Journal of Advanced Engineering Trends
ISSN : 2682 - 2091

Vol.43, No.1. January 2024

http://jaet.journals.ekb.eg

Remote sensing Image processing, DEEP learning Classification algorithms survey and review

Ibrahim Moahmed Amer(1) **Kamel Rahouma**(2) **Attia Abd AlFattah Shahin**(3)
Head of Optical payload test lab, Assembly integration and test Center, Egyptian Space Agency, Cairo, Egypt (1) Professor Minia University(2) Professor NARSS(3)

A R T I C L E I N F O
Article history:
Received:
Accepted:
Online:

Keywords:
Deep learning
remote sensing
hyperspectral
image classification

A B S T R A C T

Image processing is considered a key in remote sensing fields. It is very wide and has many branches. One of the major and widely used techniques in image processing is image classification. Image classification is used to classify the extracted features from digital images into different classes based on different characteristics. Machine learning and deep learning are two main techniques to automate the processes of image processing and classification. This paper presents a survey of the recently used classification techniques using deep learning.

This paper aims to provide a review of the newly presented theories and tools used in deep learning for remote sensing and satellite images from 2018 to 2022.

Deep learning, as a subset of machine learning, needs high-processing devices like microprocessors, Field Programmable Gate Array (FPGA), and Graphics Processing Units (GPUs). Due to the progress in GPUs and other processors, it has become much easier to deal with deep learning (DL) for BIG DATA and satellite images.

1- Introduction

In recent years, deep learning has become one of the hottest and most promising points of research in many fields, such as computer vision and image classification [1]

Classification is divided into two main categories: supervised classifications and unsupervised classifications. Classification is considered the task of labeling pixels or segments and regions in a certain image into one of multiple classes. Deep learning methods aim to use a variety of methods to learn features from data and then perform classification at cutting-edge levels. [1]

This paper provides a review of the theories and tools used in deep learning for remote sensing and satellite images. We go through twelve papers submitted between 2018 and 2022 which include “SalvageDNN” [2], “DeepSat v2” [3] “FrequentNet” [4], “multitask deep learning method for the classification of multiple hyperspectral data in a single training” [5], “review different machine learning techniques, traditional Machine Learning, and Deep Learning in satellite data processing applications.”[6], “full stage data augmentation framework to enhance the accuracy of DCNN” [7]. “Enhance performance of DNNs on general-purpose processor (GPP) cores” [8], “develop a systematic framework for practical extraction and shape analysis using DCNN” [9], “Autonomously generate a DCNN model, use the autonomous and continuous learning (ACL) method. for every single vision task.” [10], “Scale Sequence Join Deep Learning (SS-JDL) technique for joint LU and LC classification in automatic method”[11], “categorize unlabeled HRRS images, a deep model was constructed using a labelled land-cover dataset” [12], and ” Satellite Image Classification and Analysis using Machine Learning with ISRO LISS IV” [13].

This paper is organized as follows: Section 1 gives an introduction, and Section 2 states the problem of the paper. Section 3 introduces the literature review and overview of deep learning in remote sensing, providing approaches, theories, and tools. Section 4 discusses and compares the different methods introduced in Section 3. Section 5 gives points of future research based on the discussed literature. Section 6 gives a conclusion. A list of the used references is given at the end of the paper.

2- Problem statement

In remote sensing satellites, a huge amount of data is obtained from different detectors. These detectors are used in various wavelengths and electromagnetic spectrums. For earth observation satellites, detectors are used to measure the reflectance and absorption over multiple wavelengths which is known as the spectral signature of the material. The spectral signature is unique and by using these features, materials could be recognized. Image classification is the method utilized to represent the themes categories for subjects and materials in the images [14].

The purpose of this study was to increase the authors' knowledge in the area of deep learning and to look for fresh ideas for analyzing CubeSat data.

Deep learning, as a subset of machine learning, needs high-processing devices like microprocessors, FPGAs, and GPUs. Recently, due to the progress in GPUs and other processors, it has become much easier to deal with DL for BIG DATA and satellite images.

The following are the paper's contributions:

We focus on Learning algorithms and DL structures for spectral, spatial, and temporal data, a deeper theoretical knowledge of DL systems, DL optimization and training.

Additionally, we categorize DL techniques into several domains according to applications application. We present methods used by contemporary RS researchers to implement DL in RS, including novel architectures, tools, and DL components.

Revised:11 June, 2022, Accepted:4 February , 2023

3- Literature review

Deep neural networks (DNN) are being speeded up daily in many applications, especially in data processing, data analysis, and predictive and knowledge portals. DNN has reached a noticeable position in many applications, especially in image processing and big data. In [2], the authors aimed to develop highly performance-efficient DNN accelerators to overcome the urgent need to improve the outcome of the manufacturing procedures and to keep the DNN accelerators' unit costs as low as possible. The authors introduced the SalvageDNN methodology to enable solid execution of DNNs on the equipment accelerator agents with permanent errors (regularly because of the imperfection of the manufacturing process). SalvageDNN utilizes a fault-aware mapping in various parts of a given DNN on the hardware accelerator (exposed to faults) by utilizing the saliency of the DNN parameters and the fault map of the fundamental processing hardware. In addition, the authors introduced novel adjustments in a systolic array design to assist the progress of the surrender of the accelerators while guaranteeing dependable DNN execution using "SalvageDNN". In terms of space, power/energy, and performance, the overheads are unimportant.

The authors of [2] employed

$$A_i^{(l)} = f^{(l)}\left(\sum_j W_{(i,j)}^{(l)} \times A_i^{(l-1)} + b_i^{(l)}\right) \qquad (1)$$

Where $W_{(i,j)}^{(l)}$ is the weight of the link between the ith neuron of the lth layer and the jth neuron of the l - 1st layers, $b_i^{(l)}$ is the ith neuron linked to a bias., $A_i^{(l)}$ is the ith neuron generates activation, and $f^{(l)}()$ is the lth layer's activation function.

SalvagingDNN accelerators algorithm is shown in **Error! Not a valid bookmark self-reference.**

The lth layer of a DNN's neurons/filters' saliency is calculated by [2]

$$s_i^{<l>} = \sum_k \left|W_{(k,i)}^{<l+1>}\right| x\, S_k^{<l+1>} \qquad (2)$$

where $s_i^{<l>}$ is the saliency of the ith neuron/filter in the DNN's lth layer[2].

The fully-connected layer of the l + 1th layer is [2]

$$s_i^{<l>} = \sum_k \sum_r \sum_c \left|W_{\{(r,c),(i,k)\}}^{<l+1>}\right| x\, S_k^{<l+1>} \qquad (3)$$

To minimize the total saliency of the weights to be trimmed in argminmapping||S*. ∗ PM||. (4) used [2]

$$argmin_{mapping}||S^{*}. * PM||. \qquad (4)$$

where S* is an altered version of S obtained by rearranging network parameters according to the mapping approach.

The authors of [2] presented the optimization of hardware to mitigate parameter faults presented in [15] by inserting more multiplexers in the datapath for MAC computation bypass in the case of faults in the PEs. The achievement was made possible by the arrangement that reduces the saliency of the parameters to a minimum due to the disengagement of the MAC units/processing components that aren't working properly. Increasing the number of bypass connections is better to lessen the impact of parameter faults in multiplexer units. However, the side effect of increasing this number of bypass connections is that they take up more area, power, and delay overhead.

To check how fault-aware mapping affects the DNN accuracy, the authors of [2] used a moderate DNN (the VGG11 and the VGG16) [16]. The Cifar-10 and ImageNet [16] datasets were used to train the model. To accomplish a considerable comparison result, the authors used, for all arrays and defect rates, the same experiment seeds. In addition, the MAC and MUX units' areas were taken into consideration to distribute the faults. Identically in real-life situations (fault rate up to 6% of the MAC units are faulty), if the manufacturing process is established, SalvageDNN can assist in maintaining the network's baseline accuracy, whereas the DNN precision of a network planned using FAP begins to deteriorate almost immediately as the defect rate rises.

The introduced SalvageDNN is adjusted according to the fault place and it introduces less deviation in comparison to the FAP method. With the scenario of using a 256x256 sized array, SalvageDNN could help the network maintain accuracy near the baseline even with a fault rate of around 6%. This scenario shows the advantage in the result compared to the FAP approach, which is apparently dropped even with Only 2% of PEs being defective[2].

The authors of [3]employed the following:

a. The structure of the framework has two convolutional layers, one of them with 32 feature maps and the other with 64 feature maps. Both have a kernel of 3X3.
b. A Rectified Layer Unit (ReLu) layer follows both convolutional layers.

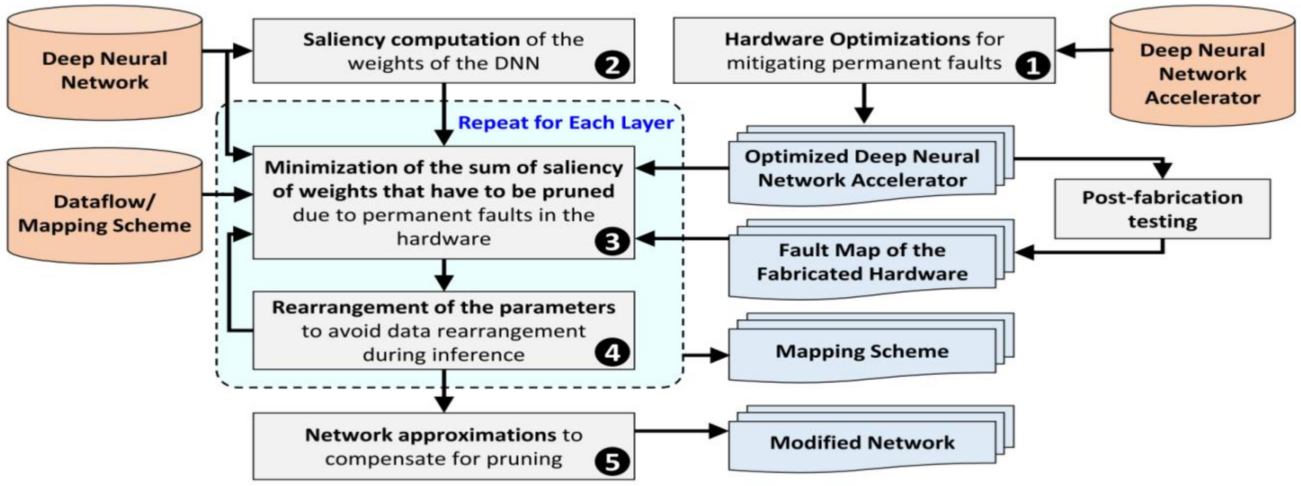

Deep Neural Network
Saliency computation of the weights of the DNN 2
Hardware Optimizations for mitigating permanent faults 1
Deep Neural Network Accelerator
Repeat for Each Layer
Minimization of the sum of saliency of weights that have to be pruned due to permanent faults in the hardware 3
Optimized Deep Neural Network Accelerator
Dataflow/ Mapping Scheme
Post-fabrication testing
Fault Map of the Fabricated Hardware
Rearrangement of the parameters to avoid data rearrangement during inference 4
Mapping Scheme
Network approximations to compensate for pruning 5
Modified Network

Figure 1: Saliency-driven fault-aware mapping of DNNs on hardware with persistent faults: an overview of the salvageDNN technique [2]

c. A max-pooling layer is used after the ReLu layer with a 2x2 kernel.

d. After that, following the max pooling, a dropout layer with a dropout rate of 0.25 is added.

e. The handcrafted features are in sequence with CNN.

f. A feature fusion layer follows both CNN and handcrafted features. The fused features are utilized as an input for the 32-neuron dense fully connected layer that follows.

g. After the fully connected layer, a batch normalization layer is applied, followed by a ReLu.

h. A fully connected dense layer with 128 neurons is used followed by ReLu.

i. The ReLu is then followed by a dropout layer with a rate of 0.2.

j. The Softmax layer, which is based on the cross-entropy loss function, is the final additional layer.

k. The Adadelta optimizer [17] has been endorsed in the framework.

