \mathbb{C}^{1\times M}$
$\beta \in \mathbb{C}^{M\times c}$
$x \in \mathbb{C}^{1\times m}$
$y \in \mathbb{C}^{1\times c}$
$\sigma(\cdot)$

FIGURE 14. The architecture of C-ELM for single-hidden-layer feedforward network.

1) C-ELM

During the past decades, complex-valued neural networks have attracted considerable attention in complex-valued image applications. Naturally, one can also extend the ELM algorithm from the real domain to the complex domain, that is, C-ELM [96]. In Fig.14, $\sigma(\cdot)$ is the complex activation function and $\boldsymbol{\beta}$ is the output weight matrix. There are many fully complex activation functions, such as circular functions,

$$\mathrm{Tan}(z) = \frac{\left(e^{(jz)} - e^{(-jz)}\right)}{j\left(e^{(jz)} + e^{(-jz)}\right)} \tag{39}$$

and hyperbolic functions,

$$\mathrm{Tanh}(z) = \frac{e^{(z)} - e^{(-z)}}{e^{(z)} + e^{(-z)}} \tag{40}$$

where $z \in \mathbb{C}$ and j is the imaginary unit. Further, $\boldsymbol{W}$ and $\boldsymbol{b}$ are randomly choose the complex input weight and the complex bias, respectively. Next, if given complex-valued training samples $\boldsymbol{X} \in \mathbb{C}^{N\times m}$ and $\boldsymbol{Y} \in \mathbb{C}^{N\times c}$, then the hidden layer output matrix $\boldsymbol{H}$ can be formulated as,

$$H = \sigma(\boldsymbol{X}\boldsymbol{W} + \boldsymbol{b}) \in \mathbb{C}^{N\times M} \tag{41}$$

where M is the number of hidden nodes. Finally, we can get the least-squares solution $\boldsymbol{\beta}$ of the linear system,

$$\boldsymbol{Y} = \boldsymbol{H}\boldsymbol{\beta} \tag{42}$$

Without loss of generality, The above equation can also be rewritten as,

$$\begin{pmatrix} \boldsymbol{H}_R & -\boldsymbol{H}_I \\ \boldsymbol{H}_I & \boldsymbol{H}_R \end{pmatrix} \begin{pmatrix} \boldsymbol{\beta}_R \\ \boldsymbol{\beta}_I \end{pmatrix} = \begin{pmatrix} \boldsymbol{Y}_R \\ \boldsymbol{Y}_I \end{pmatrix} \tag{43}$$

where $\boldsymbol{Y}_R$ and $\boldsymbol{Y}_I$ is the real part and imaginary part of $\boldsymbol{Y}$, respectively, other symbols are similar to interpretation.

One can quickly solve the above real linear system by Moore-Penrose pseudo-inverse, and $\boldsymbol{\beta} \triangleq \boldsymbol{\beta}_R + j\boldsymbol{\beta}_I$. C-ELM can complete the learning phase at a breakneck speed and obtain a much lower symbol error rate. In addition, C-ELM encounters the drawback of ELM, that is, the contradiction between the generalization ability of network and the number of hidden nodes.

2) H-ELM

For pattern recognition tasks, feature learning is often required before classification conducted in many applications. However, feature learning using ELM may not be effective for natural signals. To address this issue, inspired by the multilayer perceptron theories and deep stacked auto-encoder networks, H-ELM is proposed [97], [98].

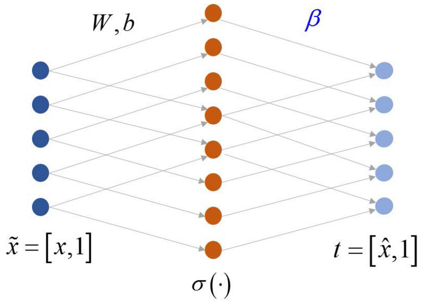


FIGURE 15. The architecture of ELM sparse auto-encoder.

The H-ELM learning framework consists of two main components, one is ELM-based sparse auto-encoder for unsupervised multilayer feature encoding, and the other is that the original ELM applied for final decision making. First, for ELM sparse auto-encoder (see in Fig.15), in order to generate more sparse and compact features of the inputs, we have the following equation,

$$\min_{\boldsymbol{\beta}} \left\|\widetilde{X} - H\boldsymbol{\beta}\right\|_F^2 + \lambda \left\|\boldsymbol{\beta}\right\|_1 \tag{44}$$

where $\boldsymbol{W}$ and b are randomly choose the input weight and the complex bias, and $\widetilde{X} = [X, \mathbf{1}]$, and $\|\cdot\|_F$ is the Frobenius norm, $\|\cdot\|_1$ is the sum of absolute values of of a matrix's all components or elements. If $\boldsymbol{\theta} \triangleq [\boldsymbol{W}; b]$, then we have $H = \sigma(\widetilde{X}\boldsymbol{\theta})$. Once $\boldsymbol{\beta}$ can be obtained, we can replace random parameter $\boldsymbol{\theta}$ with $\boldsymbol{\beta}^T$. Further, we have the hidden mapping matrix with non-randomness, that is,

$$H_\beta = \sigma(\widetilde{X}\boldsymbol{\beta}^T) \tag{45}$$

where H_β can be viewed as an non-randomness alternative to H. Its main advantage lies in an effective feature learning for x. It should be noted that $\hat{x}$ is a approximation for x in Fig.15. Second, H-ELM can be constructed by using the following hierarchical stack way in Fig.16. where L is the number of the hidden layer for multi-layer perceptron.

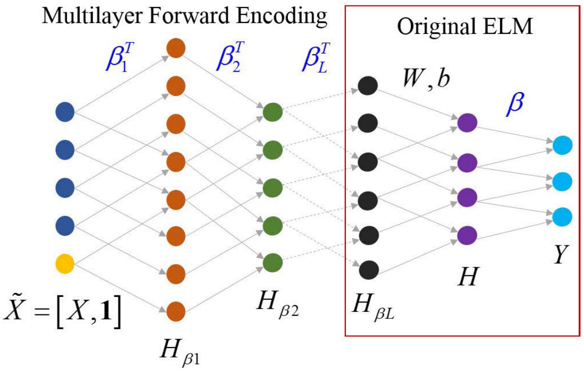


FIGURE 16. **The framework of H-ELM.**

Finally, for the decision making, we have obtained the parameter $\boldsymbol{\beta}$ by original ELM, that is,

$$\boldsymbol{\beta} = H^{\dagger}Y \tag{46}$$

where Y is the target label correspond to input training samples X. And $H^{\dagger}$ is the Moore-Penrose generalized pseudo-inverse of the matrix H. Extensive experiments show that the ELM sparse auto-encoder of H-ELM helps to generate a more excellent performance by providing more robust features.

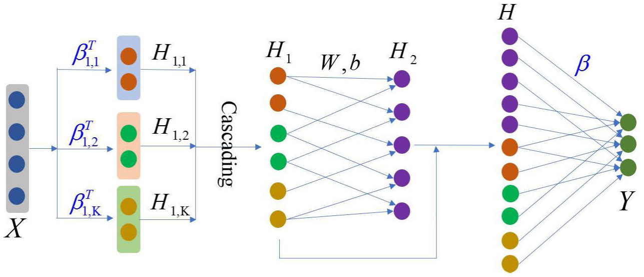


FIGURE 17. **The framework of BLS.**

3) BLS

RVFL can offer a different conventional BP learning method, which can view as the foundation or basic idea of ELM and BLS [99]. The BLS can be expanded broadly when new feature nodes and enhancement nodes are needed, see in Fig.17. Compare to ELM, the main advantage of BLS is that incremental learnings can rapidly update and remodel the system. Moreover, BLS can be applied to an extended network or to a network that only needs to compute the connecting weights of the last layer, such as ELM.

In BLS, to take the advantages of sparse auto-encoder characteristics (the same with ELM sparse auto-encoder), we have $\boldsymbol{H}_{1,k} = \sigma(\tilde{X}\boldsymbol{\beta}_{1,k}^T)$ for the kth window, $k = 1, 2, \ldots, K$. where $\tilde{X} = [X, \mathbf{1}]$ and $\boldsymbol{\beta}_{1,k}$ can be solved by Eq.(47),

$$\min_{\boldsymbol{\beta}} \left\|\tilde{X} - R_{1,k}\boldsymbol{\beta}_{1,k}\right\|_F^2 + \lambda\left\|\boldsymbol{\beta}_{1,k}\right\|_1 \tag{47}$$

where $R_{1,k} = \sigma(X\boldsymbol{W}_{1,k} + b_{1,k})$, and $\boldsymbol{W}_{1,k}$ and $b_{1,k}$ are randomly choose.

In addition, the number of hidden nodes at each window is same. After acquiring $\boldsymbol{H}_{1,k}$, $k = 1, 2, \ldots, K$, we can obtain $\boldsymbol{H}_1$ by cascading operation, that is, $\boldsymbol{H}_1 \triangleq [\boldsymbol{H}_{1,1}, \cdots, \boldsymbol{H}_{1,K}]$. And then we can obtain $\boldsymbol{H}_2$ by the following operation,

$$\boldsymbol{H}_2 = \sigma(\boldsymbol{H}_1\boldsymbol{W} + b) \tag{48}$$

where $\boldsymbol{W}$ and b are randomly choose weight matrix and bias, and σ is non-linear active function. To maintain the information of $\boldsymbol{H}_1$, $\boldsymbol{H}$ can be acquired by cascading $\boldsymbol{H}_2$ and $\boldsymbol{H}_1$. Finally, the parameter $\boldsymbol{\beta}$ can solve by the following linear system,

$$\min_{\boldsymbol{\beta}} \|Y - \boldsymbol{H}\boldsymbol{\beta}\|_F^2 + \lambda\|\boldsymbol{\beta}\|_F^2 \tag{49}$$

That is, $\boldsymbol{\beta} = \boldsymbol{H}^{\dagger}Y$.

Further, for the incremental learning of the dynamic expansion of the BLS model, and simplification BLS model using singular value decomposition, one can refer to relevant literature.

4) F-BLS

The F-BLS can replaces the feature nodes of BLS with a group of Takagi Sugeno fuzzy sub-systems, and the input data are processed by each of them [75]. Instead of aggregating the outputs of fuzzy rules produced by every fuzzy subsystem into one value immediately, all of them are sent to the enhancement layer for further non-linear transformation to preserve the character of inputs. The defuzzification outputs of all fuzzy subsystems and the outputs of the enhancement layer are combined to obtain the model output. The parameters of F-BLS consist of the weights connecting the outputs of the enhancement layer to the final output layer and the coefficients in the following part of fuzzy rules in every fuzzy subsystem, which can be calculated by pseudo-inverse rapidly. Therefore, F-BLS can still retain the fast computational nature of BLS.

D. OTHERS SERIES MODELS FOR IMAGE PROCESSING

Deep learning is rich in connotation. Next, we will continue to introduce several classical frameworks of deep learning. Some of these networks aim at improving the drawbacks of deep neural network. And some networks try to extend the deep learning framework beyond the non-neural network system.

1) DEEP FOREST

The multi-grained cascades Forest (gcForest or Deep Forest [100]), which is a novel decision tree ensemble method, can work well even when there are only small-scale training data. This ensemble method can generate a deep forest

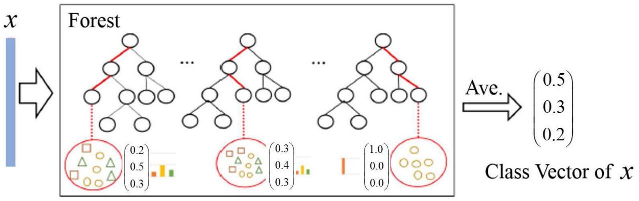


FIGURE 18. The architecture of deep forest: there are three classes for sample x [100].

(see in Fig.18), with a cascade structure which enables gcForest to do representation learning. Also, the main advantage of the deep forest is that the number of cascade levels can be adaptively determined such that the model complexity can be automatically set.

First of all, one of the essential concepts in gcForest is the forest, which can regard as an ensemble classifier consisting of multiple decision trees. For example, suppose there are three classes, then each of the forests will produce a three-dimensional class vector. Moreover, the class vector of sample x can obtain by averaging across all trees class vector for this forest. Here, the number of trees in the forest is a hyperparameter. Further, one can apply different types of forests to encourage diversity, and diversity is crucial for an ensemble framework. For example, gcForest uses two completely random tree forests and two random forests, in which each completely-random tree forest contains 500 completely random trees, and each random forest also includes 500 trees. Therefore, for the cascade forest structure, we have the following illustrations in Fig.19. Here y is the final prediction for input feature vector x. Concatenate operations can be used to avoid excessive attenuation of information with the deepening of the level.

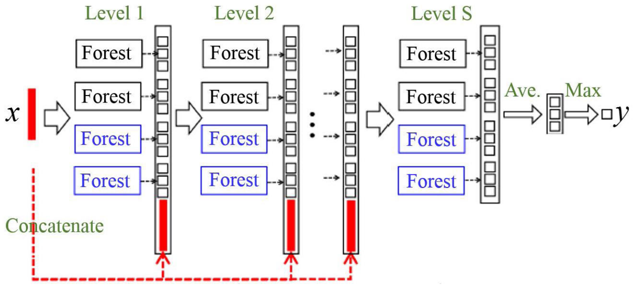


FIGURE 19. The cascade forest structure: there are three classes for sample x at each forest [100].

Further, inspired by the local receptive field of CNN, sliding windows are used to enhance cascade forest. Then we have scanned for input feature vector. For example, suppose there are 100 dimension raw features x and a window size of 10 features is used in Fig.20, then 91 local feature vectors are produced by sliding the window, and the dimension of each local feature vectors is 10, these instances extracted from the same size of windows will be used to train a completely-random tree forest and a random forest, and then the class vectors are generated and concatenated as transformed features.

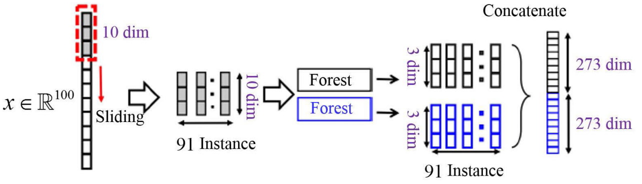


FIGURE 20. Scanning strategy: feature re-representation using the sliding window scanning [100].

Finally, we can construct the overall procedure of gcForest by the above cascade forest structure and scanning. That is, given a feature vector x, corresponding transformed feature representation can obtain by multi-grained scanning procedure, in other words, scanning with multi-sliding window size; and then go through the cascade till the last level, the final prediction will obtain via an average and maximum operation. More importantly, hyperparameters of gcForest include four parts, ones are the number of trees in a forest, second is the number of the different forest for diversity and the number of the forest for scanning, third is the number of levels for cascade forest structure, and fourth is the number of sliding window size. The same with traditional deep neural networks, gcForest can also achieve highly competitive generalization performance. Further, gcForest can also be regarded as a definite attempt of deep learning of non-neural networks.

2) ADMM-NET

ADMM-Net, which is a novel deep network architecture, can derive from the iterative procedures in ADMM algorithm [77], [101]. It is application motivation is that it improves the current magnetic resonance imaging system in reconstruction accuracy and speed.

As a starting point, we introduce ADMM algorithm. Assume x is an image to be reconstructed, and y is the observed image or under-sampled image, then we have the following generalized compressed sensing problem,

$$\min_{x} \frac{1}{2}\|y - \boldsymbol{A}x\|_2^2 + \sum_{l=1}^{L} \lambda_l g(\boldsymbol{D}_l x) \tag{50}$$

where $\boldsymbol{A} \triangleq \boldsymbol{\Psi F}$ is a measurement matrix and Ψ is an under-sampling matrix, $\boldsymbol{F}$ is a Fourier transform. $\boldsymbol{D}_l$ denotes a transform matrix for a filtering operation, $g(\cdot)$ is a regularization function derived from the data prior, such as sparse prior. This optimization problem can be solved efficiently by the ADMM algorithm. Concretely, we use the norm equation for simplicity,

$$\|t_1 - t_2\|_2^2 = \|t_1\|_2^2 + \|t_2\|_2^2 - 2 < t_1, t_2 > \tag{51}$$

and its augmented Lagrangian function is:

$$\min_{x,\boldsymbol{\beta},z} \frac{1}{2}\|y-\boldsymbol{A}x\|_2^2+\sum_{l=1}^{L}\left[\lambda_l g(z_l)+\frac{\rho_l}{2}\|\boldsymbol{D}_l x+\beta_l-z_l\|_2^2\right] \tag{52}$$

where $\boldsymbol{\beta} = [\beta_1, \cdots, \beta_L]$ and $\boldsymbol{z} = [z_1, \cdots, z_L]$. and $\boldsymbol{\rho} = [\rho_1, \cdots, \rho_L]$ are penalty parameters. Further, Eq.(52) can be solved the following three sub-problems:

$$\begin{cases} \min_x \frac{1}{2}\|y-\boldsymbol{A}x\|_2^2+\sum_{l=1}^{L}\left[\frac{\rho_l}{2}\|\boldsymbol{D}_l x+\beta_l-z_l\|_2^2\right] \\ \min_z \sum_{l=1}^{L}\left[\lambda_l g(z_l)+\frac{\rho_l}{2}\|\boldsymbol{D}_l x+\beta_l-z_l\|_2^2\right] \\ \min_{\boldsymbol{\beta}} \sum_{l=1}^{L}\left[\frac{\rho_l}{2}\|\boldsymbol{D}_l x+\beta_l-z_l\|_2^2\right] \end{cases} \tag{53}$$

Substitute $\boldsymbol{A} = \boldsymbol{\Psi F}$ into Eq.(53), then the three sub-problems have the following solutions:

$$\begin{cases} \boldsymbol{X}^{(n)}: x^{(n)} = \boldsymbol{F}^T\left[\boldsymbol{\Psi}^T\boldsymbol{\Psi}+\sum_{l=1}^{L}\rho_l\boldsymbol{F}\boldsymbol{D}_l^T\boldsymbol{D}_l\boldsymbol{F}^T\right]^{-1} \\ \qquad\qquad \times\left[\boldsymbol{\Psi}^T y+\sum_{l=1}^{L}\rho_l\boldsymbol{F}\boldsymbol{D}_l^T\left(z_l^{(n-1)}-\beta_l^{(n-1)}\right)\right] \\ \boldsymbol{Z}^{(n)}: z_l^{(n)} = S\left(\boldsymbol{D}_l x^{(n)}+\beta_l^{(n-1)}, \frac{\lambda_l}{\rho_l}\right) \\ \boldsymbol{\beta}^{(n)}: \beta_l^n = \beta_l^{(n-1)}+\eta_l\left(\boldsymbol{D}_l x^{(n)}-z_l^{(n)}\right) \end{cases} \tag{54}$$

where n denotes n-th iteration, $S(\cdot)$ is a non-linear shrinkage function with parameters λ_l/ρ_l, and η_l is an update rate for updating the multiplier. Here $\boldsymbol{X}^{(n)}$, $\boldsymbol{Z}^{(n)}$ and $\boldsymbol{\beta}^{(n)}$ denotes three types of solutions.

Based on the above iterative algorithm in the ADMM algorithm, Basic ADMM-Net can design by the following steps; the first is that generalizes $\boldsymbol{X}^{(n)}$ to reconstruction layer, the second is that decomposes $\boldsymbol{Z}^{(n)}$ to convolution layer and non-linear transform layer, the third is that extends $\boldsymbol{\beta}^{(n)}$ to multiplier update layer. Concretely, for reconstruction layer, we have rewritten as,

$$\begin{aligned} \boldsymbol{X}_R^{(n)} &: x^{(n)} \\ &= \boldsymbol{F}^T\left[\boldsymbol{\Psi}^T\boldsymbol{\Psi}+\sum_{l=1}^{L}\rho_l^{(n)}\boldsymbol{F}\boldsymbol{H}_l^{(n)T}\boldsymbol{H}_l^{(n)}\boldsymbol{F}^T\right]^{-1} \\ &\quad \times\left[\boldsymbol{\Psi}^T y+\sum_{l=1}^{L}\rho_l^{(n)}\boldsymbol{F}\boldsymbol{H}_l^{(n)T}\left(z_l^{(n-1)}-\beta_l^{(n-1)}\right)\right] \end{aligned} \tag{55}$$

where $H_l^{(n)}$ is the l-th filter with size of $w_f \times w_f$ in the iteration stage n, and replace the fixed $\boldsymbol{D}_l$ in Eq.(54). For convolution layer $\boldsymbol{C}^{(n)}$, we have $\boldsymbol{D}_l^{(n)} \otimes x^{(n)}$ replace $\boldsymbol{D}_l x^{(n)}$, and $l = 1, 2, \cdots, L$. For non-linear transform layer $\boldsymbol{N}^{(n)}$, the output of this layer is defined as,

$$z_l^{(n)} = S\left(c_l^{(n)}+\beta_l^{(n)}\right) \tag{56}$$

where $S(\cdot)$ is a piecewise linear function determined by a set of control points. Finally, for multiplier update layer $\boldsymbol{\beta}_R^{(n)}$, we have

$$\beta_l^n = \beta_l^{(n-1)}+\eta_l^{(n)}\left(c_l^{(n)}-z_l^{(n)}\right) \tag{57}$$

where $\eta_l^{(n)}$ is a parameter to be learned. Compare to convolution flow in CNNs, here new version convolution flow in stage n can be represented as

$$\cdots \rightarrow \boldsymbol{X}_R^{(n)} \rightarrow \boldsymbol{C}^{(n)} \rightarrow \boldsymbol{N}^{(n)} \rightarrow \boldsymbol{\beta}_R^{(n)} \rightarrow \cdots$$

In this novel deep architecture, its purpose to learn parameters includes $\boldsymbol{H}_l^{(n)}$ and $\rho_l^{(n)}$ for reconstruction layer, $\boldsymbol{D}_l^{(n)}$ for convolution layer, and $\eta_l^{(n)}$ in multiplier update layer. Obviously, immediate reconstruction result at each stage can be visualized under each reconstruction layer.

3) CapsuleNet

To breakthrough of scalar input and scalar output restrictions for traditional neurons, an innovative capsule unit was proposed that can output an activity vector represents the instantiation parameters of a specific type of entity such as an object or an object part, which can regard as vector version of neurons. Naturally, each layer of the CapsuleNet [102] is made up of some capsules; its main advantage is to make full use of spatial relations of data.

To keep a clear understanding of the CapsuleNet architecture, a simple CapsuleNet architecture which based on traditional CNN shows in Fig.21, and this architecture is relatively shallow with only two convolutional layers and one fully-connected layer.

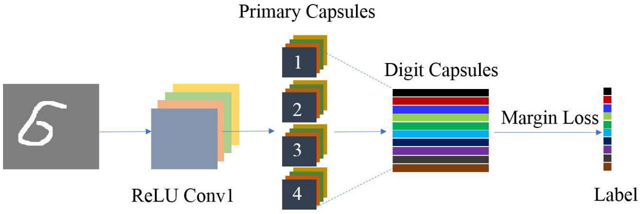


FIGURE 21. The framework of CapsuleNet with three layers [102].

Assume the input $x \in \mathbb{R}^{28\times28}$, and the label $y \in \mathbb{R}^{10}$. Conv1 has 256 convolution kernels (9×9) with a stride of 1 and ReLU activation. Therefore, the output of conv1 can express as,

$$h_i^{(1)} = ReLU(x \otimes \boldsymbol{W}_i^{(1)}) \in \mathbb{R}^{20\times20} \tag{58}$$

where $\boldsymbol{W}_i^{(1)} \in \mathbb{R}^{9\times9}$ denotes i-th convolution kernels of first hidden layer and $i = 1, 2, \cdots, 256$.

Further, the second layer (or Primary Capsules) is a convolutional capsule layer with 4 primary capsules, and each primary capsule contains eight convolutional units with a 9×9 kernel and a stride of 2, that is,

$$h_{j,s}^{(2)} = h^{(1)} \otimes \boldsymbol{W}_{j,s}^{(2)} = \sum_{k=1}^{256} h_k^{(1)} \otimes \boldsymbol{W}_{j,s,k}^{(2)} \in \mathbb{R}^{6\times6} \tag{59}$$

where $\boldsymbol{W}^{(2)}_{j,s,k} \in \mathbb{R}^{9\times 9}$, and subscript k denotes convolution kernels, s means convolutional units and j represents primary capsules. In addition, $s = 1, 2, \cdots, 8$ and $j = 1, 2, 3, 4$. For simplicity, then the output of the second layer can be rewritten as,

$$\boldsymbol{h}^{(2)} \in \mathbb{R}^{4\times 6\times 6\times 8} \tag{60}$$

Meanwhile, primary capsules of the second layer can be viewed as $4 \times 6 \times 6$ capsules, and each capsule is an 8-dimension vector. Without loss of generality, for each capsule, we have,

$$u^{(2)}_s \triangleq \boldsymbol{h}^{(2)}(j, u, v, :) \in \mathbb{R}^8 \tag{61}$$

where $s = j \times u \times v$, and $u, v = 1, 2, \cdots, 6$. Notably, no routing is used between Conv1 and primary capsules. Further, considering the number of categories is 10 for MNIST, so the number of digit capsules is 10. Concretely,

$$h^{(3)}_r = Squash\Big(\sum_s C_{r,s} \boldsymbol{W}^{(3)}_{r,s} u^{(2)}_s\Big) \in \mathbb{R}^{16} \tag{62}$$

where $Squash(\cdot)$ is a non-linear function to ensure that short vectors get shrunk to almost zero length and long vectors get shrunk to a length slightly below 1, and r means the r-th category, $r = 1, 2, \cdots, 10$. Specially, weight matrix $\boldsymbol{W}^{(3)}_{r,s} \in \mathbb{R}^{16\times 8}$, and the parameter $C_{r,s} \in \mathbb{R}$ is coupling coefficient that are determined by the iterative dynamic routing process. To allow for multiple digits, we use a separate margin loss,

$$\begin{aligned} L_r(x) = T_r \max\left(0, m^+ - \left\| h^{(3)}_r \right\|\right)^2 \\ + \lambda(1 - T_r) \max\left(0, \left\| h^{(3)}_r \right\| - m^-\right)^2 \end{aligned} \tag{63}$$

where $T_r = 1$ iff a digit of class k is present, and the parameter $m^+ = 0.9$, $m^- = 0.1$.

Finally, for the input x, the total loss is simply the sum of the losses of all digit capsules, that is,

$$L(x) = \sum_{r=1}^{10} L_r(x) \tag{64}$$

During the training phase, for all input samples $\{x_n\}^N_{n=1}$, then we can solve the correspond parameter by the optimization loss function $\sum_n L(x_n)$. For the testing phase, we can take $\hat{y}(r)$ equals $L_r(x)$, then the final prediction $\hat{y} = [\hat{y}(1), \cdots, \hat{y}(10)]$. Undoubtedly, a simple capsules system already gives unparalleled performance at segmenting overlapping digits is an early indication that CapsuleNet are a direction worth exploring.

4) ML-CSC

ML-CSC [42], which consists of a cascade of convolutional sparse layers, can provides a new interpretation of CNNs. First, we give a concrete example for better understanding CSC model. assume $\boldsymbol{x} \in \mathbb{R}^{25}$ admits a decomposition as $\boldsymbol{D\alpha}$, where $\boldsymbol{\alpha} \in \mathbb{R}^{(25\cdot m)}$ is sparse and the dictionary $\boldsymbol{D} \in \mathbb{R}^{25\times(25\cdot m)}$ has a convolutional structure. Precisely, this dictionary consists of m local n-dimensional filters at every possible location, where m denotes the width of stripe for convolutional structure. For example, $m = 2$ and $n = 5$, then CSC model can be shown in Fig.22, Further, each jth patch $P_j x \in \mathbb{R}^n(\&\mathbb{R}^5)$ from the signal x can be expressed in terms of a shift-invariant local model corresponding to a stripe from the global sparse vector, $S_j\alpha \in \mathbb{R}^{(2n-1)m}(\&\mathbb{R}^{18})$, where P_j can be viewed as patch extraction, S_j is a patch extractor in transform space.

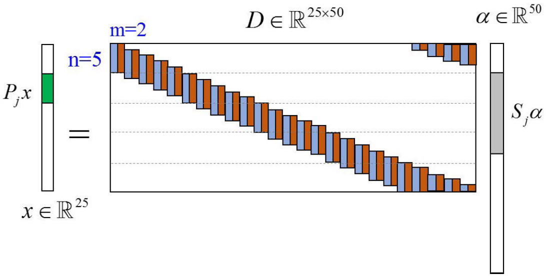

m=2
$D \in \mathbb{R}^{25\times 50}$
$\alpha \in \mathbb{R}^{50}$
n=5
$P_j x$
=
$S_j \alpha$
$x \in \mathbb{R}^{25}$

FIGURE 22. The principle of CSC.

In the context of CSC, the sparsity of the representation is better captured through the $l_{0,\infty}$ pseudo-norm. Formally,

$$\|\boldsymbol{\alpha}\|^s_{0,\infty} \triangleq \max_j \left\| S_j\boldsymbol{\alpha} \right\|_0 \tag{65}$$

Given a convolutional dictionary of appropriate dimension, a signal $\boldsymbol{x}$ admits a representation in terms of the CSC model if satisfy,

$$\boldsymbol{x} = \boldsymbol{D}\alpha, \quad \|\boldsymbol{\alpha}\|^s_{0,\infty} \le K \tag{66}$$

where $K \in \mathbb{N}$ is the sparsity degree.

We can expand CSC model to ML-CSC, that is, given a set of convolutional dictionaries $\{\boldsymbol{D}_l\}^L_{l=1}$ of appropriate dimensions, a signal $\boldsymbol{x}$ admits a representation in terms of the ML-CSC model if satisfy,

$$\begin{cases} \boldsymbol{x} = \boldsymbol{D_1\alpha}_1, & \|\boldsymbol{\alpha_1}\|^s_{0,\infty} \le K_1 \\ \boldsymbol{\alpha}_1 = \boldsymbol{D_2\alpha}_2, & \|\boldsymbol{\alpha_2}\|^s_{0,\infty} \le K_2 \\ \quad\cdots & \\ \boldsymbol{\alpha}_{L-1} = \boldsymbol{D_L\alpha}_L, & \|\boldsymbol{\alpha_L}\|^s_{0,\infty} \le K_L \end{cases} \tag{67}$$

where K_l is the sparsity degree, $l = 1, 2, \cdots, L$, L is the number of layers. Interestingly, the ML-CSC can interpret as a special case of the CSC model. Finally, these sparse coefficients $\boldsymbol{\alpha}_l$ can be solved by the deep coding algorithm [103].