Remote sensing images are considered a key point in the revolution and developments. Satellite images, as a critical branch of remote sensing systems, have very promising and challenging research, especially in image classification techniques. Satellites have many types of payloads, starting from frame cameras to hyperspectral cameras. Because of these various types, image data is different and huge. As a result of that, most existing object categorization approaches are ineffective when dealing with satellite datasets. The goal of the authors of [3] was to present two new high-resolution satellite imaging datasets (SAT-4 and SAT-6). They also presented the "DeepSat" classification system, which is based on "handcraft" traits and a deep belief network (DBN)[3].

The authors of [3] introduced an end-to-end framework utilizing a novel architecture that extracts spatial information using CNN as a baseline model. They augmented it with handcrafted features to enhance the power of discrimination [3]. The authors gave a comparison of the accuracy of several classification algorithms in SAT-4 and SAT-6.

Figure 2 illustrates the full deign of the framework[3]

The authors of [3] presented a design for the SAT-4 and SAT-6 datasets[18] which provides, as he claimed, enough labeled image patches (500,000 for SAT-4 and 405,000 for SAT-6). The handcrafted features showed improvement in discriminative feature learning. Removing the second dense layer had a bad impact on the precision of classification, and the network performance was reduced concerning the classification accuracy. The dropout layer with a rate of 0.2 gave the best classification accuracy. Compared with state-of-the-art techniques, SAT-4 had reached a classification accuracy of 99.90% and SAT-6 had reached 99.84%.

Convolution Neural Networks (CNN) have accomplished huge progress with high success in the image classification field[4]. In [4], the authors aimed to introduce a new baseline deep learner model called “FrequentNet” for image classifications. The authors, rather than utilizing “Principal Component Analysis (PCA)” vectors as the filter vectors in (PCANet [19]), in discrete Fourier analysis and wavelet analysis, basis vectors were employed. In his proposed filter Victor. In addition, they accomplished two different tasks on handwritten digits recognition and object recognition and compared the performances of “FourierNet”, “PCANet” and “RandNet” on those tasks[3].

The authors of [4] employed the following equation For N used as the number of input training images, $\{\mathrm{Ii}\}_{i=1}^{N}$ with size (m x n) patch size was set as k_1 x k_2 at all stages. $\mathbf{x}_{i;1}$; to , $\mathbf{x}_{\mathbf{i;mn}}$

are called vectorized patches, where the image index is the first,

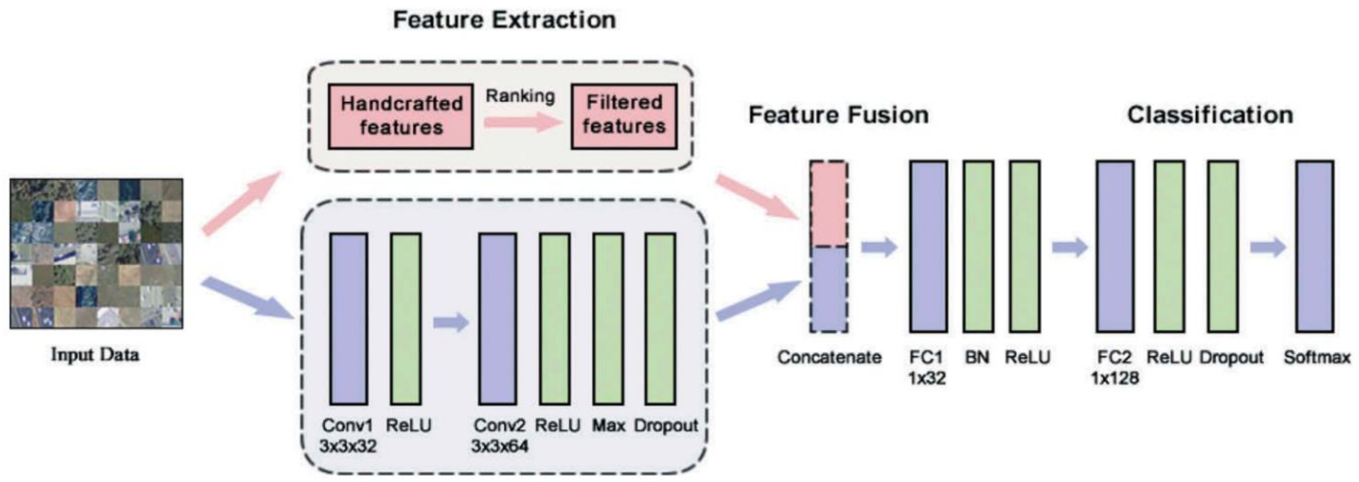


Figure 2 DeepSat V2 Classification framework architecture[3]

while the patch index is the second. The patch mean is then subtracted from each patch as shown in $\bar{X}_i$ = [$\mathbf{x}_{i;1}$; · · · ; $\mathbf{x}_{i;j}$; · · · $\mathbf{x}_{i;mn}$], 1 ≤ j ≤ mn (5) of size $k_1k_2 \times mn$. [4]

$$\bar{X}_i = [\mathbf{x}_{i;1}; \cdots; \mathbf{x}_{i;j}; \cdots \mathbf{x}_{i;mn}],\ 1 \le j \le mn \qquad (5)$$

Then stack $\bar{X}_i$ again to get (6) Its size is $k_1k_2 \times mnN$ [4]

$$\bar{X} = [\bar{X}_1 ; \cdots ; \bar{X}_i ; \cdots \bar{X}_N],\ 1 \le i \le N \qquad (6)$$

For **FourierNet**, as a first stage, a Filter with various frequencies was chosen based on the magnitude of $\boldsymbol{x}_{i;j}$ and candidate filters. A new feature map L1was obtained from every input image I_i with filters $\mathbf{v_1,..v_k,..v_{L1}}$ as shown In (7) where ∗ is the two-dimensional convolution [4]

$$I_i^k = I_i * mat_{k1,k2}(\boldsymbol{v_k}) \qquad (7)$$

For the second stage, a new set of feature maps was obtained with an identical size to the original images.

For the output stage, a histogram was used to get the final feature vector. This was done by converting all feature maps to binary grouping them by the parent feature maps. As shown in [4]

$$T_i = \sum_{k=1}^{L_1} 2^{k-1} B(I * \boldsymbol{v}_k) \qquad (8)$$

$$T_i^k = \sum_{l=1}^{L_2} 2^{l-1} B\left(I_i^k * \boldsymbol{u}_l\right) \qquad (9)$$

For hand-written digit recognition, the authors of [4] used a subset of MNIST [20] and MNIST variations [21] for the setup of the experiment. For the two-stages model, the authors of [4] primarily compared the performance between models with various basis vectors.

The experiment results showed that for one-stage models with two to eight filters, the tested accuracy increased as the number of filters increased. For the two-stage models. The results showed that FourierNet-2 and WaveNet-2 reached the same testing accuracy on these datasets [4]

For object recognition, the authors [4] used CIFAR10 [19], which has 10 classes with 50000 training samples and 10,000 test samples. These samples vary in object characteristics such as object position, scale, color, and texture. In the experiment's first stage, the authors [4] used 40 filters and reduced them to 5 in the second stage. The patch size was selected to be 5x5 to 8x8 and the block overlap to 4. The authors of [4] tried different combinations to check the accuracy, and the results are listed in Table 1 [4]

Table 1 On CIFAR10, many approaches were tested for accuracy (percentage)

Methods	Methods
Fourier-Fourier	67.70
Fourier-PCA	68.30
PCA-Fourier	69.75
PCA-PCA	70.95

Image classification in the remote sensing field gives a lot of information about land use and land cover, which is essential for environmental and urban management. Hyperspectral images contain a huge amount of spectral data due to the high number of bands, and it is considered a high challenge in the field of land use and land cover mapping, as well as classification[5]. The authors of [5], aimed to introduce a multitask deep learning approach for the classification of different hyperspectral data in a single training.

The authors of [5] trained an identical feature extractor for every piece of data, and the retrieved features were input into the SoftMax classifiers that were associated with them. A spectral knowledge base was presented to guarantee that common characteristics remained consistent across domains. The approach was evaluated using four distinct hyperspectral data sets.

The architecture of the experiment: hyperspectral ResNet (HResNet), multitask as well as single task, is illustrated in Figure 3

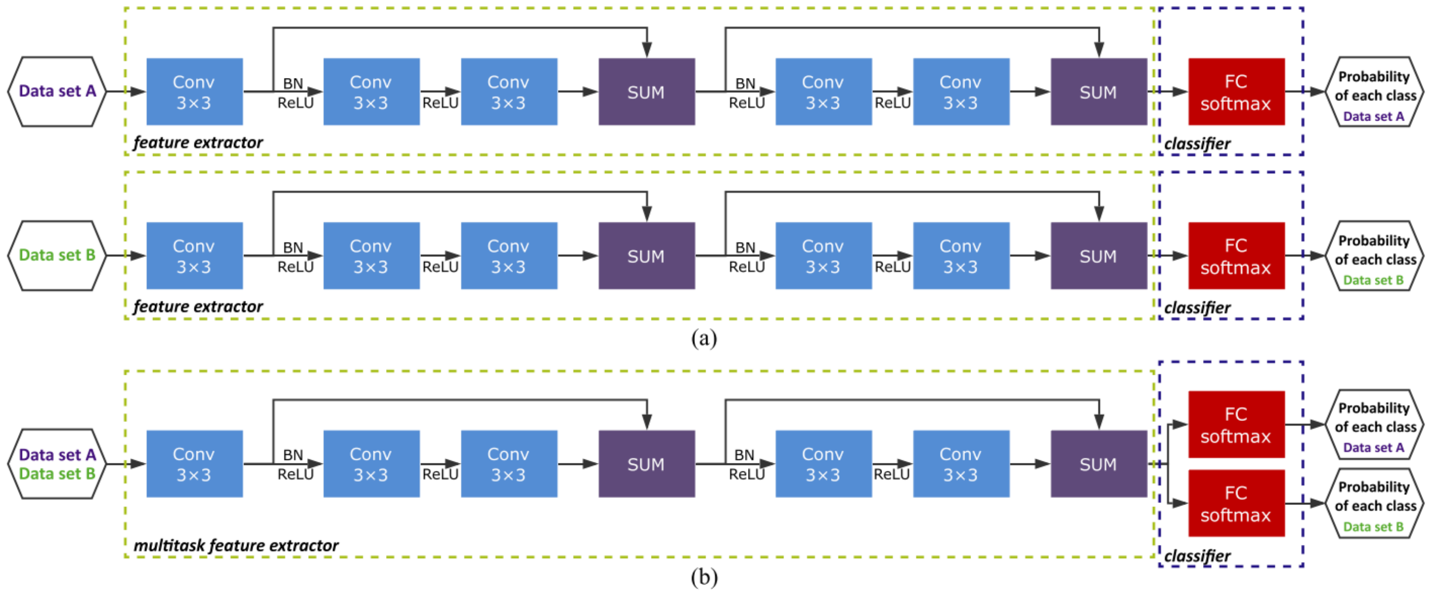


Figure 3 architecture of HResNet (a) single-task model (b) multitask model

The author of [5] employed the following

DCNN in this architecture is the integration of feature extraction ø(.) and a classifier f(.). The shallow layers are used to extract features, whereas the final fully connected layer with the softmax function is used as a classifier. The feature extractor ø(.) extracts fuses and transforms the sample x into a feature vector $x_ø$ that the classifier f(.). can better discriminate for input image sample *x*, [5]as shown in

$$x_ø = ø(x) \qquad (10)$$

The output vector xø is taken by The classifier f(.). from the feature extractor ø(.) as its source of data The fully connected layer of a CNN uses the softmax activation function. works as the classifier f(.). and gives the probability P(y = j| xø) of the jth category[5] as shown in y_{ij}=AF($\sum_{m=i-S}^{i+S}\sum_{n=j-S}^{j+S}(x_{mn}w_{mn}+b)$

$$P(y = j|\, x_ø) = \frac{exp(x_ø^T\ \omega_j + b_j)}{\sum_{k=1}^{K} exp(x_ø^T\ \omega_k + b_k)} \qquad (11)$$

In multitask system, one feature extractor ø(.) is sufficient for both or even several data sets, reducing network traffic. [5] as shown in

$$ø(.)= ø_p(.) = ø_q(.) \qquad 12$$

Therefore, a multitask DCNN (ø(.), f_p(.), f_q(.)) could be trained for both data sets.