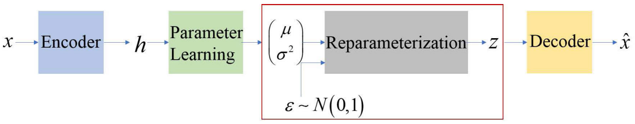


FIGURE 23. The architecture of VAE.

5) VAE

VAE is scalable and powerful generative models for unsupervised representation learning, which can encode a data sample to a latent representation and generate samples from the latent space, respectively [104]. In this framework, see in Fig.23, assumed that the input data set is controlled by a set of latent hidden variables $\boldsymbol{z}$ that are independent of each other and obey a Gaussian distribution $N(\mu, \sigma^2)$, where μ is mean, and σ^2 is variance. If we sampling $\boldsymbol{z}$-variables with $\boldsymbol{z} \sim N(\mu, \sigma^2)$, then BP algorithm cannot proceed through the parameters of the latent distribution.

In practice, VAE can uses a trick of re-parametrization, which can find the following variational approximation,

$$N(\mu, \sigma^2) \approx \mu + \sigma^2 N(0, 1) \tag{68}$$

where $N(0, 1)$ is a standard Gaussian distribution. In other words, if $\varepsilon \sim N(0, 1)$, then $z = \mu + \sigma^2 \cdot \varepsilon$ and $z \sim N(\mu, \sigma^2)$. Therefore, parameters μ and σ^2 can also be iterative updated by BP algorithm. More concisely, we have the following relationship for the encoder and sample,

$$\begin{cases} \boldsymbol{h} = \tanh(\boldsymbol{W}_e x + b_e) \\ \mu = \boldsymbol{W}_\mu \boldsymbol{h} + b_\mu \\ \sigma^2 = \boldsymbol{W}_{\sigma^2} \boldsymbol{h} + b_{\sigma^2} \\ z = \mu + \sigma^2 \varepsilon \end{cases} \tag{69}$$

where $\varepsilon \sim N(0, 1)$, and the dimension of ε is same size to z. For the decoder,

$$\begin{cases} \hat{\boldsymbol{h}} = \tanh\left(\boldsymbol{W}_{\hat{h}} z + b_{\hat{h}}\right) \\ \hat{x} = \tanh(\boldsymbol{W}_d \hat{\boldsymbol{h}} + b_d) \end{cases} \tag{70}$$

Further, The objective function for model optimization is the reconstruction error between x and $\hat{x}$. Generally, reconstruction error can also use cross-entropy or MSE. Here, we can use MSE, that is,

$$L(\theta) = \frac{1}{M} \sum_{i=1}^{M} \left\| x_i - \hat{x}_i(\theta) \right\|_2^2 + \lambda KL(\mu, \sigma^2) \tag{71}$$

where M is the number of training samples, $KL(\mu, \sigma^2)$ is to measure similarity in Eq.(68) (KL divergence essentially estimates how different two probability distributions are) and can be written as,

$$KL(\mu, \sigma^2) = \frac{1}{2}\left(1 + \log \sigma^2 - \mu^2 - \sigma^2\right) \tag{72}$$

The critical takeaway is that a VAE can be trained end-to-end using the classical BP algorithm.

6) PCANet

PCANet is an elementary deep learning network for image classification, which comprises only the fundamental data processing components: cascaded PCA, binary hashing, and block-wise histograms [80]. In the PCANet, PCA is employed to learn multistage filter banks and followed by simple binary hashing and block histograms for indexing and pooling.

Like most ConvNet models, the network hyperparameters such as the number of layers, the filter size, and the number of filters have to given to PCANet. Once the parameters fixed, the training optimization of PCANet is effortless and efficient, for the filter learning in PCANet does not involve regularized parameters and does not require numerical optimization solver.

7) DDL

DDL seeks multiple dictionaries at different scales to capture complementary coherent characteristics [81]. This framework can use for learning a hierarchy of synthesis dictionaries with an image classification goal. Specifically, we can train the dictionaries and classification parameters by a classification objective, and extract the sparse features by reducing a reconstruction loss in each layer. The reconstruction objectives, in some sense, regularize the classification problem and inject source signal information in the extracted features. The performance of the proposed hierarchical method increases by adding more layers, which consequently makes this model easier to tune and adapt. Finally, the DDL algorithm is relatively robust to adversarial perturbation and random noises.

III. FEASIBILITY ANALYSIS OF THE APPLICATION OF NEW GENERATION DEEP LEARNING IN IMAGE PROCESSING

With the improvement of deep learning theories and techniques, the significant progress and revolution have taken place in the field of image processing and computer vision. Although natural images, remote sensing images, and medical images do not share the same structure, deep network classifiers can still successfully extract the semantics. This section aims to provide an overview of the various image application domains where deep learning has garnered much interest.

A. IMAGE CLASSIFICATION

Deep learning thrives with large neural networks and large datasets. One key ingredient for success of deep learning in image classification is the use of convolutional architectures [105]–[109]. For example, deep CNNs has shown state-of-art classification performance on datasets such as ImageNet, a large visual database designed for use in visual object recognition software research. In addition, a dramatic 2012 breakthrough in solving the imageNet challenge [26] is widely considered to be the beginning of the deep learning revolution (see in Fig.24). From then on, many innovative deep frameworks based on CNNs was proposed for imagenet classification tasks, such as AlexNet, GoogleNet, VGG, and ResNet, etc. In other words, ImageNet large scale visual recognition challenge (ILSVRC) greatly promotes the development of deep CNNs.

Meanwhile, scientific researchers have put forward various training techniques one after another. For example, to reduce over-fitting in the globally connected layers, a new regularization method dropout that proved to be very useful. Furthermore, BN is a very effective regularization method that can accelerate the training of large conglomerate networks many times. Further, the obtained higher accuracy can implement by updating the residual module to use identity mappings, and so on.

ELM has demonstrated better generalization performance with extreme fast learning speed in many benchmarks and

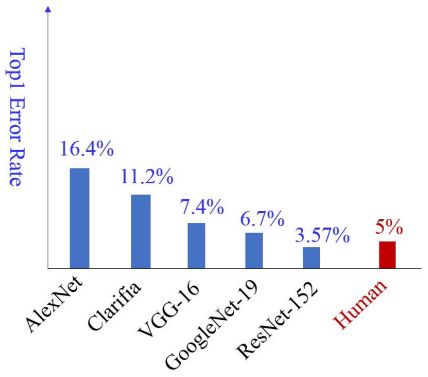


FIGURE 24. Accuracy for ImageNet datasets classification challenge with various classical deep CNN models.

real applications. Compared with the performance of support vector machine (SVM), the classification performance of ELM shows that ELM has better generalization performance with much less training time on majority cases than SVM for both feature extraction methods.

Furthermore, the improvement algorithms, which based on ELM-RVFL, have a significant improvement in classification accuracy rate. For examples, the improvement algorithm H-ELM and BLS not only achieves state-of-art results but also shortens the training time from days (spent by deep learning) to several minutes (by ELM) in MNIST OCR dataset, traffic sign recognition and 3D graphics application, Norb, Fashion MNIST, SVHN, etc. However, on some complex datasets (such as CIFAR 10/100 and ImageNet datasets), ELM series algorithms still need further improvement and combined with the prior of an image to achieve excellent generalization performance, there is still a long way to go.

Though deep neural networks are robust, they have some apparent deficiencies. One is that a massive amount of training data usually required for training deep neural networks, which can disable few-shot learning. The other is that deep neural networks are very complicated models and powerful computational facilities traditionally needed for the training process. More importantly, deep neural networks are with too many hyperparameters, and the learning performance depends seriously on the careful tuning of them [110]. For small-scale datasets, the GAN series can generate more different style samples to promote the generalization performance of the deep discriminative model for classification tasks. At the same time, it also provides a practical solution to the small sample problem. For the enormous hyperparameters, deep forest, which is a novel decision tree ensemble method, can achieve better classification performance by much fewer hyperparameters. However, the deep forest still needs further improvement on more complex datasets. In short, facing a variety of classification applications, a specific deep method is not omnipotent for classification tasks. Two ways motivate the continuous development of deep learning, one is application tasks, and data characteristics, more training tricks (including BN, dropout, early stopping, momentum, and so on) and suitable deep models, such as CV-CNN, and the other is the enhancement of model expressive ability that is independent of data, such as CapsuleNet.

In TABLE 4, We summarize some benchmark datasets and corresponding deep learning methods commonly used in image classification. Compared with the classical models, these new generation models are intended to highlight novel framework/architecture design and relatively excellent generalization performance.

B. STYLE TRANSFER

Rendering the semantic content of an image in different styles is a problematic image processing task [111]. Concretely, transferring the style from one image onto another can be considered a problem of texture transfer. In other words, the output must be semantically similar to the input despite changes in texture for style transfer. The drawback of previous traditional methods (without the support of deep learning) lies in the use of low-level features of target images to inform the texture transfer. Ideally, a style transfer algorithm should be able to extract the semantic image content from the target image and then inform a texture transfer procedure to render the semantic content of the target image in the style of the source image.

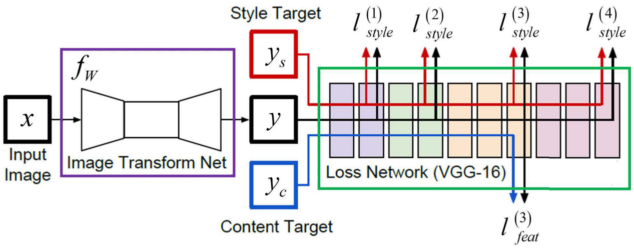


FIGURE 25. Style transfer based on CNNs architecture with image transform networks.

Next, we introduce a framework of style transfer based on the deep CNN (see in Fig.25), concretely, how the generic feature representations learned by CNN can be used to independently process and manipulate the content and the style of natural images. As shown in Fig.25, the framework consists of two components; one is image transform network f_W, the other is loss network (here is VGG-16) that is used to define style/feature loss functions. For style transfer tasks, each input image x correspond to a content target y_c and style target y_s. In addition, the content target y_c is the input image x, that is, $y_c = x$. Generally, the image transform network is a deep ResNet with parameters W, which can transform x into output images y via the mapping $y = f_W(x)$. Further, to measure differences in content and style between y and $y_t = (y_c, y_s)$, feature reconstruction loss l_{feat} and style reconstruction loss l_{style} can be defined. The right half of Fig.25 is a classical VGG-16 network, which is pre-trained on the ImageNet dataset. Further, The image transform network can be trained using stochastic gradient descent to minimize

TABLE 4. **Some classical benchmark datasets and models for image classification.**

Benchmark Datasets	Datasets Description	Datasets Size	Classical model	Some New Models
MNIST	MNIST is one of the most popular deep learning datasets. It is used to try to learn techniques and deep recognition patterns.	10 categories, Training Datasets: 60,000 Testing Datasets: 10,000	LeNet	H-ELM, BLS CapsuleNet Deep Forest
CIFAR-10	This dataset consists of 60000 32x32 colour images in 10 classes, with 6000 images per class.	10 categories, Training Datasets: 50,000 Testing Datasets: 10,000	AlexNet	ResNet FractalNet PCANet
Fashion-MNIST	Researchers intend Fashion-MNIST to serve as a direct drop-in replacement for the original MNIST dataset for benchmarking machine learning algorithms.	10 categories, Training Datasets: 60,000 Testing Datasets: 10,000	HOG+SVM	GoogleNet VGG-16 ResNet-18
SVHN	SVHN is a real-world image dataset for developing machine learning and object recognition algorithms with minimal requirement on data preprocessing and formatting.	10 categories, Training Datasets: 73,257 Testing Datasets: 26032 Additional: 531131	CNN	ResNet BLS C-ELM F-BLS
ImageNet	ILSVRC uses a subset of ImageNet with roughly 1000 images in each of 1000 categories.	1000 categories, Training Datasets: 1200000 Testing Datasets: 150,000 Validation: 50,000	AlexNet	ResNet
Caltech-101	This Datasets intends to facilitate Computer Vision research and techniques and is most applicable to techniques involving image recognition classification and categorization.	101 categories, Each object category contains about 40 to 800 images. Total Images: 9146	LeNet	ResNet CaffeNet
UC Merced Land-Use	The images were manually extracted from large images from the USGS National Map Urban Area Imagery collection for various urban areas around the country. The pixel resolution of this public domain imagery is 1 foot.	21 categories, Each category contains about 100 images. Total Images: 2100	SIFT+SVM [162]	CNN C-ELM CV-CNN
Aerial Image Dataset	Aerial Image Dataset (AID): a large-scale dataset for aerial scene classification. The goal of AID is to advance the state-of-the-arts in scene classification of remote sensing images.	30 categories, Each category contains about 220-420 images. Total Images: 10000	SIFT+SVM	CaffeNet [163] VGG-16 CV-CNN

the following weighted combination of loss functions,

$$\min_{W} J(W) = \frac{1}{4}\sum_{i=1}^{4} \lambda_i l^{(i)}\left(y^{(i)}, y_t^{(i)}\right) \tag{73}$$

where $y^{(i)}$ and $y_t^{(i)}$ is the output image and target image of i-th layer for loss network, respectively. And $l^{(i)}$ is the loss function of i-th layer for loss network, that is,

$$\begin{cases} l^{(i)} \triangleq \left(l_{style}^{(i)}, l_{feat}^{(i)}\right) \\ y_t^{(i)} \triangleq \left(y_{style}^{(i)}, y_{feat}^{(i)}\right) \end{cases} \tag{74}$$

For $i = 0$, we have $y \triangleq y^{(0)} = f_W(x)$ and $y_t \triangleq y_t^{(0)}$. In addition, for the definition of feature reconstruction loss and style reconstruction loss, please refer to the corresponding references.

Unlike the previous style transfer with image transform network, let's introduce a simple style transfer algorithm [112] based on CNN in Fig.26. First, the style image is passed through the Fig.26 network and its style representation on all layers included are computed and stored. The content (or input) image passed through the same network, and the content representation in one layer (such as the fourth layer's feature maps) is stored. Second, a random white noise image also passed through the same network, and its style features and content features can be computed. On each layer included in the style representation, style loss can obtain, and the content loss in one layer can also compute.

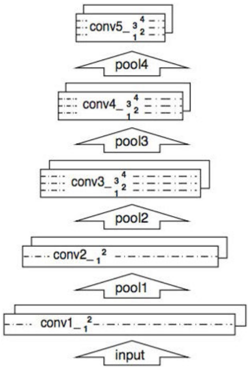


FIGURE 26. **Style transfer based on CNN architecture.**

Finally, the total loss is a linear combination of the content loss and the style loss, and its derivative concerning the pixel values of noise image can be computed using the BP algorithm. This gradient is used to iteratively update the noise image until it simultaneously matches the style features of the style image and the content features of the content image. Indeed, many CNNs frameworks can be used to extract abstract feature maps [113]. Therefore, the scheme of CNN can be used to transfer image style between arbitrary images is feasible.

C. OBJECT DETECTION

Object detection is one of the critical problems to be solved for the development of a complete scene understanding system, which has been recently successfully addressed with deep CNN giving a significant breakthrough [114], [115]. In particular, recent advances are mainly driven by the success of region proposal methods and region-based CNN (R-CNN). Furthermore, the R-CNN can train CNN end-to-end to classify (SVM classifier) the proposal regions into object categories or background, in which detection accuracy depends on the performance of the region proposal module. Further, Fast R-CNN, which is the improvement version of R-CNN, using ultra-deep networks with a softmax classifier to classify the proposal regions.

Generally, the selective search is one of the most popular methods for region proposal module and consumes in much running time as the detection network. Region proposal networks (RPN [89]) is designed to predict region proposals efficiently and to improve region proposal quality. By unifying RPN with Fast R-CNN, the obtained Faster R-CNN can significantly reduce the running time of previous detection networks. Besides, In ILSVRC and COCO 2015 competitions, Faster R-CNN and RPN are the basis of several 1st-place entries in the tracks of ImageNet detection.

Further, Mask R-CNN, which based on Faster R-CNN, can predict binary segmentation mask on each region of interest, was proposed to fix the pixel-to-pixel misalignment between network inputs and outputs for Faster R-CNN. Based on ResNet-101, Mask R-CNN [55] surpasses the winner of the 2016 COCO key-point competition, and achieve an average mask precision of 35.7 and running at 5 fps on the COCO test sets.

As is known to all, the two-stage object detection method represented by the R-CNN series has performed well in terms of accuracy. This kind of network model conducts object detection in two steps. Firstly, all candidate object regions are selected, and then classification and regression are conducted for each candidate region. However, the main disadvantage of those two-stage methods is its slow detection speed. Different from the idea of two-stage target detection, the one-stage method is to divide the region directly on the original image and carry out classification and regression prediction. However, the main disadvantage of the one-stage method is that the positive and negative samples of the bounding box (bbox) are extremely unbalanced, resulting in poor accuracy. To effectively control the ratio of positive and negative samples and prevent the occurrence of imbalance, researchers proposed a new classification loss function Focal loss based on the cross-entropy loss function.

To further verify the validity of Focal loss, researchers further designed RetinaNet [149]. RetinaNet is a single, unified network composed of a backbone network and two task-specific subnetworks, The backbone is responsible for computing a convolutional feature map over an entire input image and is an off-the-self convolutional network. The first subnet performs convolutional object classification on the backbone's output; the second subnet performs convolutional bounding box regression. The structure of the RetinaNet network is very concise. Note that the original intention of researchers is not to innovate the network structure, but to verify the effectiveness of Focal Loss. Experimental results show that RetinaNet can achieve an excellent balance in recognition accuracy and speed.

In TABLE 5, We summarize some benchmark datasets and corresponding deep learning methods commonly used in object detection. Compared with the classical models, these new models are also intended to highlight novel framework/architecture design and relatively excellent generalization performance. In addition to the target detection methods described above, the H-ELM-based fast detection algorithm consists of two parts [97]; one is that a sliding window is used to extract a fixed-size image patch, the other is to design classifier based on H-ELM. It is worth pointing out that this framework can achieve excellent performance on some simple object detection datasets without any additional samples preprocessing. For practical and complex computer vision applications, the ELM series still need to explore the functions of robust feature extraction and classifier.

D. SUPER RESOLUTION

Single image super-resolution (SR) aims to reconstruct a high-resolution (HR) image from one single low resolution (LR) input image. As the pioneer CNN model for SR, SRCNN predicts the non-linear LR-to-HR mapping function via a FCN and significantly outperforms classical non-deep learning methods [53]. However, one fundamental problem remains unsolved mainly: how does one recover the finer texture details when super-resolution at large up-scaling factors?

In recent years, GANs can provide a robust framework for generating plausible-looking natural images with high perceptual quality. Naturally, SRGAN [116] (see in Fig.27), which is a GAN-based network optimized for a new perceptual loss, can recover photo-realistic textures from heavily down-sampled images of public benchmarks. Further, a discriminator network D which can optimize in an alternating manner along with G to solve the following adversarial min-max problem:

$$\min_{\theta_G}\max_{\theta_D}\mathbb{E}_{I^{\mathrm{LR}}\sim P_{\mathrm{train}}(I^{\mathrm{LR}})}\Big[\log\big(1-D_{\theta_D}(G_{\theta_G}(I^{\mathrm{LR}}))\big)\Big] + \mathbb{E}_{I^{\mathrm{HR}}\sim P_{\mathrm{train}}(I^{\mathrm{HR}})}\Big[\log\big(D_{\theta_D}(I^{\mathrm{HR}})\big)\Big] \quad (75)$$

Generally, it allows training a generative model G to fool a differentiable discriminator D that is trained to distinguish SR (or generated) images from real images. Without loss of generality, for training samples I_n^{HR} with corresponding I_n^{LR}, $n = 1, 2, \cdots, N$. θ_G can be solved by the following

TABLE 5. Some classical benchmark datasets and models for object detection.

Benchmark Datasets	Datasets Description	Datasets Size	Classical model	Some New Models
PASCAL VOC	The goal of this datasets is to recognize objects from a number of visual object classes in realistic scenes. The most commonly combination for benchmarking is using 2007 train/val and 2012 train/val for training and 2007 test for validation.	20 categories The train/val data has 11,530 images containing 27,450 ROI annotated objects	R-CNN	Fast R-CNN Faster R-CNN R-FCN Mask R-CNN RefineDet [163]
MS COCO	Objects are labeled using per-instance segmentations to aid in precise object localization. This dataset contains photos of 91 objects types. Object detection on MS COCO are more difficult in comparison to PASCAL VOC.	Dataset contains photos of 91 objects types, a total of 2.5 million labeled instances in 328k images	R-CNN ResNet (bbox)	YOLO SSD CenterNet [165] CornerNet [166] M2Det [167]
ImageNet	ImageNet is an image database organized according to the WordNet hierarchy. About 15 million images, 22,000 categories, each carefully screened and tagged by hand. Task types include: classification, Object localization, Object detection, Scene classification, etc.	For object detection dataset with boundary box, training dataset includes 500,000 images, which belong to 200 types of objects.	OverFeat [168] R-CNN SPP-Net [169]	YOLO FPN [170] Mask R-CNN R-FCN
Open Images	Open Images is a dataset of 9M images annotated with image-level labels, object bounding boxes, object segmentation masks, and visual relationships At present, largest existing dataset with object location annotations.	The training set contains 12.2M bounding-boxes across 500 categories on 1.7M images.	—-	YOLOV3
DOTA	DOTA is a large-scale dataset for object detection in aerial images. It can be used to develop and evaluate object detectors in aerial images. These DOTA images are annotated by experts in aerial image interpretation.	15 object categories, Fully annotated DOTA images contains 188, 282 instances. Image size range from 800 × 800 to 4000 × 4000	R-CNN	Faster R-CNN SSD R-FCN ClusDet [171]
NWPU VHR-10	NWPU VHR-10 dataset is a publicly available10-class geospatial object detection dataset. These ten classes of objects are airplane, ship, storage tank, baseball diamond, tennis court, basketball court, ground track field, harbor, bridge,and vehicle.	10 categories, dataset contains totally 800 very-high-resolution remote sensing images	—-	YOLO YOLT [172]

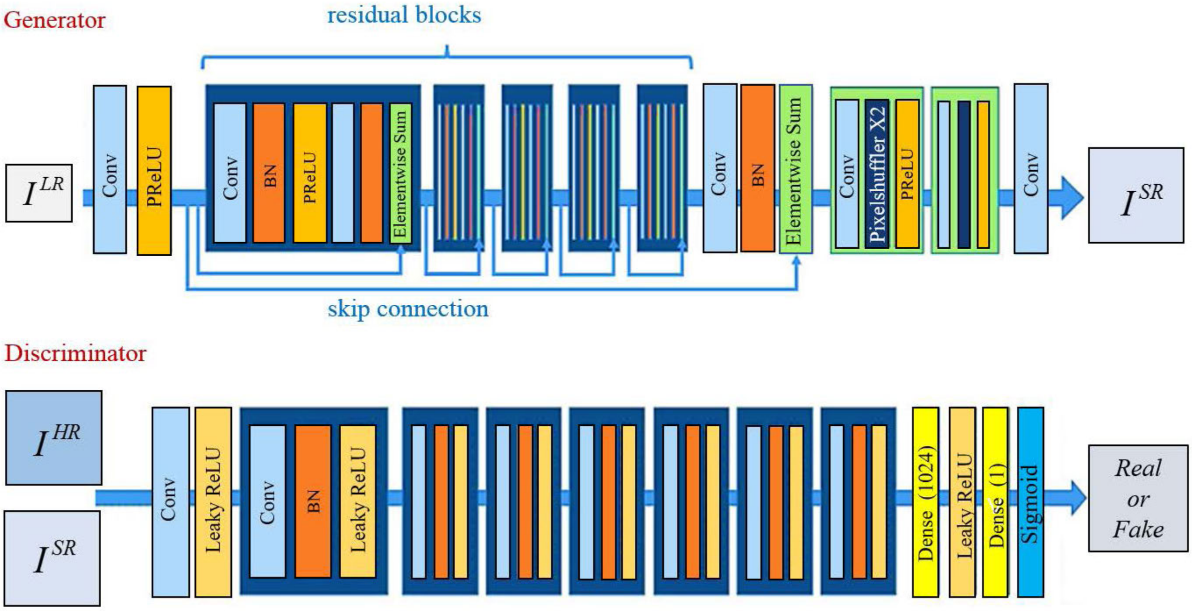


FIGURE 27. The framework of SRGAN for super resolution.

loss function,

$$\min_{\theta_G} \frac{1}{N}\sum_{n=1}^{N} loss^{\text{SR}}\left(I_n^{\text{HR}}, G_{\theta_G}(I_n^{\text{LR}})\right) \tag{76}$$

where $loss^{\text{SR}}$ is perceptual loss function, it can be divided into content loss and adversarial loss. Moreover, the generator model is the core of SRGAN, which illustrated in Fig.27 are residual blocks with an identical layout.

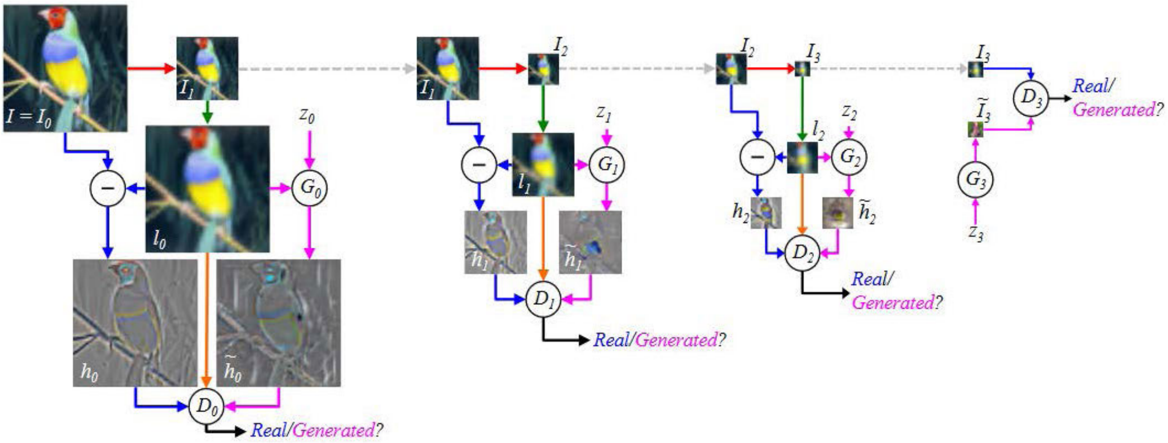


FIGURE 28. The training procedure of LAPGAN [95].

Finally, extensive quantitative and qualitative evaluations on benchmark datasets show that SRGAN can recover the finer texture details when super-resolution at large up-scaling factors.

Besides, based on the idea of CGAN, LAPGAN can generates high-quality pictures based on GAN to solve the problem of poor quality of data generated by GAN [95]. The framework of LAPGAN is illustrated in Fig.28. Starting with a 64 × 64 input image I from training set, we take $I = I_0$ and blur and downsample it by a factor of two to produce I_1, then we can up sample I_1 by a factor of two, giving a low-pass version l_0 of I_0. Further, we compute high-pass $h_0 = I_0 l_0$, h_0 is input to the discriminative model D_0 that computes the probability of it being real vs generated. The same procedure is repeated at scales 1 and 2, using I_1 and I_2. At level 3, I_3 is an 8 × 8 image, simple enough to be modeled directly with a standard GANs G_3 and D_3. Note that the models at each level can be trained separately. Moreover, the most significant difference between the PGGAN [150] and SRGAN and LAPGAN are that the structure of the latter two is fixed, but the structure of PGGAN is continuously changing as the training progresses. The most significant benefit of this is that most of the iterations of PGGAN are completed at a lower resolution, and the training speed is 2-6 times higher than that of traditional GANs. The main advantage of PGGAN is that it can generate high-quality samples. There are many HD pictures in our daily life, the application value of PGGAN is quite a significant promising.

E. IMAGE COMPRESSION

Image compression plays a crucial role in the transmission and storage. To reduce storage requires considerable memory, and degrade transmission requires high bandwidths, previous methods mainly focus on compressive sensing, which is a technology-based on sparse coding. Recently, image compression systems based on deep CNN architecture have become an active area of research. For example, ADMM-Net with convolution and non-linear transform can achieve high reconstruction accuracy for compressive sensing magnetic resonance imaging, and keeping the computational efficiency of the ADMM algorithm [101].

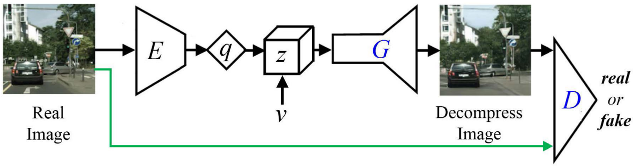


FIGURE 29. Global generative compression based on GAN architecture.