It could be noticed that the structure of a Deep Convolution Neural Network is a combination of a feature extractor and a classifier. The problem of overfitting could be solved and processed by using one feature extractor for multiple datasets. This entails training a deep learning model using data from several domains. Multitask learning for hyperspectral data classification is based on the idea that Earth observation data should have similar spectral-spatial properties since they represent the brightness and reflectance of ground objects, which is independent of the dataset[5].

A fully connected layer using the SoftMax function for the classification of the second data distinguishes the multitask version from the single-task version. It should be noted that the introduced multitask deep learning is a method that could be integrated with any network including a large number of parameters[5].

The investigations employed four hyperspectral data sets; the “Pavia University” (PU) dataset, the Pavia Center (PC) dataset, the “Indian Pines” (IN) dataset, and “Salinas Valley” (SA) [5].

Lastly, concerning the capitulation time, the comparison result is listed in Table 2 [5]

Table 2 COMPARE THE TIME IT TAKES TO COMPUTE[5]

	Single-task			Multitask learning	
	OA (%)	Time (s)		OA (%)	Time (s)
Pavia University	72.5	15.9	34.6	75.1	23.2
Pavia Center	96.6	18.7		97.3	
Indian pines	48.1	25.3	49.7	53.6	32.5
Salinas Valley	81.3	24.4		77.8	

Due to the rapid development of imagers' sensors in satellite technology, the imager output data has a huge amount of information concerning spectral, spatial, and radiometric data. This huge amount of information leads to speeding up the data processing with sustained accuracy.[6]. The authors of [6]aimed to review different machine learning techniques, including traditional machine learning and deep learning, in satellite data processing applications[6]. Various techniques of machine learning were developed and introduced to deal with satellite images.

The review was divided into three main parts.

a. Satellite data processing Machine Learning

b. Recent Satellite Data Processing Machine Learning

c. Deep Learning methods

a) Satellite data processing Machine learning

Machine learning in satellite data processing, Figure 4 illustrates all satellite data processing stages using machine learning as an integral part, starting from preprocessing till reaching decision making[6] N. Sisodiya [6] Reviewed the main three applications, so-called, classification, segmentation, and denoising have been studied in [22]. According to the review, each application field required satellite image data. As an example," land use", "land cover" and vegetation monitoring necessitate classification of the satellite image under investigation, whereas urban growth monitoring, road and building extraction, and detection necessitate segmentation. Finally, denoising is a type of preprocessing that is on par with classification and segmentation in terms of relevance. [6]

To fulfill the work, the authors of [6] introduced different algorithms listed in them Table 3 as well as their application fields.

b) Recent Satellite Data Processing Machine Learning

Recent Satellite Data Processing Machine Learning is listed in Table 4[6]

Table 3 Various uses for classical machine learning techniques[6]

Processing stage	ML method	Application
Preprocessing	"GMM", "ANN"	"Denoising"
	"PCA", "LLE", "ISOMAP"	"Dimensionality reduction"
Processing/analysis	Sparse coding	"Sparse representation"
	"HOG"," SURF SIFT", [23], "decision trees", "random forest", "genetic algorithm", "HMRF"[1]," "SVM", "MRF"	"Feature selection"
	"Decision tree", "multilayer perceptron", "logistic regression", "SVM", "K-means", "GP", "NN"[24], "ARIMA"	"Segmentation" and "classification"
	"K-nn", "SVM," "SOM", "GMM"," K-means", "fuzzy clustering" [25], [26], "hierarchical clustering" [27], "hybrid clustering"	"Clustering"
	"Image transformations, correlation analysis"	"Change detection"

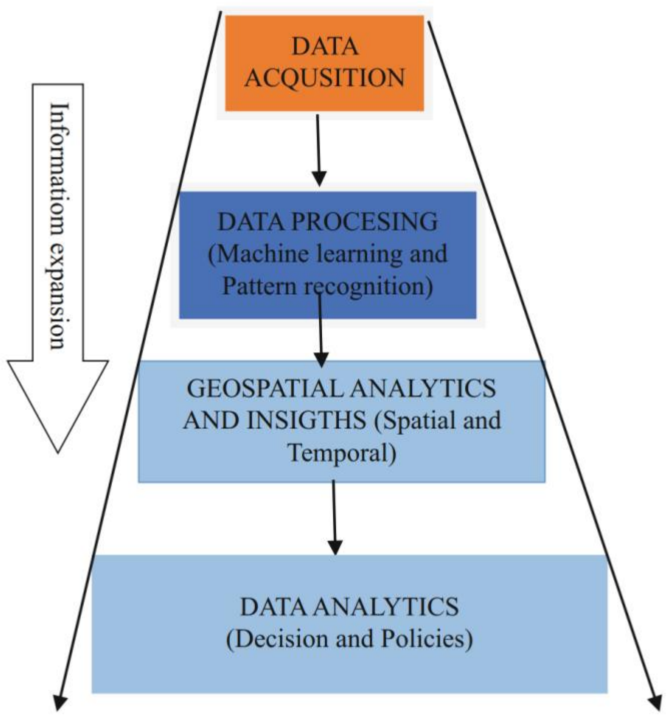

DATA ACQUSITION
DATA PROCESING (Machine learning and Pattern recognition)
GEOSPATIAL ANALYTICS AND INSIGTHS (Spatial and Temporal)
DATA ANALYTICS (Decision and Policies)
Informatiom expansion

Figure 4 The stages of satellite data processing[6]

Table 4 Recent Satellite Data Processing Machine Learning[6]

Learning type	ML method	Application
Manifold Learning	"unnormalized graph Laplacian" [28]," LapR and HLapR" [29], "Graph-based Laplacian energy", "LLE"[30]	"Dimensionality reduction", "feature extraction",
Semisupervised	"adapted graph-based SVM", "TSVM"[31], "modified TSVM"[32], [33], Fisher "discriminant classifier"	"Dimensionality reduction", "feature extraction"
Transfer Learning	"NN", "DASVM", [34] "maximum margin-based" "clustering" [35]	"Classify time series data"
Active Learning	"SVM"," maximum likelihood"	selection of most relevant sample, object-oriented classification, target detection
Structured Learning	"SSVM" [36]	Multiclass classification and prediction

c) Deep Learning methods

Deep learning is a special type of neural network that has typically four or five layers deep, of nonlinear transformation.

It is categorized as unsupervised machine learning. It is not used as an image classifier only but is also used as a feature-based image registration[37]. The comparison of a variety of deep learning methods, each with its own set of uses and benefits, are listed in Table 5[6]

Table 5 comparison of a variety of deep learning methods, each with its own set of uses and benefits. [6]

DL method	Data type	Application	Advantages
"CNN"	"SAR", "HIS", "multispectral images", "LiDAR data", "RGB", "pansharpening images"	"Image segmentation", "urban area classification" "extraction", "urban growth prediction", "road extraction", oceanographic "target classification", "thematic classification", "automatic target detection", "super-resolution" for sea surface "temperature analysis", "object detection and recognition", "3D object classification", "human detection and activity" "classification, semantic labeling multisource earth observation data", "anomaly detection", "plants disease recognition", "denoising", "poverty", "multiscale classification"	An input was "Automatic feature extraction for learning" from a big labeled dataset. It's simple to learn. CN may be used to create a pre-trained model that can be utilized for transfer learning. . Recurrent CNN, convolutional LSTM, and other forms of DL architectures can be integrated with CNN to improve the performance of the design
"Recurrent NN"	Spatiotemporal and time series data	Ocean and weather forecasting, tracking for multiple objects	sequential data
"Recursive NN" (LSTM)	Spatiotemporal data	Precipitation nowcasting[37]	Its ability to operate with sequential data is aided by the inclusion of a memory unit.
"Deep belief network" (DBN)	"Polarimetric SAR"," HI"S [1], "high-resolution image"," time series data", "radar data", "spatiotemporal data"	"Classification"," drought index prediction", "object detection", prediction of traffic flow, urban LULC	It's possible to utilize it with the few labeled datasets that are available.

It is used mainly to deal with big data, for high-resolution satellite images as well as for hyperspectral satellite images. [6] An enhanced training sample selection technique was examined using the active learning approach offered by two-staged spatial computing to achieve greater classification accuracy.

The authors of [6]introduced an experiment as a use case in artificial intelligence. Concerning the way of using the acquired satellite data processing and analysis, there are two main data categories: a) direct applications, in which machine learning techniques process the satellite images Immediately, this provides scene experiences such as object detection and change detection. b) Derived applications in which more complex and advanced models use a group of features. Which thus used to determine decision-making strategies and uses the acquired data obtained from the satellite for such things as profit earned by retailers and crop yield estimation, price prediction, and economic growth monitoring.

<u>According to the previously explanation, there are five points to be discussed:</u> [6]

1- <u>Object detection in a high-resolution image</u>

Object detection is indeed considered a key point of challenge in satellite imagery because of the following:

A. Objects are located in very small areas compared to satellite images.
B. A shortage of available training data sets
C. Modification, adaptation, and optimization of algorithms to work properly at various scales and objects.

Change detection and time series analysis-based applications are two key challenges in object detection that have been solved by a few famous deep learning architectures such as FasterRCNN and You Only Look Once (YOLO) [38]. Unsolved problems are still a mature study topic.

2- <u>Change detection</u>

Due to temporal resolution, the images are captured at different times. Change detection is used to investigate the change in the region under capture. For binary changes, it's not always clear if a pixel belongs to one of the two expected classes or whether it's a multiclass alteration. [6]

For multiclass classification, it is needed to use supervised methods. It is hard to develop a multiclass classification solution because of the next points: [6]

a. It is hard and costly to collect ground truth data.
b. Normalization of data
c. The atmospheric and climatic effects as well as lenses effects
d. The choice is a machine learning technique due to intensive information about remote sensing.

3- <u>Profit earned by retailers</u>

Unlike the direct applications mentioned above these get their focus data from satellite imagery that serves as the foundation for complicated frameworks. In the retail sector, the number of cars parked in a parking garage may be used to estimate the profit that can be earned by the shop. Monthly, quarterly or annual reports might be used to account for the projected benefit at that time[6].

4- <u>Crop yield estimation and price prediction.</u>

The Normalized Vegetation Indicator (NDVI) is a vegetation index that provides critical information for agricultural production evaluations and price forecasting. Ranchers, ware dealers, protection strategy producers, agribusiness government arrangements, and many others use farming-related expertise to meet their needs. Ranchers are being assisted by artificial intelligence (AI) to determine the ideal times and locations for farming a given crop[6].

5- <u>Economic growth monitoring</u>

Satellite images can show the financial activity of hard-to-reach countries. The number of high-rise buildings and expanding construction, power usage, and the number of automobiles and roads are all factors that may be used to gauge economic advancement[6].