Based on deep CNN, a novel CNN architecture can be designed for semantic perceptual image compression, which can generate a map that highlights semantically-salient regions so that they can encode at a higher quality as compared to background regions. To train a deep compression system with significantly lower bitrates, we can design a GAN framework (see in Fig.29), which is global generative Compression, which can be designed to learn a generative model over images, which viewed as a decoder for image compression [117]. Besides, Global generative Compression can be considered to be a combination of GANs and learned compression. Concretely, with an encoder E and quantizer q, we can encode the real image x to a compressed representation $\boldsymbol{c}$, that is,

$$\boldsymbol{c} = q\big(E(x)\big) \tag{77}$$

Then this latent code $\boldsymbol{c}$ can be concatenated with noise $\boldsymbol{v}$ drawn from a fixed prior distribution $P_{\boldsymbol{v}}$, to form the input vector $\boldsymbol{z} = [\boldsymbol{c}, \boldsymbol{v}]$ of generator G. Further, G can generate an decompress image $\hat{x} = G(z)$ that is consistent with the real image distribution P_x. This process can be expressed by the following loss function,

$$\begin{aligned}\min_{E,G}\max_{D}\, & \mathbb{E}_{x\sim P_x}\big[\log(D(x))\big] \\ & + \mathbb{E}_{\boldsymbol{v}\sim P(\boldsymbol{v})}\big[\log(1 - D(G(z)))\big] \\ & + \lambda\mathbb{E}\big[d(x, G(z))\big] + \beta H(\boldsymbol{c})\end{aligned} \tag{78}$$

where β balances the distortion term against the GAN loss function and entropy terms, and d is a loss that measures how perceptually similar $\hat{x}$ to x, H is entropy to estimate the average number of bits on representation vector $\boldsymbol{c}$. Also, a finite quantizer q given in advance.

Finally, Global generative compression based on GAN can significantly accelerate the convergence of the network, and preserving the overall image content while generating the structure of different scales [118].

F. SEMANTIC SEGMENTATION

Semantic segmentation understands an image at the pixel level, i.e., assign each pixel in the image an object class. For the semantic segmentation, one of the popular initial deep learning approaches was patch classification, where each pixel was separately classified using a patch of the image around it. However, the computational efficiency of this patch-wise training is too low. Furthermore, the required fixed size patch images are also very hindrance. To address these issues, FCN allows segmentation maps to be generated for the image of any size and is also much faster compared to the patch classification approach. The main advantages of FCN framework are that replace fully connected layers with convolutional layer and implement end-to-end training. Almost all the following state of the art approaches on semantic segmentation adopted this design method. Next, we introduce the following two representative improvements on FCN architecture.

First, apart from fully connected layers, one of the main problems with using CNN for segmentation is pooling layers, which can make the where information was discarding. To tackle this issue, SegNet is designed to be an efficient encoder-decoder architecture for pixel-wise semantic segmentation [119]. In Fig.30, for the encoder part, we can discard the fully connected layers in favor of retaining higher resolution feature maps at the deepest encoder output. In addition, each encoder layer has a corresponding decoder layer. Then the final output of SegNet is fed to a softmax classifier to produce class probabilities for each pixel independently. Notably, the main advantage for SegNet is using pooling indices transferred to the decoder to improve the segmentation.

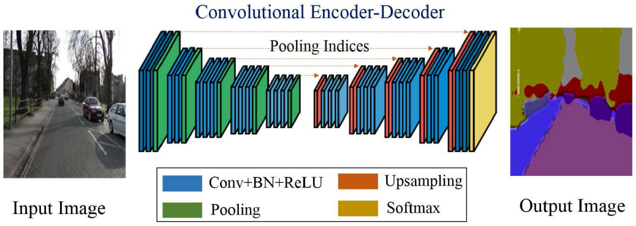


FIGURE 30. The framework of SegNet for object segmentation.

Second, the major disadvantage of FCN is a lack of suitable strategy to utilize global scene category clues. To incorporate appropriate comprehensive features, spatial pyramid pooling can embed in the FCN-based network, the obtained PSPNet [120] can capture both local and global context information to make the final prediction more reliable.

Most deep networks architecture on semantic segmentation use natural image datasets can not directly applicable to biomedical images. The U-Net can achieve outstanding performance on biomedical image segmentation, which can view as the extent of FCN architecture. Notably, the main idea of U-Net is to supplement a usual contracting network by successive layers, where pooling operators can replace by upsampling operators. MOreover, the main advantage of U-Net is that it works with very few training images and yields more precise segmentation [121]. Meanwhile, U-Net applies to various biomedical segmentation tasks.

In TABLE 6, We also summarize some benchmark datasets and corresponding deep learning methods commonly used in semantic segmentation. Compared with the classical models, these new generation models are also intended to highlight novel framework or architecture design and relatively excellent generalization performance.

G. IMAGE DENOISING

Image denoising can be described as the problem of mapping from a noisy image to a noise-free image. The best currently available denoising methods approximate this mapping with cleverly deep learning algorithms. Deep learning technologies can be chosen for image denoising based on the following three reasons. First, deep network architecture can learn more extractions. Second, BN and ReLU can accelerate the training speed of deep networks. Third, deep learning models can train more samples and improve efficiency employing GPU [35].

DnCNNs can use BN and ResNet to perform image denoising [151]. This framework not only deals with blind image denoising, but also addresses image super-resolution task, and JPEG image deblocking. Fig.31 illustrates the architecture of the DnCNN. The input of our DnCNN is a noisy observation of $y = x + z$, where z is additive noise, and x is a clean image. One can adopt the residual learning formulation to train a residual mapping $\mathcal{R}(y) \approx z$, and then we have a clean image of $x = y - \mathcal{R}(y)$. For N noisy-clean training image (patch) pairs $(x_i, y_i)_{i=1}^{N}$, then we have the following optimization object function,

$$J(\theta) = \frac{1}{2N}\sum_{i=1}^{N}\|\mathcal{R}(y,\theta) - (y_i - x_i)\|_F^2 \tag{79}$$

where θ denotes the trainable parameters in DnCNNs, experiments demonstrate that DnCNN can exhibit high effectiveness in several general image denoising tasks.

Many other typical methods also obtain excellent performance for image denoising. For example, the fusion of the dilated convolution and ResNet is used for image denoising, and this framework is fit for combing disparate sources of experts of image denoising [152]. FFDNet can uses noise level map and noisy image as an input to deal with different

TABLE 6. Some classical benchmark datasets and models for semantic segmentation.

Benchmark Datasets	Datasets Description	Datasets Size	Classical model	Some New Models
KITTI	KITTI is one of the most popular datasets for use in mobile robotics and autonomous driving, and itself does not contain ground truth for Semantic Segmentation. Various researchers have manually annotated parts of the datasets to fit their necessities.	Road detection Challenge : Three class Ground truth for 323 images Tracking Challenge : Ten object categories Annotated 253 images	FCN	SegNet U-Net SfM-Net [173] Up-CNN [174]
Cityscapes	Cityscapes is a recently released dataset for semantic urban scene understanding. It contains 5,000 high quality pixel-level annotated images collected from 50 cities in different seasons.	Dataset contains 19 categories. Training images: 2975 Validation: 500 Testing: 1525	FCN CRF-RNN [175]	DeepLab [176] PSPNet [177] ENet [178]
PASCAL VOC	PASCAL VOC 2012 segmentation dataset contains 20 object categories and one background class. This dataset is arguably the most popular for semantic segmentation.	21 classes categorized Training images: 1464 Validation images: 1449 Test images: Private	FCN CRF-RNN	ParseNet DeepLab Dilation [178]
Stanford Background	The Stanford Background Dataset is a new dataset introduced in Gould et al. (ICCV 2009) for evaluating methods for geometric and semantic scene understanding. The dataset contains 715 images chosen from existing public datasets	Images to be of outdoor scenes (320 × 240 pixels) contain one foreground object, and have the horizon position within the image.	FCN	rCNN [179] 2D-LSTM [181]
NYUDv2	Dataset consists 1149 indoor RGB-D images captured with Kinect device. Its really useful for certain robotic tasks at home. However, its relatively small scale hinder its application for deep learning architectures.	40 indoor object classes Training images: 795 Testing images: 654	FCN	LSTM-CF [182] DeepLab SegNet
ShapeNet Part	ShapeNet Part is a subset of ShapeNet repository which focuses on fine grained 3D object segmentation. ShapeNet is an ongoing effort to establish a richly-annotated, large-scale dataset of 3D shapes.	16 categories, dataset contains 31693 meshes. Each shape class is labeled with 2 or 5 parts.	—-	PointNet [183]

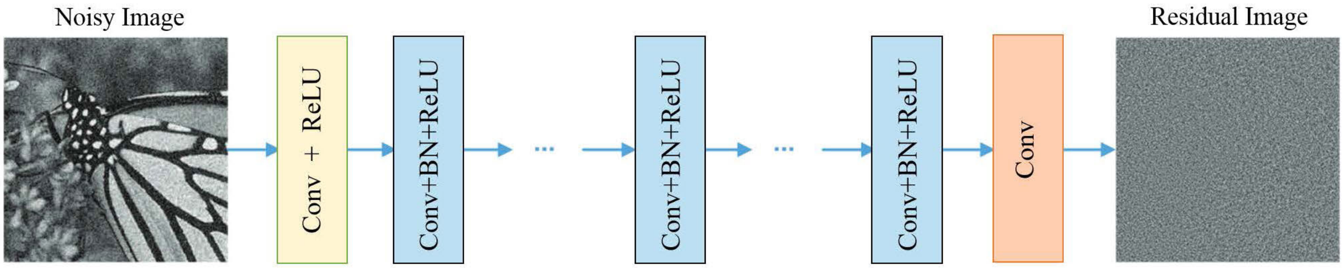


FIGURE 31. The architecture of DnCNNs [151].

noise levels [153]. IRCNN can fuses the model-based optimization method and CNN to perform image denoising, which can deal with different inverse problems and multiple tasks with one single-mode [154].

H. OTHER IMAGE PROCESSING TASKS

In addition to the above image processing tasks, we will briefly introduce some of the successful applications of deep learning technology.

First, let us start with the image retrieval task. CBIR is one of the fundamental research challenges extensively studied in the multimedia community for decades. However, the semantic gap issues that exist between low-level image pixels captured by machines and high-level semantic information perceived by a human is still one of the most challenging problems in current CBIR research. Among various techniques, deep CNN has been actively investigated as a promising direction to bridge the semantic gap in the long term [122]. On the one hand, deep CNN can obtain useful high-level feature representations of images using large-scale network architecture. On the other hand, a deep CNN model on classification or similarity loss function can be retrained easily since the features extracted by the pre-trained deep CNN model may not be better than the traditional hand-crafted features [123].

Second, for the change detection, it means two images of the same area can be captured at two different time instances, and then these two images are processed to identify the changes part, the output is a binary change map, which indicates the location of the changes. Generally, the goal of change detection is to detect significant changes while rejecting unimportant ones by preprocessing operations (i.e., geometric and radiometric adjustments). In recently, Deep learning has a powerful ability to learn abstract features. In particular, as the unsupervised deep learning methods successively proposed, such as GAN, VAE, ML-CSC, therefore researching the application of deep learning techniques to change detection has become a promising research direction [124], [125].

Third, for face recognition, that is a relatively mature application direction for deep learning. In particular, CNN has achieved promising results in face recognition recently. From deep hidden identity feature (deepID [126]) to faceNet [127], CNN plays a very important role in feature extraction. On the widely used Labelled Faces in the Wild dataset, faceNet can even achieve a new record recognition accuracy of 99.63%.

IV. CONCLUSION

A. OPEN ISSUES AND COPING STRATEGIES

In practice, the lack of large training datasets with labels has been repeatedly mentioned for various image processing application tasks. Therefore this is one of a challenge to apply large-scale deep learning algorithms [128]. Also, the class imbalance is a common problem that has comprehensively studied in classical machine learning, yet minimal systematic analysis is available in deep learning. Generally, the class imbalance can degrade the generalization performance of deep learning. Furthermore, under the specific application tasks, the prior or spatial correlation of image has not been fully utilized. As a result, the robustness of the deep network is weak. Below, we mainly discuss the practical solutions to these problems and the research directions of the deep models from the following three aspects.

1) FEW-SHOT LEARNING FOR DEEP LEARNING

In general, when the amount of training images with labels is not large enough, the deep networks can effectively avoid over-fitting phenomena using the following standard four methods [128].

- The first is data augmentation; we can manually increase the size of the training images; that is, one can produce a batch of new image sets from the available small training images by shift, rotation, noise addition, flipping, color jittering, and random crop.
- The second is regularization; over-fitting can be suppressed by adding a regularization term after the loss function. However, the disadvantage is the introduction of a hyper-parameter that requires manual adjustment.
- The third is a dropout; in essence, this is also a regularization method, which can be achieved by randomly zeroing the output of some neurons in a hidden layer.
- The fourth is adopt the unsupervised layer-wise pre-training and fine-tuning; that is, layer-wise unsupervised pre-training can achieve by using AE or RBM, then adding classification layer to perform supervised end-to-end fine tuning [129], [130].

Apart from the above four commonly used methods, more optimization designs framework for deep learning have also been proposed and applied to few-shot learning. Below, we introduce some of the most representative network frameworks.

- The first is the deep generative model, which is a powerful way of learning any data distribution function using unsupervised learning, which can generate a new image with some variations. Two of the most commonly used and efficient approaches are VAE and GAN. Notably, these two approaches have achieved tremendous success in just a few years. However, an open problem is still worth exploring is how to generate more detailed and high-quality images under the premise of guaranteeing network convergence.
- The second is a recursive cortical network (RCN) [131], which is a probabilistic and-or graph model in essence. RCN can acquire the learning ability on small samples by hierarchical and production modeling. Concretely, through the separation modeling of the edges and planes of objects and the hierarchical modeling of complex changes such as textures and scales, the whole model has a strong ability of generalization and robustness to changes in appearance. Naturally, the combination of probabilistic and graph models and deep networks may be a promising research direction.
- The third is transfer learning, which focuses on storing knowledge gained while solving one problem and applying it to a different but related problem [132], [133]. For example, we have a classification task in one domain of interest, but sufficient training image sets in another domain. If transfer learning can be trained successfully, then significantly improve the performance of learning by avoiding many expensive samples. Generally, transfer learning works in two similar domains, and the performance is excellent. However, the standard of measuring the difference between the two domains has not improved. Nevertheless, this does not seem to affect its combination with deep learning may become the next research hotspot.
- The fourth is bayesian CNN [134], which placing a probability distribution over the CNN's kernels, can offer better robustness to over-fitting on small sample learning.

2) CLASS IMBALANCE FOR DEEP LEARNING

Classification of imbalanced data sets is a significant research problem as many real-world image sets have skewed class distributions in which most images belong to a few majority classes, and the minority classes contain a limited amount of other image sets [135]. Many approaches have been proposed for tackling this issue, and these approaches can be roughly divided into two categories,

- The first is data manipulation techniques that target changing the data distribution to make data sets less imbalanced. For example, the most common approach is the sampling, which can operate on the data itself to increase its balance. However, over-sampling can quickly introduce undesirable noise with over-fitting risks, and under-sampling is often preferred to remove valuable information [136].
- The second is algorithm/model-oriented approaches, which aim to develop new learning mechanisms to work for imbalanced images datasets. For example, cost-sensitive learning, which can assign higher misclassification costs to the minority class than to the majority. More common class imbalanced methods in machine learning can refer to literature [137], [138].

Although deep learning has reached great success in many research topics, as mentioned earlier, very few approaches have been made to target it for imbalanced image data. Undoubtedly, applying deep learning directly on imbalanced images datasets may result in poor performance. Below, we introduce four approaches that may help improve the performance of deep learning in class imbalance.

- The first is a deep generative model, which can generate more new images with some variations from the minority classes. For example, HexaGAN, a generative adversarial network framework that shows promising classification performance for class imbalance problem [155]. However, the convergence of deep generative models based on minority class image datasets becomes an insurmountable challenge.
- The second is the deep forest, which can give full play to the superiority of its basic unit decision tree in class imbalance. For example, class weights random forest can be assigned individual weights for each class instead of a single weight [156]. The validation test on UCI data sets demonstrates that for imbalanced medical data, class weights random forest enhanced the overall performance of the classifier while producing high accuracy in identifying both majority and minority class.
- The third is to combine deep CNN with bootstrapping strategy, and during the bootstrapping process, a set of pseudo balanced training batches are generated based on the properties of the data set and fed into the deep CNN for classification [139].
- The fourth is active learning, which is a more efficient alternative to resampling methods to form a balanced training image datasets. Naturally, a model that combines deep learning with active learning can be one of the most promising direction for class imbalance [140], [141].

3) PRIOR KNOWLEDGE FOR DEEP LEARNING

With the disappearance of data dividends, deep learning has increasingly demonstrated its limitations, particularly in relying on large-scale image datasets with labels and the inability to use prior knowledge effectively, et., these limitations hinder the further development of deep learning [142]. How to effectively apply a large amount of prior knowledge/rules? How to make the result of the deep learning model consistent with prior knowledge [143]? Based on the prior knowledge or rules is what bridges the gap between large neural networks and relatively small datasets, these problems have gradually become one of the mainstream research directions.

Actually, there are many ways to incorporate prior knowledge into deep neural networks [144], [145]. The simplest type of prior knowledge often used is weight decay, which assumes the weights come from a normal distribution with zero mean and some fixed variance. Then this type of prior knowledge is added as an additional term to the loss function. Especially, weight decay can view as the same as the Bayesian approach to modeling prior knowledge. Similar to weight decay, it is possible to construct other loss terms that penalize mappings contradicting our domain knowledge. However, integrating prior knowledge into deep learning is not always easy, the main reason is that the main description of knowledge representation is not an abstract quantitative feature, but a relationship between features [146]. Also, deep learning places too much emphasis on the independence of the system and exclude general prior knowledge. Therefore, deep learning combined with prior knowledge/rules of the image may become a breakthrough in solving small sample problems, but this promising direction still needs further research.

B. FUTURE TRENDS OF DEEP LEARNING IN IMAGE PROCESSING

Although deep learning has achieved excellent performance in some image processing applications, it still needs considerable effort to tackle the openness mentioned above problems. In particular, it would be precious to develop a safe and efficient deep learning-based image processing system. In the process of image processing and analysis, security issues related to deep learning seem to have rarely mentioned. The security problem here refers to the degradation of generalization performance caused by the vulnerability of the model to several malicious attacks [157]. Of course, there are many kinds of malicious attacks, such as forging data sets that are not consistent with the image representation, tampering with the statistical characteristics of some hidden layer parameters, abusing the evaluation index of the model, and so on. It is worth pointing out that the security problem of deep learning is not unrelated to the black box characteristics of this model. Once someone attacked the deep learning system, the possibility of this system repair will decrease [158]. We expect that security issues will become an essential ingredient and future trend for designing a deep learning system in image processing [159].

For complex image processing tasks, simple models do not seem to be effective, and useful models are often not simple. Effective models here means that the model can obtain relatively excellent generalization performance, and simple models imply that the capacity of the model is relatively low. For example, ResNet can quickly get more than 90% or even higher test accuracy in CIFAR 10/100 classification task but deep stacked auto-encoder network can only hover around 60% [160]. It is generally accepted that the ResNet has a more extensive model capacity than the deep stacked auto-encoder network. Here, the concept of model capacity involves the number of parameters of the model, the topological structure of the model, and the complexity of the model. Note that model complexity is not the only factor that can determine the performance difference between the two deep learning models. It should also include parameter optimization learning mode, training skills, and parameter initialization. However, the quantification of model capacity is still a potential research direction in the future. Then the research result will help us to summarize the models related to deep learning in specific image processing tasks. Further, similar to the wavelet base library, the deep learning library can help us to make an appropriate model selection.

Since it is usually expensive to obtain sufficient annotated data in several image processing tasks, unsupervised deep learning have been considered as promising research directions at all time. The recent breakthroughs in unsupervised deep learning like the GAN series can provide a gateway to harness the massive amount of unlabelled image datasets. In more image compression processing tasks, there is a higher requirement for the reversibility of the system, which is different from the deep learning model, including GAN. Therefore, the development of the reversible deep learning model is still a potential research direction in the field of unsupervised learning in the future.

C. CONCLUSION

Nowadays, deep learning has dramatically promoted the research progress of image processing. Correspondingly, various applications related to image processing are also helping the rapid development of deep learning in all aspects of network structure, layer design, and training tricks. Therefore, the development of deep learning in image processing is in the ascendant, and there is still a vast space for the future of AI. However, the deeper structure makes BP algorithm more difficult. At the same time, the scale of training images without labels is also rapidly increasing; this urgently requires more new deep models and a new parallel computing system to more effectively interprets the content of the image and form a suitable analysis mechanism. In this context, we have summarized a new generation of deep learning methods used in image processing, but also presents the dedicated discussion on open challenges, unsolved problems, and potential future trends. There are a large number of new developing deep learning techniques and emerging deep models each year, here, we provide a comprehensive framework for comprehensive understanding towards the critical aspects of this field, clarify the most important advancements and shed some light on future studies. More importantly, this survey aims to help or arouse other researchers to catch a glimpse of the state-of-the-art deep learning methods in the field of image processing and facilitate the applications of these deep learning technologies in their research tasks. By further studying the relationship between deep learning and image processing applications, it can not only help us understand the reasons for the success of deep learning but also inspire new deep models and training methods. We also hope that our theoretical understanding of the properties of deep learning will continuously improve in the nick of time, as it currently lags far behind the practice.

REFERENCES

[1] H. Jiang, Q. Tian, J. Farrell, and B. A. Wandell, "Learning the image processing pipeline," *IEEE Trans. Image Process.*, vol. 26, no. 10, pp. 5032–5042, Oct. 2017.

[2] F. Erden, S. Velipasalar, A. E. Cetin, and A. Z. Alkar, "Sensors in assisted living: A survey of signal and image processing methods," *IEEE Signal Process. Mag.*, vol. 33, no. 2, pp. 36–44, Mar. 2016.

[3] M. Koohzadi and N. M. Charkari, "Survey on deep learning methods in human action recognition," *IET Comput. Vis.*, vol. 11, no. 8, pp. 623–632, Dec. 2017.

[4] X. Huang, E. Uffelman, O. Cossairt, M. Walton, and A. K. Katsaggelos, "Computational imaging for cultural heritage: Recent developments in spectral imaging, 3-D surface measurement, image relighting, and X-ray mapping," *IEEE Signal Process. Mag.*, vol. 33, no. 5, pp. 130–138, Sep. 2016.

[5] G. AlRegib, M. Deriche, Z. Long, H. Di, Z. Wang, Y. Alaudah, M. A. Shafiq, and M. Alfarraj, "Subsurface structure analysis using computational interpretation and learning: A visual signal processing perspective," *IEEE Signal Process. Mag.*, vol. 35, no. 2, pp. 82–98, Mar. 2018.

[6] N. Ganatra and A. Patel, "A survey on diseases detection and classification of agriculture products using image processing and machine learning," *Int. J. Comput. Appl.*, vol. 180, no. 13, pp. 7–12, 2018.

[7] A. Madabhushi and G. Lee, "Image analysis and machine learning in digital pathology: Challenges and opportunities," *Med. Image Anal.*, vol. 33, no. 6, pp. 170–175, 2016.

[8] J. A. Richards, J. Xiuping, and J. A. Richards, *Remote Sensing Digital Image Analysis: An Introduction*. Berlin, Germany: Springer, 2006.

[9] X. Qian, J. Wang, S. Guo, and Q. Li, "An active contour model for medical image segmentation with application to brain CT image," *Med. Phys.*, vol. 40, no. 2, 2013, Art. no. 021911.

[10] A. Krizhevsky and G. Hinton, "Learning multiple layers of featuresfrom tiny images," Dept. Comput. Sci., Univ. Toronto, Toronto, ON, Canada, 2009, pp. 1–60, vol. 1, no. 1.

[11] A. Ahar, A. Barri, and P. Schelkens, "From sparse coding significance to perceptual quality: A new approach for image quality assessment," *IEEE Trans. Image Process.*, vol. 27, no. 2, pp. 879–893, Feb. 2017.

[12] H. Li, Y. Li, and F. Porikli, "DeepTrack: Learning discriminative feature representations online for robust visual tracking," *IEEE Trans. Image Process.*, vol. 25, no. 4, pp. 1834–1848, Apr. 2016.

[13] Y.-B. Sheng and L. Zhou, "Distributed secure quantum machine learning," *Sci. Bull.*, vol. 62, no. 14, pp. 1025–1029, 2017.

[14] Z. Y. Ran and B. G. Hu, "Parameter identifiability in statistical machine learning: A review," *Neural Comput.*, vol. 29, no. 5, pp. 1151–1203, 2017.

[15] B. S. C. Wade, S. H. Joshi, B. A. Gutman, and P. M. Thompson, "Machine learning on high dimensional shape data from subcortical brain surfaces: A comparison of feature selection and classification methods," *Pattern Recognit.*, vol. 63, pp. 731–739, Mar. 2017.

[16] R. Salakhutdinov, J. B. Tenenbaum, and A. Torralba, "Learning with hierarchical-deep models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1958–1971, Aug. 2013.

[17] H.-C. Shin, M. R. Orton, D. J. Collins, S. J. Doran, and M. O. Leach, "Stacked autoencoders for unsupervised feature learning and multiple organ detection in a pilot study using 4D patient data," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1930–1943, Aug. 2013.

[18] B. Chen, G. Polatkan, G. Sapiro, D. Blei, D. Dunson, and L. Carin, "Deep learning with hierarchical convolutional factor analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 887–901, Aug. 2013.

[19] G. Ditzler, R. Polikar, and G. Rosen, ''Multi-layer and recursive neural networks for metagenomic classification,'' *IEEE Trans. Nanobiosci.*, vol. 14, no. 6, pp. 608–621, Sep. 2015.

[20] Z. Fan, D. Bi, L. He, M. Shiping, S. Gao, and C. Li, ''Low-level structure feature extraction for image processing via stacked sparse denoising autoencoder,'' *Neurocomputing*, vol. 243, no. 2, pp. 12–20, 2017.

[21] G. E. Hinton, S. Osindero, and Y.-W. Teh, ''A fast learning algorithm for deep belief nets,'' *Neural Comput.*, vol. 18, no. 7, pp. 1527–1554, 2006.

[22] Y. Bengio, ''Learning deep architectures for AI,'' *Found. Trends Mach. Learn.*, vol. 2, no. 1, pp. 1–127, 2009.

[23] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, ''Backpropagation applied to handwritten zip code recognition,'' *Neural Comput.*, vol. 1, no. 4, pp. 541–551, 1989.

[24] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, ''Densely connected convolutional networks,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, vol. 1, no. 1, pp. 2261–2269.

[25] 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, Nov. 1998.

[26] A. Krizhevsky, I. Sutskever, and G. E. Hinton, ''ImageNet classification with deep convolutional neural networks,'' in *Proc. NIPS*, vol. 1. 2012, pp. 1097–1105.

[27] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, ''ImageNet: A large-scale hierarchical image database,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, vol. 1, no. 1, pp. 248–255.

[28] J. Edstrom, Y. Gong, D. Chen, J. Wang, and N. Gong, ''Data-driven intelligent efficient synaptic storage for deep learning,'' *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 64, no. 12, pp. 1412–1416, Dec. 2017.

[29] T. Nitta, ''On the singularity in deep neural networks,'' in *Neural Information Processing*. Cham, Switzerland: Springer, 2016.

[30] S. Krig, ''Feature learning and deep learning architecture survey,'' in *Computer Vision Metrics*. Cham, Switzerland: Springer, 2016.

[31] R. Ren, T. Hung, and K. C. Tan, ''A generic deep-learning-based approach for automated surface inspection,'' *IEEE Trans. Cybern.*, vol. 48, no. 3, pp. 929–940, Mar. 2018.

[32] F. Xing, Y. Xie, H. Su, F. Liu, and L. Yang, ''Deep learning in microscopy image analysis: A survey,'' *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 10, pp. 4550–4568, Oct. 2017.

[33] H. Petersson, D. Gustafsson, and D. Bergstrom, ''Hyperspectral image analysis using deep learning—A review,'' in *Proc. IEEE Int. Conf. Image Process. Theory, Tools Appl.*, Dec. 2017, vol. 1, no. 1, pp. 1–6.

[34] M. I. Razzak, S. Naz, and A. Zaib, ''Deep learning for medical image processing: Overview, challenges and future,'' 2017, *arXiv:1704.06825*. [Online]. Available: https://arxiv.org/abs/1704.06825

[35] C. Tian, Y. Xu, L. Fei, and K. Yan, ''Deep learning for image denoising: A survey,'' 2018, *arXiv:1810.05052*. [Online]. Available: https://arxiv.org/abs/1810.05052

[36] D. Li, ''A tutorial survey of architectures, algorithms, and applications for deep learning,'' *APSIPA Trans. Signal Inf. Process.*, vol. 1, no. 26, pp. 3–32, 2014.

[37] A. Kamilaris and F. X. Boldú, ''Deep learning in agriculture: A survey,'' *Comput. Electron. Agricult.*, vol. 147, pp. 70–90, Apr. 2018.

[38] Q. Wang, X. Li, and D. Xu, ''An improved deep learning framework brief-net based on convolutional neural networks,'' *ICIC Express Lett.*, vol. 11, no. 8, pp. 1323–1330, 2017.

[39] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, ''Generative adversarial nets,'' in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, vol. 3, no. 1, pp. 2672–2680.

[40] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, ''Extreme learning machine: Theory and applications,'' *Neurocomputing*, vol. 70, nos. 1–3, pp. 489–501, 2006.

[41] Y.-H. Pao, G.-H. Park, and D. J. Sobajic, ''Learning and generalization characteristics of the random vector functional-link net,'' *Neurocomputing*, vol. 6, no. 2, pp. 163–180, 1994.

[42] J. Sulam, V. Papyan, Y. Romano, and M. Elad, ''Multilayer convolutional sparse modeling: Pursuit and dictionary learning,'' *IEEE Trans. Signal Process.*, vol. 66, no. 15, pp. 4090–4104, Aug. 2018.

[43] Q. Zhang, L. T. Yang, and Z. Chen, ''Deep computation model for unsupervised feature learning on big data,'' *IEEE Trans. Services Comput.*, vol. 9, no. 1, pp. 161–171, Jan./Feb. 2016.

[44] K. Nguyen, C. Fookes, and S. Sridharan, ''Improving deep convolutional neural networks with unsupervised feature learning,'' in *Proc. IEEE Int. Conf. Image Process.*, Sep. 2015, vol. 1, no. 1, pp. 2270–2274.

[45] A. Dosovitskiy, P. Fischer, J. T. Springenberg, M. Riedmiller, and T. Brox, ''Discriminative unsupervised feature learning with exemplar convolutional neural networks,'' *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 9, pp. 1734–1747, Sep. 2016.