Deep learning has produced outstanding outcomes in a variety of computer vision fields, especially big data[7]. The authors of [7] aimed to introduce a full-stage data augmentation framework to enhance the precision of DCNN, which, in addition, could be considered as a part of a prototype model without the need to present extra model training costs. [7]

The authors of [7] introduced a full-stage data augmentation framework This comprises training and testing steps for natural image classification using deep learning Data augmentation is used throughout the training phase to ensure that the network can mine structural information and, finally, coverage in the correct spot. Augmentation in two phases should be predictable to guarantee the exact transfer of domain information. The experiment was done on two datasets, fine-grained and coarse-grained which are CIRAR-10 and CIFAR-100 [39], respectively. The results were compared with different algorithms. The total flow chart of training and testing of DCNN is illustrated in **Error! Reference source not found.** [7]

The authors of [7]employed the following equations In the forward propagation stage, each layer's input is the output of the preceding layer.. output $\mathbf{h}_l$ of the l-th layer in DCNN for $l = 1, \dots, L-1$ is illustrated in

$$\boldsymbol{h}_l = \sigma(\boldsymbol{W}_0^l \boldsymbol{h}_{l-1} + \boldsymbol{b}_0^l) \tag{13}$$

$\mathbf{h}_0 = \mathbf{x}$ and $\sigma(\cdot)$ is the nonlinear activation function for each element like “Leaky-ReLU” [40]. As illustrated inσ (x)= $x,\ \ if\ x>0 xa,\ \ if\ x\leq 0$ ⍰⍰) where a is a fixed hyperparameter in (1, +∞)[7].

$$\sigma(x) = \begin{cases} x, & if\ x > 0 \\ \frac{x}{a}, & if\ x \leq 0 \end{cases} \tag{14}$$

The final output of DCNN[7] is illustrated in **Error! Reference source not found.**

$$f(\boldsymbol{x}) = softmax(\boldsymbol{W}_0^L \boldsymbol{h}_{L-1} + \boldsymbol{b}_0^L) \tag{15}$$

Softmax(.) is defined as in (16) where C is the final layer's number of neurons[7] as shown in

$$\text{Softmax}(f)_i = \frac{e^{f_i}}{\sum_{j=1}^{c} e^{f_j}}\ for\ i = 1,2, \dots .. C \tag{16}$$

At last, the training loss of DCNN[7] is illustrated in

$$\int(x_i, y_i) = -\frac{1}{C} \sum_{j=1}^{C} [\, y_i^j \log f(x_i)^j + (1 - y_i^j) \log(1 - f(x_i)^j] + \lambda \sum_{j=1}^{C} \|\boldsymbol{W}^{\boldsymbol{k}}\|_F \tag{17}$$

Minimizing the loss by updating parameters is the target in the back-propagation stage (weights W and biases b) in DCNN, as in (18) and (19) where α represents the learning rate and M represents the batch size[7]

$$\boldsymbol{W}_0^l = \boldsymbol{W}_{t-1}^l - \alpha . \frac{1}{M} \sum_{i=1}^{M} \frac{\partial \int(\boldsymbol{x}_i, \boldsymbol{y}_i)}{\partial \boldsymbol{W}_{t-1}^l} \tag{18}$$

$$\boldsymbol{b}_0^l = \boldsymbol{b}_{t-1}^l - \alpha . \frac{1}{M} \sum_{i=1}^{M} \frac{\partial \int(\boldsymbol{x}_i, \boldsymbol{y}_i)}{\partial \boldsymbol{W}_{t-1}^l} \tag{19}$$

For data augmentation, each test image is augmented to M̃ for M images via data used in the training process[7] as illustrated in

$$f(x) = \frac{M}{\tilde{M}} \sum_{i=1}^{\tilde{M}/M} f(\tilde{\boldsymbol{x}}_i) \tag{20}$$

Based on majority voting, the final prediction[7] is shown in

$$f(\boldsymbol{x}) = \frac{M}{\tilde{M}} \sum_{i=1}^{\tilde{M}/M} f[\boldsymbol{g}_i(\boldsymbol{x})] = \frac{M}{\tilde{M}} \sum_{i=1}^{\tilde{M}/M} \tilde{f}_i(\boldsymbol{x}) \tag{21}$$

The authors of [7] utilized the two benchmarks (CIFAR-10 and CIFAR-100) to evaluate the efficiency of full-stage data augmentation frameworks in a variety of situations. Both datasets are labeled subsets of the larger collection of 80 million small pictures. They also have 50,000 training photos and 10,000 test images on hand. The sole difference between the two datasets is that in CIFAR-10, the photos are evenly divided into 10 classes, whereas in CIFAR-100, the images are evenly dispersed into 100 classes.

a. The original images are color-normalized and zero-padded to be 40x40 pixels.

b. Images are zero-padded to be 40x40 pixels.

c. For data augmentation, both the training and testing stages are reduced to 32x32 pixels.

d. As per stage training and testing, the trimming stage is followed by a random horizontal flip with a chance of 50%. The sample size was tenfold increased to account for model stability.

e. Each image subtracts its unique three-channel color (R/G/B) mean value to accelerate the convergence of the DCNN model.

The network architecture of both datasets is shown in Figure **6**. The main difference between both designs is that the network trained on CIFAR-100 used a deeper and broader structure than the network trained on CIFAR-10. This design difference is used because finer-grained data needs more capacity for the mode to be characterized [7].

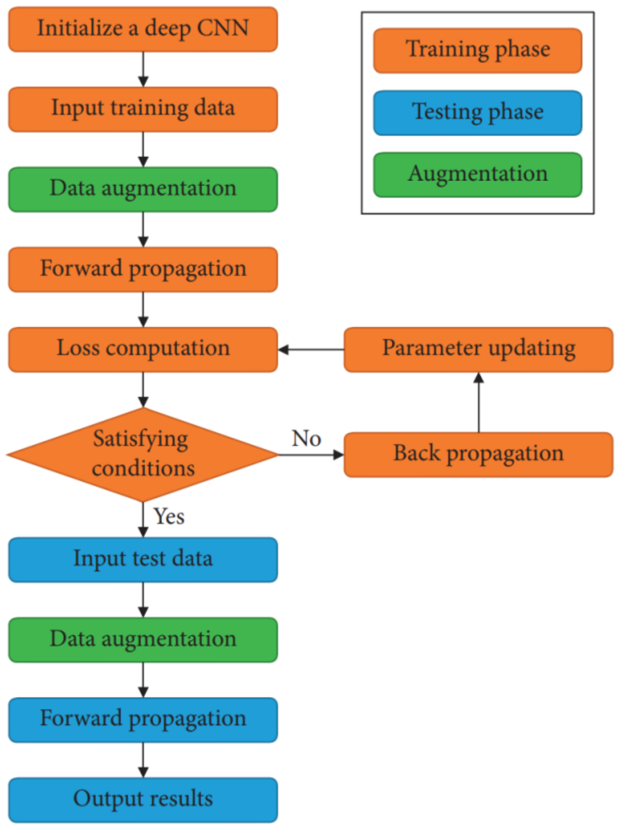
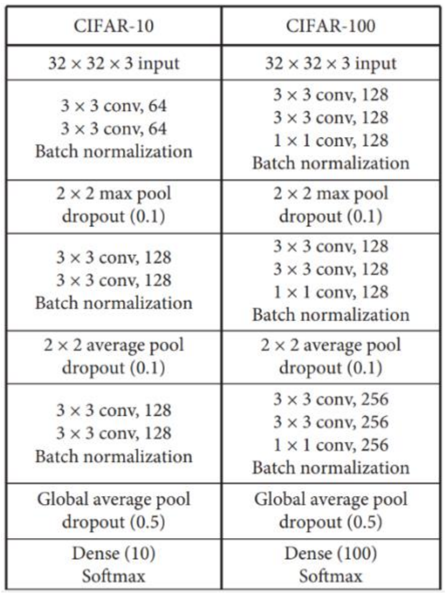


Figure 5 flow chart of training and testing of DCNN[7]



CIFAR-10	CIFAR-100
32 × 32 × 3 input	32 × 32 × 3 input
3 × 3 conv, 64 3 × 3 conv, 64 Batch normalization	3 × 3 conv, 128 3 × 3 conv, 128 1 × 1 conv, 128 Batch normalization
2 × 2 max pool dropout (0.1)	2 × 2 max pool dropout (0.1)
3 × 3 conv, 128 3 × 3 conv, 128 Batch normalization	3 × 3 conv, 128 3 × 3 conv, 128 1 × 1 conv, 128 Batch normalization
2 × 2 average pool dropout (0.1)	2 × 2 average pool dropout (0.1)
3 × 3 conv, 128 3 × 3 conv, 128 Batch normalization	3 × 3 conv, 256 3 × 3 conv, 256 1 × 1 conv, 256 Batch normalization
Global average pool dropout (0.5)	Global average pool dropout (0.5)
Dense (10) Softmax	Dense (100) Softmax

Figure 6 the model of two special architecture DCNNs[7]

Every convolution layer was followed by a batch normalization layer[41], which was followed by an activation function. Instead of fully connected layers, which are often employed in conventional networks, a global average pooling layer [42] was utilized to tackle the "overfitting" problem. However, the category probability was produced using the last fully connected layer with a softmax function. The learning rate fell exponentially with a decay rate of 0.9 as the training repetitions progressed [7]

For the platform and hardware, the CAFFE deep learning framework [43] was used for all training and testing procedures of DCNNs, based on a workstation with a Core i7-8700k CPU, NVIDIA GeForce GTX 1080 GPU, 16 GB of RAM, and 1 TB of storage. It was clear that the hardware platform and framework had just a little influence on the training efficiency of the deep learning model rather than the actual classification performance. Fivefold cross-validation results were computed for final evaluation and comparison to prove the validity of the introduced full-stage data augmentation. Moreover, the classification results of both datasets were shown individually in terms of the degree of refinement of item classifications [7].

For results of coarse-grained image classification, first, the bassline classification results were reported before and after the usage of the full-stage data augmentation method. The results showed that the full-stage data augmentation framework pointed to an apparent enhancement in the classification accuracy for the DCNN model. CIFAR-10's average classification accuracy had improved from 85.7 percent to 93.4 percent. Furthermore, data augmentation in the training phase was more successful than in the testing phase, increasing accuracy by around 3%. The performance of the standard training data augmentation approach was improved by using the complete stage data augmentation framework without paying more cost compared to the latest methods, the results showed that the proposed method had essentially enhanced the classification accuracy from 89.59% to 93.41% [7].

The average classification accuracy of CIFAR-100 before and after applying the whole-stage data augmentation approach increased from 62 percent to 70 percent, which is higher than that of CIFAR-10. Using a whole-stage data augmentation architecture, classification accuracy was increased from 64.32 percent to 69.22 percent when compared to current approaches. [7].

Due to multiple tasks in AI and machine learning, DNN has been developed as the technique of choice for solving these tasks. The design of DNN accelerators is a key challenge for architectural designers. On the other hand, these accelerators require enormous amounts of multiply-accumulate and on-chip memory and are restrictive in the zone and cost-obligated frameworks, for example, wearable gadgets and IoT sensors [8]. The authors of [8] aimed to achieve DNN performance on GPP cores by utilizing a key feature of DNNs, namely sparsity, or the prevalence of zero values. Moreover, the authors introduced Sparsity-aware Core Extensions (SPARCE), a collection of low-overhead micro-architectural and ISA extensions that may effectively determine if an operand (e.g., the result of a stack instruction) is zero and so bypass a series of subsequent operations that rely on it. [8]. Sparsity in the error data structure is illustrated in **Error! Reference source not found.**

$$\frac{\partial E}{\partial y_l}(x+p, y+q) = \begin{cases} 0, & if\ y_{l+1}(x,y) \neq y_l(x+p, y+q) \\ \frac{\partial E}{\partial y_{l+1}} & othewise \end{cases} \qquad (22)$$

To enlarge performance benefits, SPARECE guaranteed that the commands to be skipped are avoided from indeed being brought, as squashing commands comes with penalties. SPARECE had two key micro-architectural improvements. They are a Sparsity Register File (SpRF) and Sparsity-Aware Skip Address (SASA). SPARCE was modeled using the gem5 architectural simulator. The register operand (Rn) points to the SASA location in memory [8] as shown in

$$SASA\text{-}\ LD\ [R_n],\ \#size \qquad (23)$$

The model was compared with 6 of the latest models of image recognition DNNs in image, video, text, and speech processing [44], [45], [46], in both training and inference by using the “CAFFE deep learning framework”.