[46] T. Wiatowski and H. Bölcskei, ''A mathematical theory of deep convolutional neural networks for feature extraction,'' *IEEE Trans. Inf. Theory*, vol. 64, no. 3, pp. 1845–1866, Mar. 2018.

[47] X. Shen, X. Tian, A. He, S. Sun, and D. Tao, ''Transform-invariant convolutional neural networks for image classification and search,'' in *Proc. ACM Multimedia Conf.*, 2016, vol. 1, no. 1, pp. 1345–1354.

[48] D. Yu, H. Wang, P. Chen, and Z. Wei, ''Mixed pooling for convolutional neural networks,'' in *Proc. 9th Int. Conf. Rough Sets Knowl. Technol.*, 2014, vol. 1, no. 1, pp. 364–375.

[49] Z. Zhang, H. Wang, F. Xu, and Y.-Q. Jin, ''Complex-valued convolutional neural network and its application in polarimetric SAR image classification,'' *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 12, pp. 7177–7188, Dec. 2017.

[50] C. A. Popa, ''Complex-valued convolutional neural networks for real-valued image classification,'' in *Proc. IEEE Int. Joint Conf. Neural Netw.*, May 2017, vol. 1, no. 1, pp. 816–822.

[51] R. Haensch and O. Hellwich, ''Complex-valued convolutional neural networks for object detection in PolSAR data,'' in *Proc. Eur. Conf. Synth. Aperture Radar*, Jun. 2010, vol. 1, no. 1, pp. 1–4.

[52] J. Gao, B. Deng, Y. Qin, H. Wang, and X. Li, ''Enhanced radar imaging using a complex-valued convolutional neural network,'' *IEEE Geosci. Remote Sens. Lett.*, vol. 16, no. 1, pp. 35–39, Jan. 2019.

[53] C. Dong, C. C. Loy, K. He, and X. Tang, ''Image super-resolution using deep convolutional networks,'' *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 2, pp. 295–307, Feb. 2015.

[54] K. Umehara, J. Ota, and T. Ishida, ''Application of super-resolution convolutional neural network for enhancing image resolution in chest CT,'' *J. Digit. Imag.*, vol. 31, no. 4, pp. 441–457, 2018.

[55] K. He, G. Gkioxari, P. Dollár, and R. Girshick, ''Mask R-CNN,'' *IEEE Trans. Pattern Anal. Mach. Intell.*, to be published.

[56] Z. Huang, Z. Zhong, L. Sun, and Q. Huo, ''Mask R-CNN with pyramid attention network for scene text detection,''2018, *arXiv:1811.09058*. [Online]. Available: https://arxiv.org/abs/1811.09058

[57] K. Zhao, J. Kang, J. Jung, and G. Sohn, ''Building extraction from satellite images using mask R-CNN with building boundary regularization,'' in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2018, vol. 1, no. 1, pp. 242–2424.

[58] Q. Zhang, Z. Cui, X. Niu, S. Geng, and Y. Qiao, ''Image segmentation with pyramid dilated convolution based on ResNet and U-Net,'' in *Proc. Int. Conf. Neural Inf. Process.*, 2017, vol. 1, no. 1, pp. 364–372.

[59] H. Ren, M. El-Khamy, and J. Lee, ''DN-ResNet: Efficient deep residual network for image denoising,'' 2018, *arXiv:1810.06766*. [Online]. Available: https://arxiv.org/abs/1810.06766

[60] 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.*, 2015, vol. 1, no. 1, pp. 1–10.

[61] A. Sevastopolsky, ''Optic disc and cup segmentation methods for glaucoma detection with modification of U-net convolutional neural network,'' *Pattern Recognit. Image Anal.*, vol. 27, no. 3, pp. 618–624, Jul. 2017.

[62] J. Stawiaski, ''A pretrained densenet encoder for brain tumor segmentation,'' 2018, *arXiv:1811.07542*. [Online]. Available: https://arxiv.org/abs/1811.07542

[63] L. Yu, X. Yang, and J. Qin. *3D FractalNet: Dense Volumetric Segmentation for Cardiovascular MRI*. [Online]. Available: Accessed: 2016. http://appsrv.cse.cuhk.edu.hk/lqyu/papers/MICCAI16 HVSMR.pdf

[64] C. Jia, L. I. Weihua, and L. I. Xiaochun, ''High-resolution remote sensing image segmentation based on weight adaptive fractal net evolution approach,'' *Remote Sens. Land Resour.*, vol. 25, no. 4, pp. 22–25, 2013.

[65] J. Dai, Y. Li, K. He, and J. Sun, ''R-FCN: Object detection via region-based fully convolutional networks,'' 2016, *arXiv:1605.06409*. [Online]. Available: https://arxiv.org/abs/1605.06409

[66] R. Yeh, C. Chen, T. Y. Lim, M. Hasegawa-Johnson, and M. N. Do, ''Semantic image inpainting with perceptual and contextual losses,'' 2016, *arXiv:1607.07539v2*. [Online]. Available: https://arxiv.org/abs/1607.07539v2

[67] A. Spurr, E. Aksan, and O. Hilliges, ''Guiding InfoGAN with semi-supervision,'' 2017, *arXiv:1707.04487*. [Online]. Available: https://arxiv.org/abs/1707.04487

[68] L. Wei, S. Zhang, W. Gao, and Q. Tian, "Person transfer GAN to bridge domain gap for person re-identification," 2017, *arXiv:1711.08565*. [Online]. Available: https://arxiv.org/abs/1711.08565

[69] A. Dash, J. C. B. Gamboa, S. Ahmed, M. Liwicki, and M. Z. Afzal, "TAC-GAN—Text conditioned auxiliary classifier generative adversarial network," 2017, *arXiv:1703.06412*. [Online]. Available: https://arxiv.org/abs/1703.06412

[70] Y. Xue, T. Xu, H. Zhang, L. R. Long, and X. Huang, "SegAN: Adversarial network with multi-scale L_1 loss for medical image segmentation," *Neuroinformatics*, vol. 16, nos. 3–4, pp. 383–392, 2018.

[71] M. A. Kiasari, D. S. Moirangthem, and M. Lee, "Coupled generative adversarial stacked auto-encoder: CoGASA," *Neural Networks*, vol. 100, pp. 1–9, Apr. 2018.

[72] R. V. Babu and S. Suresh, "Fully complex-valued ELM classifiers for human action recognition," in *Proc. Int. Joint Conf. Neural Netw.*, Jul./Aug. 2011, vol. 1, no. 1, pp. 2803–2808.

[73] M. Ramanathan, Y. W. Yau, and E. K. Teoh, "Human posture detection using H-ELM body part and whole person detectors for human-robot interaction," in *Proc. Int. Conf. Hum. Agent Interact.*, 2016, vol. 1, no. 1, pp. 239–242.

[74] Y. Shi, Y. Wei, D. Pan, W. Deng, H. Yao, T. Chen, G. Zhao, and M. Tong, and Q. Liu, "Student body gesture recognition based on Fisher broad learning system," *Int. J. Wavelets, Multiresolution Inf. Process.*, vol. 17, no. 1, 2019, Art. no. 1950001.

[75] F. Shuang and C. L. P. Chen, "Fuzzy broad learning system: A novel neuro-fuzzy model for regression and classification," *IEEE Trans. Cybern.*, to be published.

[76] M. Li, Z. Ning, B. Pan, S. Xie, X. Wu, and Z. Shi, "Hyperspectral image classification based on deep forest and spectral-spatial cooperative feature," in *Proc. Int. Conf. Image Graph.*, 2017, pp. 325–336. [Online]. Available: http://link.springer.com/chapter/10.1007

[77] Y. Yan, H. Li, Z. Xu, and Z. Xu, "Deep ADMM-Net for compressive sensing MRI," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, vol. 12, no. 3, pp. 10–18.

[78] K. Duarte, Y. S. Rawat, and M. Shah, "VideoCapsuleNet: A simplified network for action detection," 2018, *arXiv:1805.08162*. [Online]. Available: https://arxiv.org/abs/1805.08162

[79] D. J. Im, S. Ahn, R. Memisevic, and Y. Bengio, "Denoising criterion for variational auto-encoding framework," 2015, *arXiv:1511.06406*. [Online]. Available: https://arxiv.org/abs/1511.06406

[80] T.-H. Chan, K. Jia, S. Gao, J. Lu, and Z. Zeng, Y. Ma, "PCANet: A simple deep learning baseline for image classification?" *IEEE Trans. Image Process*, vol. 24, no. 12, pp. 5017–5032, Dec. 2015.

[81] S. Mahdizadehaghdam, A. Panahi, H. Krim, and L. Dai, "Deep dictionary learning: A PARametric NETwork approach," 2018, *arXiv:1803.04022*. [Online]. Available: https://arxiv.org/abs/1803.04022

[82] M. Tygert, J. Bruna, S. Chintala, Y. LeCun, S. Piantino, and A. Szlam, "A mathematical motivation for complex-valued convolutional networks," *Neural Comput.*, vol. 28, no. 5, pp. 815–825, 2016.

[83] L. Jiao, J. Zhao, F. Liu, and S. Yang, "Deep complex convolutional neural network," in *Deep learning, optimization and recognition*, 3nd ed. Beijing, China: Tsinghua Univ. Press, 2017, pp. 167–179.

[84] A. Kappeler, S. Yoo, Q. Dai, and A. K. Katsaggelos, "Video super-resolution with convolutional neural networks," *IEEE Trans. Comput. Imag.*, vol. 2, no. 2, pp. 109–122, Jun. 2016.

[85] Y. Liang, J. Wang, S. Zhou, Y. Gong, and N. Zheng, "Incorporating image priors with deep convolutional neural networks for image super-resolution," *Neurocomputing*, vol. 194, pp. 340–347, Jun. 2016.

[86] S. Lu, Y. Wang, and Y. Wu, "Novel high-precision simulation technology for high-dynamics signal simulators based on piecewise Hermite cubic interpolation," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 54, no. 5, pp. 2304–2317, Oct. 2018.

[87] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, vol. 1, no. 1, pp. 770–778.

[88] H. Ide and T. Kurita, "Improvement of learning for CNN with ReLU activation by sparse regularization," in *Proc. IEEE Int. Joint Conf. Neural Netw.*, May 2017, vol. 1, no. 1, pp. 2684–2691.

[89] S. Ren, K. He, and R. Girshick, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2015, pp. 91–99.

[90] T.-Y. Lin and P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, vol. 1, no. 2, pp. 936–944.

[91] E. Shelhamer, J. Long, and T. Darrell, "Fully convolutional networks for semantic segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 4, pp. 640–651, Apr. 2017.

[92] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," 2015, *arXiv:1511.06434*. [Online]. Available: https://arxiv.org/abs/1511.06434

[93] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel, "InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets," in *Proc. Neural Inf. Process. Syst.*, 2016, vol. 1, no. 1, pp. 2172–2180.

[94] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," 2017, *arXiv:1703.10593*. [Online]. Available: https://arxiv.org/abs/1703.10593

[95] E. L. Denton, S. Chintala, and R. Fergus, "Deep generative image models using a Laplacian pyramid of adversarial networks," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2015, vol. 1, no. 1, pp. 1486–1494.

[96] M.-B. Li, G.-B. Huang, P. Saratchandran, and N. Sundararajan, "Fully complex extreme learning machine," *Neurocomputing*, vol. 68, nos. 1–4, pp. 306–314, Oct. 2005.

[97] J. Tang, C. Deng, and G.-B. Huang, "Extreme learning machine for multilayer perceptron," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 4, pp. 809–821, Apr. 2015.

[98] J. Hu, J. Zhang, C. Zhang, and J. Wang, "A new deep neural network based on a stack of single-hidden-layer feedforward neural networks with randomly fixed hidden neurons," *Neurocomputing*, vol. 171, pp. 63–72, Jan. 2016.

[99] C. L. P. Chen and Z. L. Liu, "Broad learning system: An effective and efficient incremental learning system without the need for deep architecture," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 1, pp. 10–24, Jan. 2018.

[100] Z.-H. Zhou and J. Feng, "Deep forest: Towards an alternative to deep neural networks," 2017, *arXiv:1702.08835v2*. [Online]. Available: https://arxiv.org/abs/1702.08835v2

[101] Y. Yang, J. Sun, and H. Li. *ADMM-Net: A Deep Learning Approach for Compressive Sensing MRI*. Accessed: 2018. [Online]. Available: http://papers.nips.cc/paper/6406-deep-admm-net-for-compressive-sensing-mri.pdf

[102] S. Sabour, N. Frosst, and G. E. Hinton, "Dynamic routing between capsules," 2017, *arXiv:1710.09829*. [Online]. Available: https://arxiv.org/abs/1710.09829

[103] L. Jiao, L. Bo, and L. Wang, "Fast sparse approximation for least squares support vector machine," *IEEE Trans. Neural Netw.*, vol. 18, no. 3, pp. 685–697, May 2007.

[104] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," in *Proc. Int. Conf. Learn. Represent.*, 2014, vol. 1, no. 1, pp. 1–14.

[105] S.-H. Zhong, Y. Liu, and Y. Liu, "Bilinear deep learning for image classification," in *Proc. Int. Conf. Multimedea*, 2011, vol. 1, no. 1, pp. 343–352.

[106] Z. Pan, Y. Liu, G. Liu, M. Guo, and Y. Li, "Topic network: Topic model with deep learning for image classification," in *Proc. Int. Conf. Knowl. Sci., Eng. Manage.*, 2015, vol. 1, no. 1, pp. 525–534.

[107] T.-H. Chan, K. Jia, S. Gao, J. Lu, Z. Zeng, and Y. Ma, "PCANet: A simple deep learning baseline for image classification?" *IEEE Trans. Image Process.*, vol. 24, no. 12, pp. 5017–5032, Dec. 2015.

[108] W. Zhao and S. Du, "Spectral–spatial feature extraction for hyperspectral image classification: A dimension reduction and deep learning approach," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 8, pp. 4544–4554, Aug. 2016.

[109] P.-H. Liu, S.-F. Su, M.-C. Chen, and C.-C. Hsiao, "Deep learning and its application to general image classification," in *Proc. IEEE Int. Conf. Inform. Cybern. Comput. Social Syst.*, Aug. 2015, vol. 1, no. 1, pp. 7–10.

[110] Z. Zhao, L. Jiao, J. Zhao, J. Gu, and J. Zhao, "Discriminant deep belief network for high-resolution SAR image classification," *Pattern Recognit.*, vol. 61, pp. 686–701, Jan. 2017.

[111] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," 2016, *arXiv:1603.08155*. [Online]. Available: https://arxiv.org/abs/1603.08155

[112] L. A. Gatys, A. S. Ecker, and M. Bethge, "Image style transfer using convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, vol. 1, no. 1, pp. 2414–2423.

[113] A. Shinya, N. D. Tung, T. Harada, and R. Thawonmas, "Object-specific style transfer based on feature map selection using CNNs," in *Proc. Nicogr. Int.*, Jun. 2017, vol. 1, no. 1, p. 88.