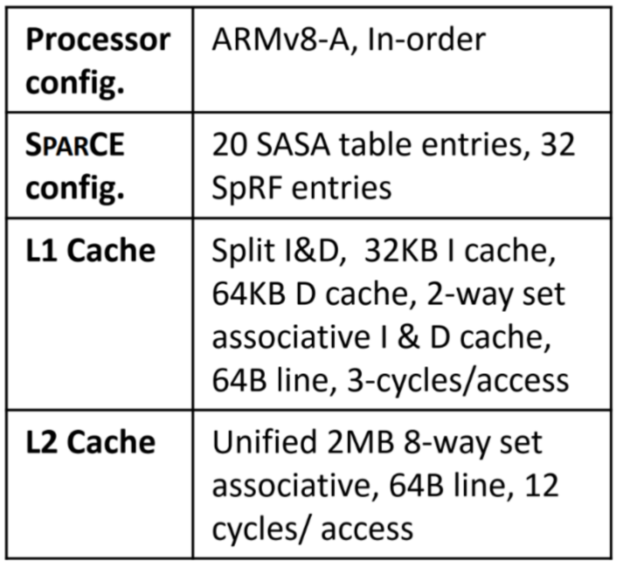
The authors of [8] introduced micro-architectural in SPARCE. It was modeled using a cycle-accurate gem5 simulator [47]. The model was integrated into the “CAFFE deep learning framework”. CAFFE was used to create appropriate matrices for each layer, which were then input into the gem5 simulator for matrix calculations. CAFFE received the findings and used them to create the next layer of inputs. Gem5 simulation parameters are illustrated in SalvagingDNN accelerators algorithm is shown in **Error! Not a valid bookmark self-reference.**

Figure 7 [8].

SPARCE was tested in two scenarios by monitoring application-level runtime:

The first one was aimed at embedded scalar processors, which are utilized in low-power edge/IoT applications and require high-performance libraries to run. The used processor was ARM v8 as the baseline in-order processor architecture. A direct convolution routine, named (DIR-CONV-Scalar), was prototyped and used in the experiments. [8]

The second was targeting a reasonably sophisticated mobile processor. Why The in-order processor architecture ARM v8 was chosen as the baseline, with 4-way SIMD as the default. The matrices were calculated with the highly optimized OpenBLAS software. [48] based on “GEMM”, which is called OpenBLAS-SIMD4[8].



Processor config.	ARMv8-A, In-order
SPARCE config.	20 SASA table entries, 32 SpRF entries
L1 Cache	Split I&D, 32KB I cache, 64KB D cache, 2-way set associative I & D cache, 64B line, 3-cycles/access
L2 Cache	Unified 2MB 8-way set associative, 64B line, 12 cycles/ access

Figure 7 Gem 5 simulation parameters[8].

SPARCE was implemented at RTL using VHDL and it was synthesized to IBM 45nm technology using Synopsys compiler. The design occupied only 1.04% of the ARM Cortex A35 core area. So, the area overhead of SPARCE was somehow negligible permitting its sending within the resource-constrained embedded platforms. [8]

SPARCE was compared to 6 of the latest image-recognition DNNs as shown in **Figure 8 Application benchmarks**Figure 8. To obtain the performance of SPARCE under inference conditions, pre-trained models from the CAFFE Model Zoo were employed [8].

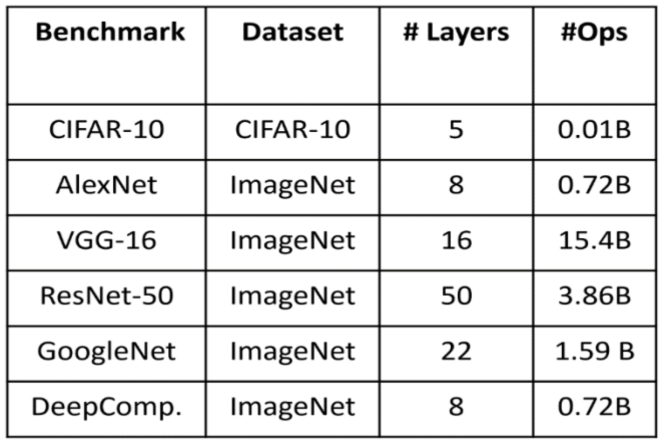
The CIFAR-10 benchmark was only utilized for training since training ImageNet models on gem5 was too time-consuming. Except for AlexNet, which had static sparsity in weights, all benchmarks demonstrated dynamic sparsity in features and errors. [8].

From a performance and energy enhancement point of view, the comparison result is shown in Figure 9.

Results illustrated that for Dir-Conv-Scalar, application runtime was reduced by between 19% and 31% all over the benchmark. While for OpenBLAS-SIMD4, the profit in the runtime reduction is between 8%-15%.

The AlexNet [49] benchmark reached the maximum profits because Its features and weight are both sparse, in contrast to other benchmarks that have a dense weight data structure. As compared to forward propagation, the backpropagation step improved better throughout training. This is because the error data structure is more sparse than the features data structure.[8]

Practical shape acquisition and evaluation are considered challenging tasks, especially when dealing with raw images with complex backgrounds [9].



Benchmark	Dataset	# Layers	#Ops
CIFAR-10	CIFAR-10	5	0.01B
AlexNet	ImageNet	8	0.72B
VGG-16	ImageNet	16	15.4B
ResNet-50	ImageNet	50	3.86B
GoogleNet	ImageNet	22	1.59 B
DeepComp.	ImageNet	8	0.72B

Figure 8 Application benchmarks[8]

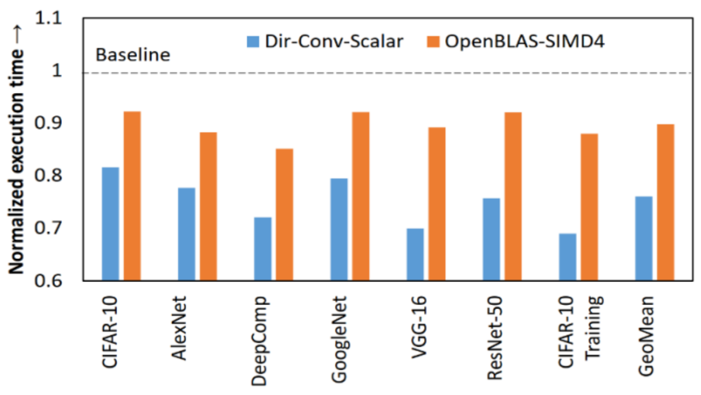


Figure 9 application-level execution time enhancement[8]

The authors of [9] aimed to develop a systematic framework for practical extraction and shape analysis using DCNN (lightweight U-net [50],[51]) and digital image processing. The authors of [9] built the network according to the following steps as illustrated in Figure 10

1. Crop and label manually raw images of particles.
2. Train the neural network.
3. Using the well-trained network, extract useful projections from photos of any size with a complicated backdrop.
4. Use the enhanced erosion and flood filling method and the B-spline curve technique to separate and smooth the practical boundaries.
5. Separate and smooth the practical borders using the increased erosion and flood filling method and the B-spline curve methodology.

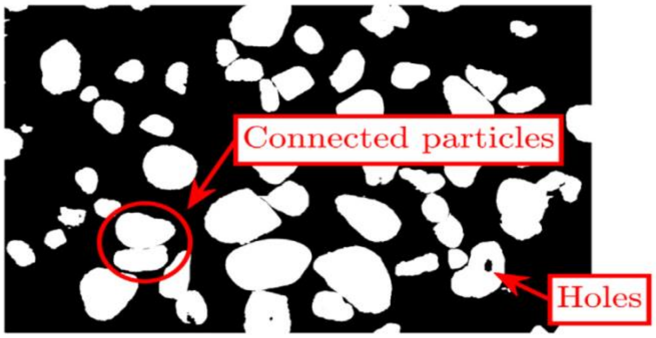
The authors of [9] employed the following: Convolution operation is the core of the network, as illustrated in Figure 11 firstly the input matrix shown in black was stretched and padded with zero elements. After that, a kernel was slid onto the stretched input matrix with stride (s=1) in the x and y directions. Feature map elements (yij) were obtained by

$$y_{ij}=AF(\sum_{m=i-S}^{i+S}\sum_{n=j-S}^{j+S}(x_{mn}w_{mn}+b) \qquad \text{where } S=(k_n-1)/2 \qquad (24)$$

The MaxPool operation applied in this experiment is defined by using k_n=2, S =2 replacing all (24) as maxout function [9] as shown in

$$y_{ij}=max(x_{mn})m=\{2j-1,2j\} \qquad (25)$$

The proposed lightweight U-net is illustrated in



. Downblocks, upblocks, and bottom layers are the three sorts of components that make up this structure These blocks are used to decode and encode the input matrix[9].

The batch normalization and the dropout operations are combined together to get the following benefits: 1) reducing the number of parameters of the network and 2) network accuracy will not be destroyed due to the modification. 3) If the GPU used for training various versions of the network was the same, the lightweight U-net might reach a higher accuracy than the original version did[9].

Preparing training data is a requirement for network training. Four samples are captured in this experiment to obtain raw images of a gravel-sand mixture with 3648x2432 pixel dimensions. To obtain complete binary masks, the particle projections are manually labeled from the original photos. With the sliding window, crop into small-size images/masks both full-size binary masks and raw images as shown in Figure 13[9]

The small-size images are processed via multiple data augmentation techniques after the cropping, to increase the amount and variety of training data. 90% of the images were haphazardly chosen to train the network as a training set, while the rest 10% were used as a testing set for accuracy evaluation[9].

The selection of an acceptable loss function and the use of techniques to reduce it were the two key tasks of the network training process. The used loss function was the weighted binary cross entropy (WBCE) function[52]. The ADAM [53] training algorithm was used to minimize the loss function[9]

As illustrated in

$$\text{WBCE} = \text{mean}\,(l_m * w_m) \qquad (26)$$

Where :

$l_m = SM(-GT.*EW_{log}(O_m+\mathcal{E})-(1-GT).*\ EW_{log}(1-\ O_m+\mathcal{E}))$

$w_m = 1-\beta+(2\beta-1)*GT$

$\beta = 1-\text{mean}(GT)$

There were two issues with the full-size masks that would affect the shape analysis accuracy, which as shown in Figure 14 **issues on output full-size mask** are: 1) some isolated holes can be found in particle portions; and 2) some neighboring particles may connect together and must be isolated before the shape analysis. The first issue could be solved using the classical Fillhole [54] algorithm. Several morphological image analysis techniques [55], [55]could solve the second issue. [9]

Shape analysis depended on DIP[56],[57]. Removing noises within limits was an essential prerequisite. These noises on particle limits were gotten from various sources, e.g., "rough particle surface and low-resolution particle images[58]". There are many techniques to achieve noise elimination, such as the locally weighted regression (LOESS) smoothing method[56] or filter techniques[57]. The B-spline curve method was used here because other techniques, e.g., the LOESS method and filter method, were designed for star-like particles, the polar span of which consistently has just a single convergence point with the limit. Although the B-spline curve can reconstruct both star-like and nonstar-like molecules[59].

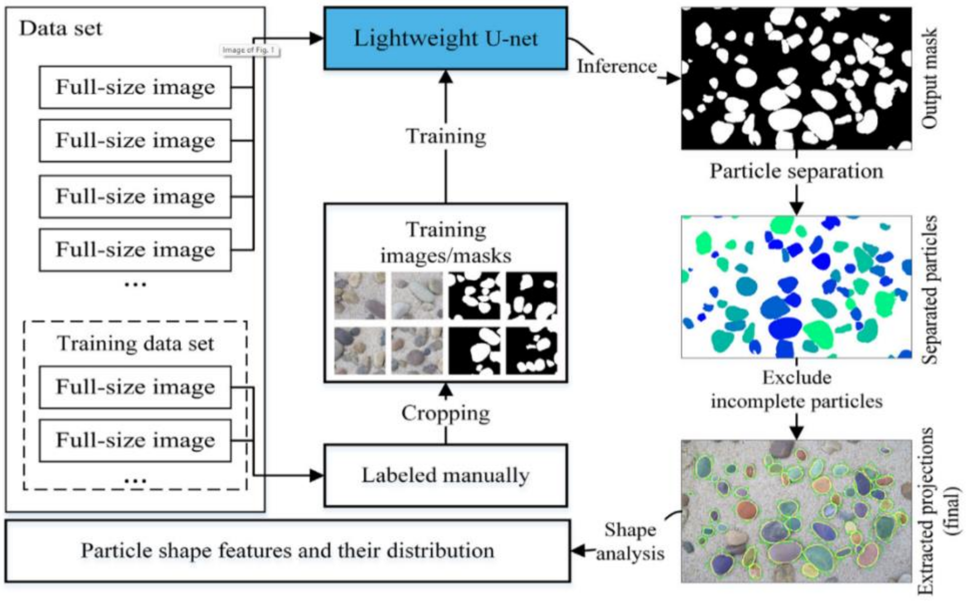
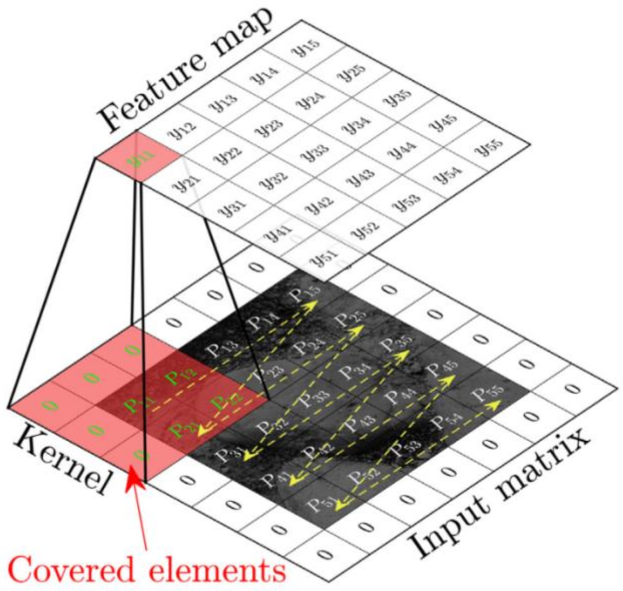
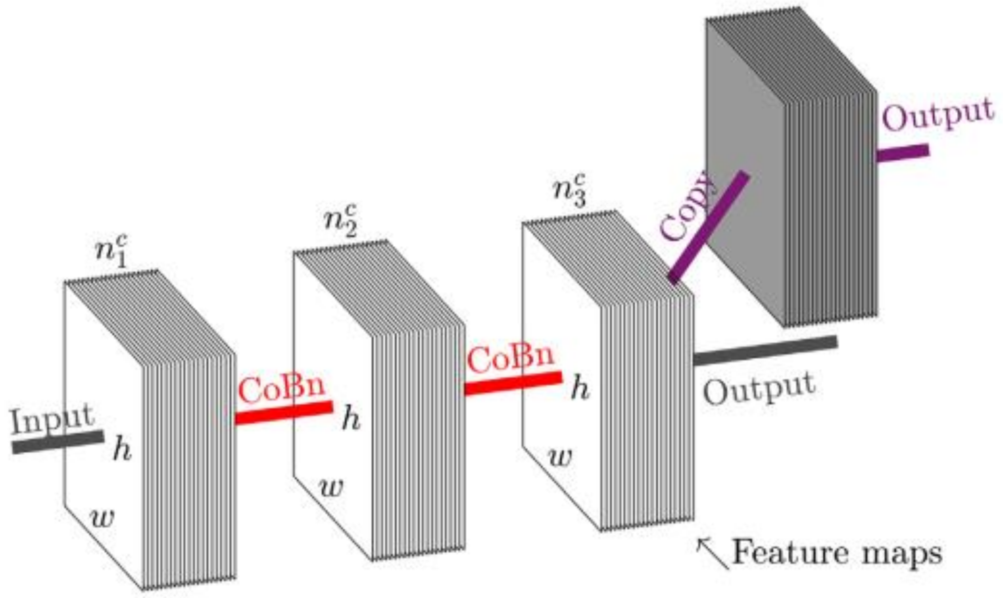


Figure 10 the workflow[9]

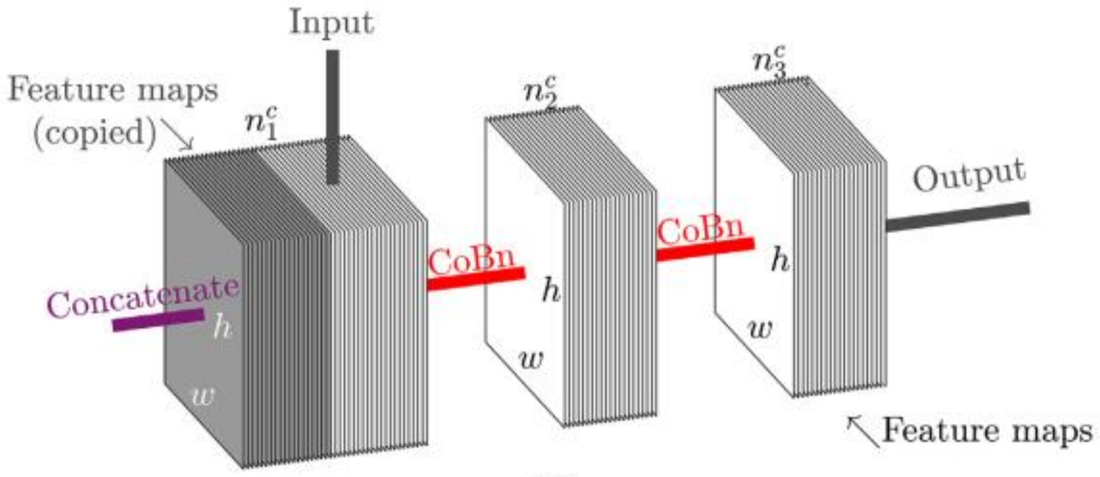


.

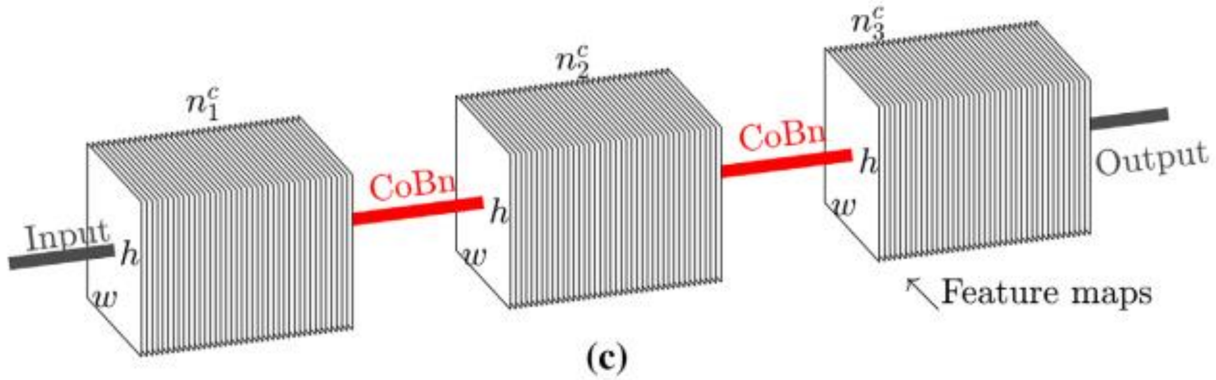
Figure 11 the convolution operation (For clarity, the pixel grids have been expanded) [9]



(a)



(b)



(c)

Figure 12 (a) U-net downblock architecture, (b) U-net upblock, (c) bottom layers, [9]

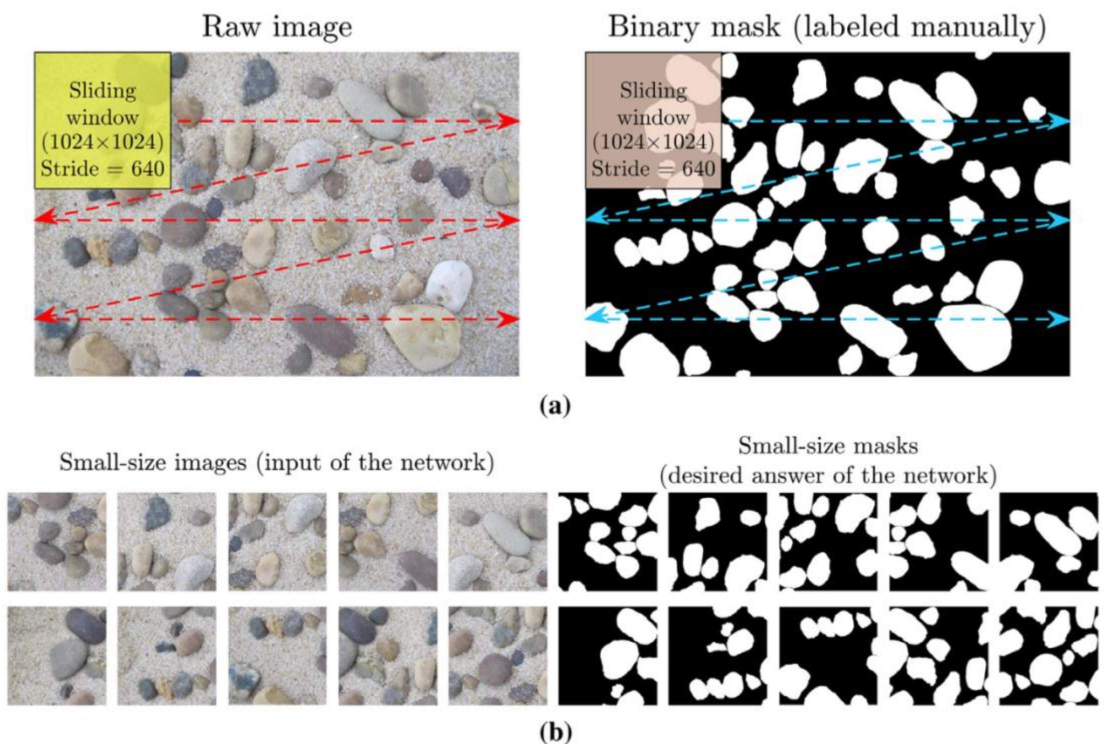

Raw image
Binary mask (labeled manually)
Sliding window (1024×1024) Stride = 640
Sliding window (1024×1024) Stride = 640
(a)
Small-size images (input of the network)
Small-size masks (desired answer of the network)
(b)

Figure 13 (a) Cropping full-size images and masks into training images and masks (b) images/ cropped to a smaller size[9]

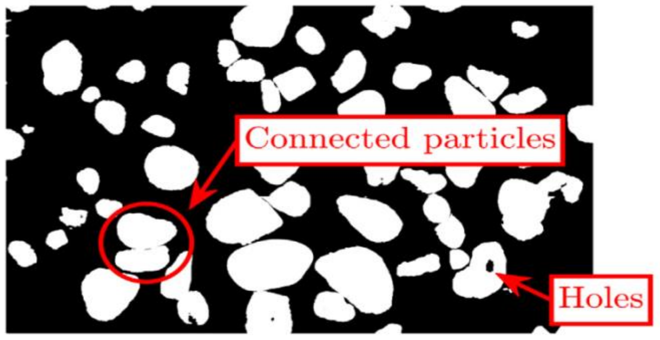


Figure 14 issues on output full-size mask[9]

Shape analysis depended on DIP[56],[57]. Removing noises within limits was an essential prerequisite. These noises on particle limits were gotten from various sources, e.g., “rough particle surface and low-resolution particle images[58]”. There are many techniques to achieve noise elimination, such as the locally weighted regression (LOESS) smoothing method[56] or filter techniques[57]. The B-spline curve method was used here because other techniques, e.g., the LOESS method and filter method, were designed for star-like particles, the polar span of which consistently has just a single convergence point with the limit. Although the B-spline curve can reconstruct both star-like and nonstar-like molecules[59].

Deep convolutional neural networks (DCNNs) have achieved groundbreaking success in image classification and multiple applications of vision in recent years. DCNNs have an upper hand over conventional solutions in that they provide a consistent feature extraction and classification system that relieves users of the time-consuming task of manually extracting features. DCNNs, on the other hand, are far from self-sufficient, as their efficiency is heavily dependent on handcrafted architectures, which require a great deal of skill and experience to build, and can no longer be enhanced until hyper-parameter tuning converges[10]. The authors of [10] aimed to autonomously generate a DCNN model, using the autonomous and continuous learning (ACL) method for every single vision mission. The authors of [10] convert the architecture of a DCNN into an integer code by partitioning it into several convolutions, pooling, completely linking, batch normalization, activation, and drop-out procedures may all be found in stacked meta-convolutional blocks and fully linked blocks To evolve a population of DCNN architectures, the authors of [10] use genetic evolutionary operations such as selection, mutation, and crossover. The algorithm was tested on six different image classification tasks. They are to say, “MNIST”, “Fashion-MNIST”, “EMNIST-Letters”, “EMNIST-Digits”, “CIFAR10” and “CIFAR100”. The results show that if more time is required, the proposed ACL algorithm can evolve the DCNN architecture continuously and can find a suboptimal DCNN architecture with comparable performance to the state of the art.