[114] J. Fan, T. Zhao, Z. Kuang, Y. Zheng, J. Zhang, J. Yu, and J. Peng, "HD-MTL: Hierarchical deep multi-task learning for large-scale visual recognition," *IEEE Trans. Image Process.*, vol. 26, no. 4, pp. 1923–1938, Apr. 2017.

[115] W. Wang, J. Shen, and L. Shao, "Video salient object detection via fully convolutional networks," *IEEE Trans. Image Process.*, vol. 27, no. 1, pp. 38–49, Jan. 2018.

[116] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi, "Photo-realistic single image super-resolution using a generative adversarial network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, vol. 1, no. 1, pp. 105–114.

[117] E. Agustsson, M. Tschannen, F. Mentzer, R. Timofte, and L. Van Gool, "Generative adversarial networks for extreme learned image compression," 2018, *arXiv:1804.02958*. [Online]. Available: https://arxiv.org/abs/1804.02958

[118] G. Yang, S. Yu, and H. Dong, "DAGAN: Deep de-aliasing generative adversarial networks for fast compressed sensing MRI reconstruction," *IEEE Trans. Med. Imag.*, vol. 37, no. 6, pp. 1310–1321, Jun. 2018.

[119] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A deep convolutional encoder-decoder architecture for scene segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 12, pp. 2481–2495, Dec. 2017.

[120] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, vol. 1, no. 1, pp. 6230–6239.

[121] 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.*, 2015, vol. 1, no. 1, pp. 234–241.

[122] R. Fu, B. Li, Y. Gao, and P. Wang, "Content-based image retrieval based on CNN and SVM," in *Proc. IEEE Int. Conf. Comput. Commun.*, Oct. 2016, vol. 1, no. 1, pp. 638–642.

[123] J. Wan, D. Wang, S. C. H. Hoi, P. Wu, J. Zhu, Y. Zhang, and J. Li, "Deep learning for content-based image retrieval: A comprehensive study," in *Proc. ACM Int. Conf. Multimedia*, 2014, pp. 157–166. [Online]. Available: http://dayongwang.info/pdf/2014-MM.pdf

[124] F. Liao, E. Koshelev, M. Milton, Y. Jin, and E. Lu, "Change detection by deep neural networks for synthetic aperture radar images," in *Proc. IEEE Int. Conf. Comput., Netw. Commun.*, Jan. 2017, vol. 1, no. 1, pp. 947–951.

[125] L. Zhang, L. Zhang, and B. Du, "Deep learning for remote sensing data: A technical tutorial on the state of the art," *IEEE Geosci. Remote Sens. Mag.*, vol. 4, no. 2, pp. 22–40, Jun. 2016.

[126] W. Ouyang, X. Wang, X. Zeng, S. Qiu, P. Luo, Y. Tian, H. Li, S. Yang, Z. Wang, C. C. Loy, and X. Tang, "DeepID-Net: Deformable deep convolutional neural networks for object detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 7, pp. 1320–1334, Jul. 2016.

[127] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, vol. 1, no. 1, pp. 815–823.

[128] Y.-X. Wang and M. Hebert, "Learning to Learn: Model regression networks for easy small sample learning," in *Proc. Eur. Conf. Comput. Vis.*, 2016, vol. 1, no. 1, pp. 616–634.

[129] J. Maggu and A. Majumdar, "Greedy deep transform learning," in *Proc. IEEE Int. Conf. Image Process.*, Sep. 2017, vol. 1, no. 1, pp. 1822–1826.

[130] G. Hu, X. Peng, Y. Yang, T. M. Hospedales, and J. Verbeek, "Frankenstein: Learning deep face representations using small data," *IEEE Trans. Image Process.*, vol. 27, no. 1, pp. 293–303, Jan. 2018.

[131] D. George, W. Lehrach, K. Kansky, M. Lázaro-Gredilla, C. Laan, B. Marthi, X. Lou, Z. Meng, Y. Liu, H. Wang, A. Lavin, and D. S. Phoenix, "A generative vision model that trains with high data efficiency and breaks text-based CAPTCHAs," *Science*, vol. 358, no. 6, p. eaag2612, 2017.

[132] M. E. Taylor and P. Stone, "Transfer learning for reinforcement learning domains: A survey," *J. Mach. Learn. Res.*, vol. 10, pp. 1633–1685, Jul. 2009.

[133] Y. Bengio, "Deep learning of representations for unsupervised and transfer learning," in *Proc. Workshop Unsupervised Transf. Learn.*, 2012, vol. 7, no. 5, pp. 1–21.

[134] Y. Gal and Z. Ghahramani, "Bayesian convolutional neural networks with Bernoulli approximate variational inference," 2015, *arXiv:1506.02158*. [Online]. Available: https://arxiv.org/abs/1506.02158

[135] X.-Y. Liu, J. Wu, and Z.-H. Zhou, "Exploratory undersampling for class-imbalance learning," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 39, no. 2, pp. 539–550, Apr. 2009.

[136] X.-Y. Liu, J. Wu, and Z.-H. Zhou, "Exploratory under-sampling for class-imbalance learning," in *Proc. IEEE Int. Conf. Data Mining*, Dec. 2006, vol. 1, no. 1, pp. 965–969.

[137] M. Wasikowski and X. Chen, "Combating the small sample class imbalance problem using feature selection," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1388–1400, Oct. 2010.

[138] F. Shakeel, A. S. Sabhitha, and S. Sharma, "Exploratory review on class imbalance problem: An overview," in *Proc. Int. Conf. Comput., Commun. Netw. Technol.*, Jul. 2017, vol. 1, no. 1, pp. 1–8.

[139] S. Guan, M. Chen, and H. Y. Ha, "Deep learning with MCA-based instance selection and bootstrapping for imbalanced data classification," in *Proc. IEEE Conf. Collaboration Internet Comput.*, Oct. 2016, vol. 1, no. 1, pp. 288–295.

[140] S. Ertekin, J. Huang, and C. L. Giles, "Active learning for class imbalance problem," in *Proc. Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, 2007, vol. 1, no. 1, pp. 823–824.

[141] K. Tomanek and U. Hahn, "Reducing class imbalance during active learning for named entity annotation," in *Proc. Int. Conf. Knowl. Capture*, 2009, vol. 1, no. 1, pp. 105–112.

[142] M. Diligenti, S. Roychowdhury, and M. Gori, "Integrating prior knowledge into deep learning," in *Proc. IEEE Int. Conf. Mach. Learn. Appl.*, Dec. 2017, vol. 1, no. 1, pp. 920–923.

[143] J. Tang, L. Jin, Z. Li, and S. Gao, "RGB-D object recognition via incorporating latent data structure and prior knowledge," *IEEE Trans. Multimedia*, vol. 17, no. 11, pp. 1899–1908, Nov. 2015.

[144] S. N. Tran and A. S. d'Avila Garcez, "Deep logic networks: Inserting and extracting knowledge from deep belief networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 2, pp. 246–258, Feb. 2018.

[145] D. Zhang, J. Han, J. Han, and L. Shao, "Cosaliency detection based on intrasaliency prior transfer and deep intersaliency mining," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 6, pp. 1163–1176, Jun. 2016.

[146] M. Khodayar and M. Teshnehlab, "Robust deep neural network for wind speed prediction," in *Proc. Iranian Joint Congr. Fuzzy Intell. Syst.*, Sep. 2016, vol. 1, no. 1, pp. 1–5.

[147] K. Wang, C. Gou, Y. Duan, Y. Lin, X. Zheng, and F.-Y. Wang, "Generative adversarial networks: Introduction and outlook," *IEEE/CAA J. Autom. Sinica*, vol. 4, no. 4, pp. 588–598, Sep. 2017.

[148] H. Zhang, T. Xu, H. Li, S. Zhang, X. Wang, X. Huang, and D. N. Metaxas, "StackGAN++: Realistic image synthesis with stacked generative adversarial networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 8, pp. 1947–1962, Aug. 2019.

[149] T. Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár "Focal loss for dense object detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, to be published.

[150] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive growing of GANs for improved quality, stability, and variation," 2017, *arXiv:1710.10196*. [Online]. Available: https://arxiv.org/abs/1710.10196

[151] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a Gaussian Denoiser: Residual learning of deep CNN for image denoising," *IEEE Trans. Image Process.*, vol. 26, no. 7, pp. 3142–3155, Jul. 2017.

[152] T. Wang, M. Sun, and K. Hu, "Dilated deep residual network for image denoising," 2017, *arXiv:1708.05473*. [Online]. Available: https://arxiv.org/abs/1708.05473

[153] K. Zhang, W. Zuo, and L. Zhang, "FFDNet: Toward a fast and flexible solution for CNN based image denoising," *IEEE Trans. Image Process.*, vol. 27, no. 9, pp. 4608–4622, 2018.

[154] K. Zhang, W. Zuo, and L. Zhang, "Learning a single convolutional super-resolution network for multiple degradations," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, vol. 6, no. 1, pp. 3262–3271.

[155] U. Hwang, D. Jung, and S. Yoon, "HexaGAN: Generative adversarial nets for real world classification," 2019, *arXiv:1902.09913*. [Online]. Available: https://arxiv.org/abs/1902.09913

[156] M. Zhu, J. Xia, X. Jin, M. Yan, G. Cai, J. Yan, and G. Ning, "Class weights random forest algorithm for processing class imbalanced medical data," *IEEE Access*, vol. 6, pp. 4641–4652, 2018.

[157] Y.-C. Chen, Y.-J. Li, A. Tseng, and T. Lin, "Deep learning for malicious flow detection," 2018, https://arxiv.org/abs/1802.03358

[158] F. Wu, J. Wang, J. Liu, and W. Wang, "Vulnerability detection with deep learning," in *Proc. IEEE Int. Conf. Comput. Commun.*, Dec. 2018, pp. 1298–1302.

[159] Y. Zhang, D. Ying, and C. Liu, "Situation, trends and prospects of deep learning applied to cyberspace security," *J. Comput. Res. Develop.*, vol. 55, no. 6, pp. 1117–1142, 2018.

[160] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *Proc. ECCV*, 2016, vol. 9908, no. 1, pp. 630–645.

[161] F. Zhang, B. Du, and L. Zhang, "Saliency-guided unsupervised feature learning for scene classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 4, pp. 2175–2184, Apr. 2015.

[162] G.-S. Xia, J. Hu, F. Hu, B. Shi, X. Bai, Y. Zhong, and L. Zhang, "AID: A benchmark dataset for performance evaluation of aerial scene classification," 2016, *arXiv:1608.05167*. [Online]. Available: https://arxiv.org/abs/1608.05167

[163] S. Zhang, L. Wen, X. Bian, Z. Lei, and S. Z. Li, "Single-shot refinement neural network for object detection," 2017, *arXiv:1711.06897*. [Online]. Available: https://arxiv.org/abs/1711.06897

[164] K. Duan, S. Bai, L. Xie, H. Qi, Q. Huang, and Q. Tian, "CenterNet: Keypoint triplets for object detection," 2019, *arXiv:1904.08189*. [Online]. Available: https://arxiv.org/abs/1904.08189

[165] H. Law and J. Deng, "CornerNet: Detecting objects as paired keypoints," 2018, *arXiv:1808.01244*. [Online]. Available: https://arxiv.org/abs/1808.01244

[166] Q. Zhao, T. Sheng, Y. Wang, Z. Tang, Y. Chen, L. Cai, and H. Ling, "M2Det: A single-shot object detector based on multi-level feature pyramid network," 2018, *arXiv:1811.04533*. [Online]. Available: https://arxiv.org/abs/1811.04533

[167] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, "OverFeat: Integrated recognition, localization and detection using convolutional networks," 2013, *arXiv:1312.6229*. [Online]. Available: https://arxiv.org/abs/1312.6229

[168] P. Purkait, C. Zhao, and C. Zach, "SPP-Net: Deep absolute pose regression with synthetic views," 2017, *arXiv:1712.03452*. [Online]. Available: https://arxiv.org/abs/1712.03452

[169] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," 2016, *arXiv:1612.03144*. [Online]. Available: https://arxiv.org/abs/1612.03144

[170] F. Yang, H. Fan, P. Chu, E. Blasch, and H. Ling, "Clustered object detection in aerial images," 2019, *arXiv:1904.08008*. [Online]. Available: https://arxiv.org/abs/1904.08008

[171] A. Van Etten, "You only look twice: Rapid multi-scale object detection in satellite imagery," 2018, *arXiv:1805.09512*. [Online]. Available: https://arxiv.org/abs/1805.09512

[172] S. Vijayanarasimhan, S. Ricco, C. Schmid, R. Sukthankar, and K. Fragkiadaki, "SfM-Net: Learning of structure and motion from video," 2017, *arXiv:1704.07804*. [Online]. Available: https://arxiv.org/abs/1704.07804

[173] G. L. Oliveira, W. Burgard, and T. Brox, "Efficient deep models for monocular road segmentation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2016, vol. 1, no. 1, pp. 4885–4891.

[174] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. H. S. Torr, "Conditional Random Fields as Recurrent Neural Networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 1529–1537.

[175] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, Apr. 2018.

[176] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2881–2890.

[177] A. Paszke, A. Chaurasia, S. Kim, and E. Culurciello, "ENet: A deep neural network architecture for real-time semantic segmentation," in *Proc. ICLR*, 2017, pp. 1–10. [Online]. Available: https://openreview.net/pdf?id=HJy_5Mcll

[178] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," 2015, *arXiv:1511.07122*. [Online]. Available: https://arxiv.org/abs/1511.07122

[179] P. O. Pinheiro and R. Collobert, "Recurrent convolutional neural networks for scene labeling," in *Proc. ICML*, 2014, pp. 82–90.

[180] W. Byeon, T. M. Breuel, F. Raue, and M. Liwicki, "Scene labeling with LSTM recurrent neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 3547–3555.

[181] Z. Li, Y. Gan, X. Liang, Y. Yu, H. Cheng, and L. Lin, "LSTM-CF: Unifying context modeling and fusion with LSTMs for RGB-D scene labeling," 2016, *arXiv:1604.05000*. [Online]. Available: https://arxiv.org/abs/1604.05000

[182] C. R. Qi, H. Su, K. Mo, L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 652–660.

LICHENG JIAO (SM'89–F'18) received the B.S. degree from Shanghai Jiao Tong University, Shanghai, China, in 1982, and the M.S. and Ph.D. degrees from Xi'an Jiaotong University, Xi'an, China, in 1984 and 1990, respectively.

He is currently a Professor with the School of Artificial Intelligence, Xidian University, Xi'an. His research interests include image processing, natural computation, machine learning, and intelligent information processing. He is a member of the IEEE Xi'an Section Execution Committee and the Chairman of Awards and Recognition Committee, the Vice Board Chairperson of the Chinese Association of Artificial Intelligence, a Councilor of the Chinese Institute of Electronics, a Committee Member of the Chinese Committee of Neural Networks, and an Expert of the Academic Degrees Committee of the State Council.

JIN ZHAO received the M.S. degree in mathematics and applied mathematics from Shaanxi Normal University, Xi'an, China, in 2012, and the Ph.D. degree from the Key Laboratory of Intelligent Perception and Image Understanding of the Ministry of Education of China, Xidian University, Xi'an, in 2019. His research interests include sparse representation and meta learning, deep learning algorithm design, and performance analysis.