The authors of [10] employed the following equation:

A DCNN which has N_n^C as convolution blocks and N_n^F as fully connected blocks are expressed in **Error! Reference source not found.**) and code length in **Error! Reference source not found.**)

$$S_n = \{\{[NSPBAD]_i\}_{i=1}^{N_n^C}, \{[NBAD]_j\}_{j=1}^{N_n^C}, O\} \qquad (27)$$

$$L_n = N_n^C * l_c + N_n^F * l_f \qquad (28)$$

$N_n^C \epsilon [1,20]$ and $N_n^F \epsilon [1,20]$ at the initialization step.

In the case where the location of the cross point k_i is within the m_i-th convolutional block [NSPBAD]mi, the two new DCNNs code lengths after the crossover[10]. are expressed in

$$\begin{cases} L_i' = L_i + (m_i - m_j) * l_c \\ L_j' = L_j + (m_j - m_i) * l_c \end{cases} \qquad (29)$$

In the case where the location of the cross point k_i is within the m_i-th convolutional blocks [NBAD]m_i, the two new DCNNs code lengths after the crossover [10]. are expressed in

$$\begin{cases} L_i' = L_i + (m_i - m_j) l_f \\ L_j' = L_j + (m_j - m_i) l_f \end{cases} \qquad (30)$$

A population is evolved and improved using the theoretical genetic DCNN designer of individuals, each encoding a DCNN architecture that is admissible. The population is started at random. This algorithm is summarized in Figure 15 [10].

The introduced genetic DCNN designer was utilized to generate a DCNN for each dataset that could handle the relevant picture classification issue with acceptable accuracy. Every dataset's training data were randomly divided into two portions for each experiment: The validation set accounts for 10% of the data, while the remainder was utilized for training. **ImgeDataAugmentation** tools provided by Keras [60] were used to magnify the training set, which included cropping by no more than 12.5% of width and height, rotation at 90^o, 180^o, and other random rotation by no more than 5^o to get augmented copies for training images[10].

For all DCNN designs, the **cross-entropy loss function** was employed with a maximum iteration number of 100, the min-batch stochastic gradient descent with a batch size of 256, and the learning rate was set as low as 0.0001 and progressively reduced exponentially[10].

It reveals that the DCNN designed by the authors of [10] genetic DCNN designer had a maximum accuracy of 99.5%, which was higher than the DCNN developed by EXACT (98.32 percent).

The best DCNN architecture's performance generated by the authors of [10] genetic DCNN designer has become stable, as shown in Figure 16 by the plot of the highest classification accuracy achieved in each generation.

Deep convolution neural network (DCNN) has multiple important open issues in many applications, like the classification of remotely sensed images. One of the most important issues is choosing suitable scales for these images.

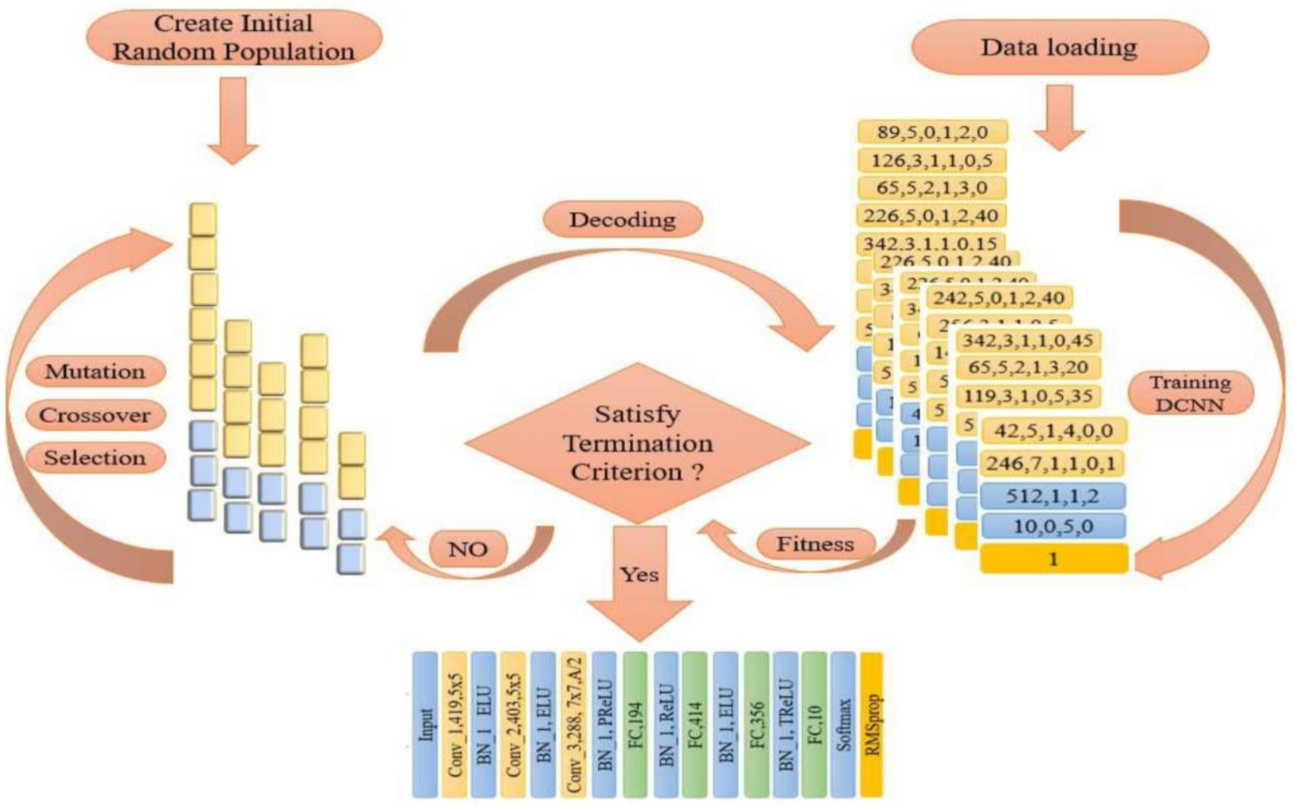


Figure 15 Diagram of the proposed genetic DCNN designer[10]

That is because of the effect of the spatial scale on the ground targets' recognition [11]. It is a critical issue, especially when used in the classification of large-scale "land use" (LU) and "land cover" (LC) jointly [61]. The authors of [11] aimed to present a simple and sparing "Scale Sequence Join Deep Learning" (SS-JDL) technique for joint LU and LC classification in an automatic method. In this technique, an arrangement of scales is installed in the iterative cycle of fitting the joint distribution implicit in the "joint deep learning" (JDL) method, thus supplanting the past paradigm of scale choice.

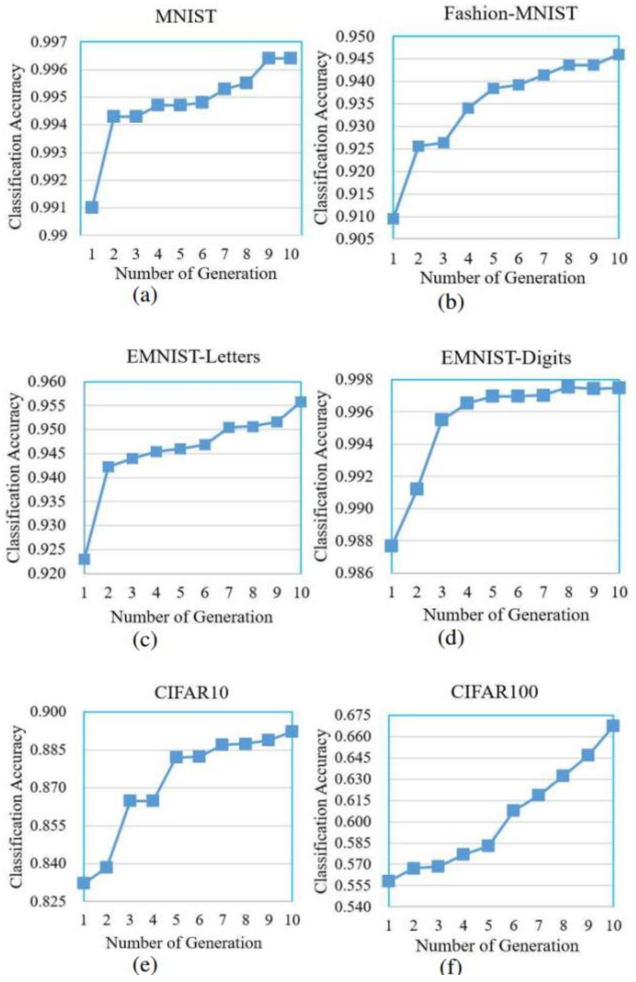


Figure 16 Highest classification accuracy achieved in each generation on ach dataset[10]

To determine the CNN input patch size, the sequence of scales is required to infer self-sufficiently and successive transmission is required from small-scale features to large-scale characterization, and information from simple LC states to complex LU perspectives. Aerial digital photography was used to test SS-JDL's effectiveness. This photograph represents three complex and heterogeneous scenes, two of them in Southern England and one in North West England (Manchester) [11]

The introduced method had two significant perspectives, which were how to create and utilize a "scale sequence" and "joint learning" between the LU and LC forecast at each scale within the scale order. The general architecture of the introduced SS-JDL method is shown in Figure 17, in which LU and LC classifications were jointly inferred over the scale sequence[11].

In this method, the authors of [11] employed the following: a scale sequence (S) is used to characterize LU and LC across many scales. S is obtained as in(31) where the function of linear interpolation is referred to as Linespace and θ_{min}, is the minimum scale and θ_{max} is the maximum scale.

$$\boldsymbol{S}=Linespace\ (\theta_{min},\theta_{max},\mathrm{n}) \quad (31)$$

The LU classification probability is influenced by the LC classification probabilities, while the i-th iteration's results are influenced by the preceding iteration's results. [11].as shown in

$$P\ (LU(\theta)^{i},\ LC^{i}\ )=P(LU\ (\theta)^{i},\ LC^{i}\mid LU(\theta)^{i-1},\ LC^{i-1}\ ) \quad (32)$$

Land cover classification (MLCi) is obtained [11] as described in

$$M_{LC}{}^{i}=Concate\ (M,LU(\theta)^{i-1}) \quad (33)$$

The MLP model is trained through the LC training samples (TLC) [11] as shown in

$$\mathrm{mlpmodel}^{i}=\mathrm{MLP.Train}\ (\mathrm{M}_{LC}{}^{i}\ ,\ \mathrm{T}_{LC}) \quad (34)$$

Prediction of LC classification probability (MLCproi)is obtained by using the MLP model (mlpmodeli) [11] as shown in

$$M_{LCpro}{}^{i}=mlpmodel^{i}.\ predict\ (M_{LC}{}^{i}) \quad (35)$$

Using TLU, a CNN model is created[11]. as shown in

$$\mathrm{cnnmodel}^{i}=CNN.\ Train(M_{LU}{}^{i},\boldsymbol{T}_{LU}\ ,\ \theta^{i}) \quad (36)$$

The LU classification probabilities (MLUproi) are obtained by using cnnmodel [11]as shown in

$$M_{LUpro}{}^{i}=cnnmodel^{i}.\ Predict(M_{LU}{}^{i}\ ) \quad (37)$$

Every iteration yields probability for land cover (MLCproi) and land usage (MLUproi). By outputting the greatest probabilities, the probabilistic land cover (MLCpron) and land use (MLUpron) classes are turned into the comparable LC (MLCresult) and LU (MLUresult) classes..as shown in

$$M_{LCresult}=argmax(M_{LCpro}{}^{n}). \quad (38)$$

$$M_{LUresult}=argmax(M_{LUpro}{}^{n}) \quad (39)$$

The datasets of the elected areas are aerial photos with a spatial resolution of 50 cm and a spectral resolution of 4 bands (R, G, B, and NIR). The datasets were split randomly into 60 % for training and 40 % for validation[11].

In the methodology, SS-JDL, parameters of MLP and OCNN classifiers should be predefined to get the best classification accuracy. Both models were parameterized as in [62] and [63].

The scale size for SS-JDL was set to 28x28 as the lowest scale and 140x140 as the maximum scale due to the object information. To create a scale sequence, a variety of scales between the lowest and maximum were interpolated into the network. The number of iterations grew as more scales were incorporated. Using the SS-JDL and JDL approaches, Figure 18 illustrates the influence of iterations on overall accuracy for both LU and LC [11].

For the range of different scales, **Error! Reference source not found.** shows, employing SS-JDL and JDL approaches, how the window size influenced the overall accuracy of the LU and LC classifications.

The JDL classification provided a forward scale sequence (FSS) based on the minimum and maximum sizes. Table 6 shows the predominance of FSS compared to IGSS, RSS, and BSS, and the overall accuracy and efficiency.

For a long time, there have been a huge quantity of high spatial-resolution remote sensing (HRRS) images accessible for land-cover mapping. In any case, finding an efficient method for achieving precise land-cover classification with high-resolution and heterogeneous inaccessible sensing images is frequently difficult due to the increased information complexity provided by increasing spatial resolution and data unsettling influences caused by different conditions of image acquisition.[12]. The authors of [12] aimed to suggest an architecture to put in To categorize unlabeled HRRS images, a deep model was constructed using a labelled land-cover dataset.

The concept of this method relies on DNN to provide relevant information presented in various forms of land cover and to offer a pseudo-labeling and sample selection approach to improve the generalizability of deep models. ADCNN is initially pre-trained with a very well-characterized land-cover dataset, referred to as the original dataset. After that, the pre-trained CNN model is used to categorize a target image that has no labels in a patch-wise way[12].

Table 6 The accuracy and computing time of four sampling strategies, including forward scale sequence (FSS), backward scale sequence (BSS), random scale sequence (RSS), and iterative greedy scale sequence (IGSS), were compared (IGSS)

Sampling scheme	Overall accuracy (%)			Computational time (h)
	S1 (LC, LU)	S2 (LC, LU)	S3 (LC, LU)	S1, S2, S3
FSS	91.06, 88.94	90.43, 88.26	90.62, 88.48	7.52, 7.86, 7.32
BSS	86.73, 83.84	86.68, 83.05	87.04, 84.26	7.52, 7.86, 7.64
RSS	87.24, 84.32	87.59, 84.13	87.74, 83.85	8.95, 9.37, 9.28
IGSS	90.35, 87.69	89.76, 87.14	89.43, 87.25	35.58, 37.94, 36.65

The authors of [12] build a hybrid classification by combining patch-wise classification and hierarchical segmentation with the target image to get pixel-wise land-cover classification They also created a large-scale land-cover dataset consisting of 150 Gaofen-2 satellite images for CNN pre-training.

Several strategies for analyzing RS images using spectral and spectral-spatial properties to classify the image composition of various land-cover categories have been investigated [64], [65], [66], [67], [68], [69], [70]. Due to the rich and structural information offered by the continually growing spatial resolution, spectral and spectral-spatial characteristics have difficulty explaining the contextual information included in the pictures. [71], [72], [73], [74].

A method of training transferable deep models was introduced by the authors of [12] that can be used to classify land cover using high spatial resolution unlabeled RS images from multiple sources. Furthermore, the authors of [12] design a hybrid land-cover classification system that may be used to classify multiple types of land cover at the same time. It is possible to extract precise category and boundary information from HRRS images Experiments on multi-source HRRS images, such as Gaofen-2, Gaofen-1, Jilin-1, Ziyuan-3, Sentinel-2A, and Google Earth platform data, show that the proposed technique is effective.

The authors of [12] provided a large-scale land-cover classification dataset, that is, GID. GID is made up of 50 high-resolution Gaofen-2 images that cover 50,000 square kilometers in China.

The authors of [12] presented the following method to efficiently classify land cover using multi-source HRRS images: efficiently classify land cover. This new method is used to construct transferable deep models that have been pre-trained on labelled land-cover datasets but can now be utilized with unlabeled HRRS images. The authors of [12] made the following assumption: a large-scale dataset that has been well-annotated and a newly collected image that needs labeling data There were two domains defined: source domain DS and target domain DT.

The authors of [12]used DS to train a deep model for the RS domain beforehand. Convolutional layers, pooling layers, and fully-connected layers are the three layers that make up the DCNN model. The hierarchical feature extractors are the convolutional layers; the spatial down-sampling of feature maps is the pooling layer; and finally, classifiers are the fully-connected layers that generate predictive classification probabilities from the input data. Residual Networks (ResNet) [75]

The authors of [12] utilize ResNet-50, as illustrated in Figure 20, as the classifier for a simple vs. computationally efficient classifier. This is a trade-off to think about.

As a consequence of technological changes in acquisition conditions, classification results on multi-source RS images are not satisfactorily achievable by CNNs, although they can generalize to a certain extent. To establish a more precise classification rule than relying just on source material that has been labeled, the authors of [12] utilize available information from the unlabeled target data to CNN models, transferring for RS image identification, which is recorded under varied circumstances.

A semi-supervised transfer learning model was introduced by the authors of [12] to classify multi-source RS images. The architecture of the model is illustrated in Figure 21. There are two steps to the model. The first one is ” pseudo-label assignment” [76] and the second one is “relevant sample retrieval". By using the source domain training set with the relevant specimen, joint fine-tuning improves the classification model [77] [78].

The authors of [12]employed (40) as the output vector of the softmax layer in Pseudo-label assignment

$$\boldsymbol{P}_i = \{P_{i,1}, P_{i,2}, \ldots\ldots, P_{i,K}\}, \qquad \boldsymbol{P}_i \in \boldsymbol{R}^K \quad (40)$$

Where $P_{i,K}$ indicates the probability that patch x_i is a member of class k, $K \in \ldots \{1, , .. K\}$, and K is the number of classes in totally .

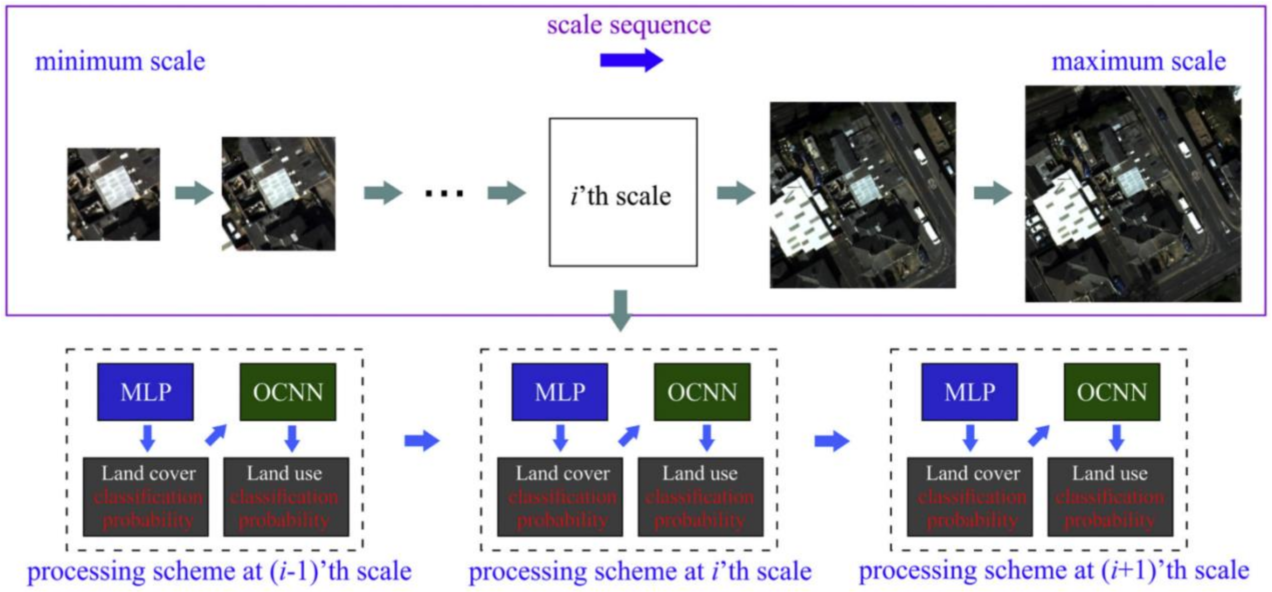

scale sequence
minimum scale
maximum scale
i'th scale
MLP
OCNN
Land cover classification probability
Land use classification probability
processing scheme at (i-1)'th scale
processing scheme at i'th scale
processing scheme at (i+1)'th scale

Figure 17 the general architecture of SS-JDL method for LC and LU classification[11]

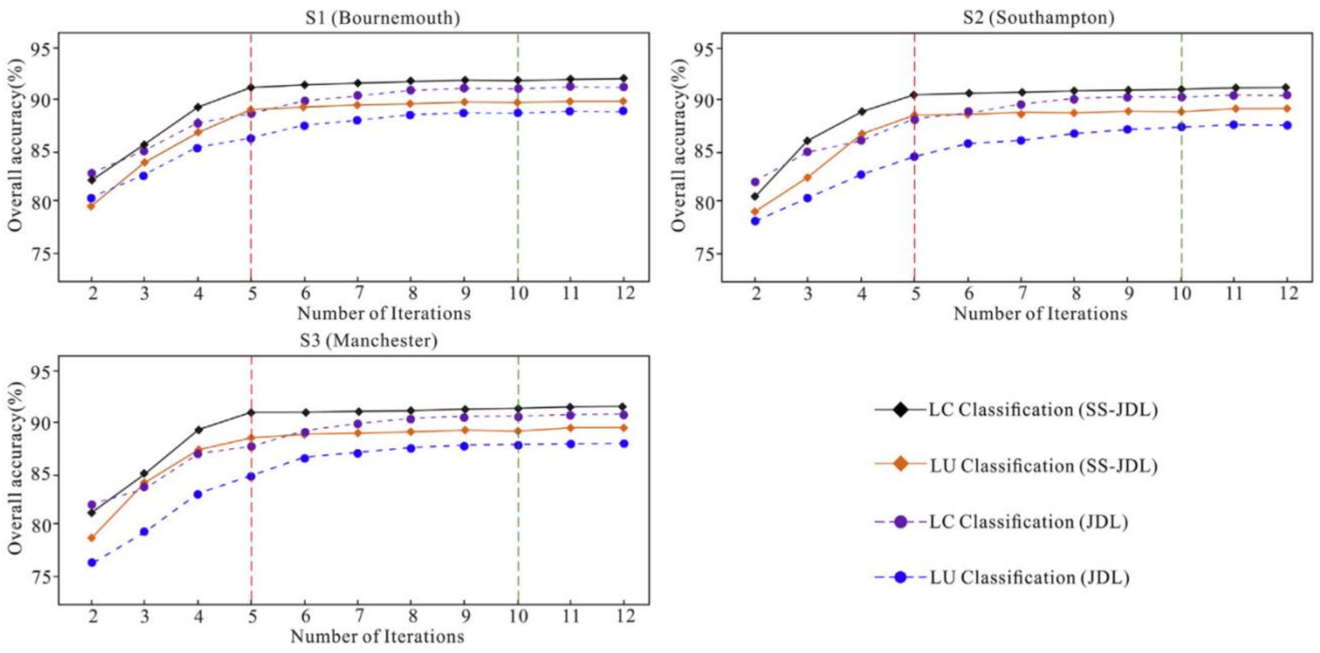


Figure 18 The influence of iteration upon overall accuracy for the LU and LC classifications using the proposed SS-JDL and the JDL method[11].

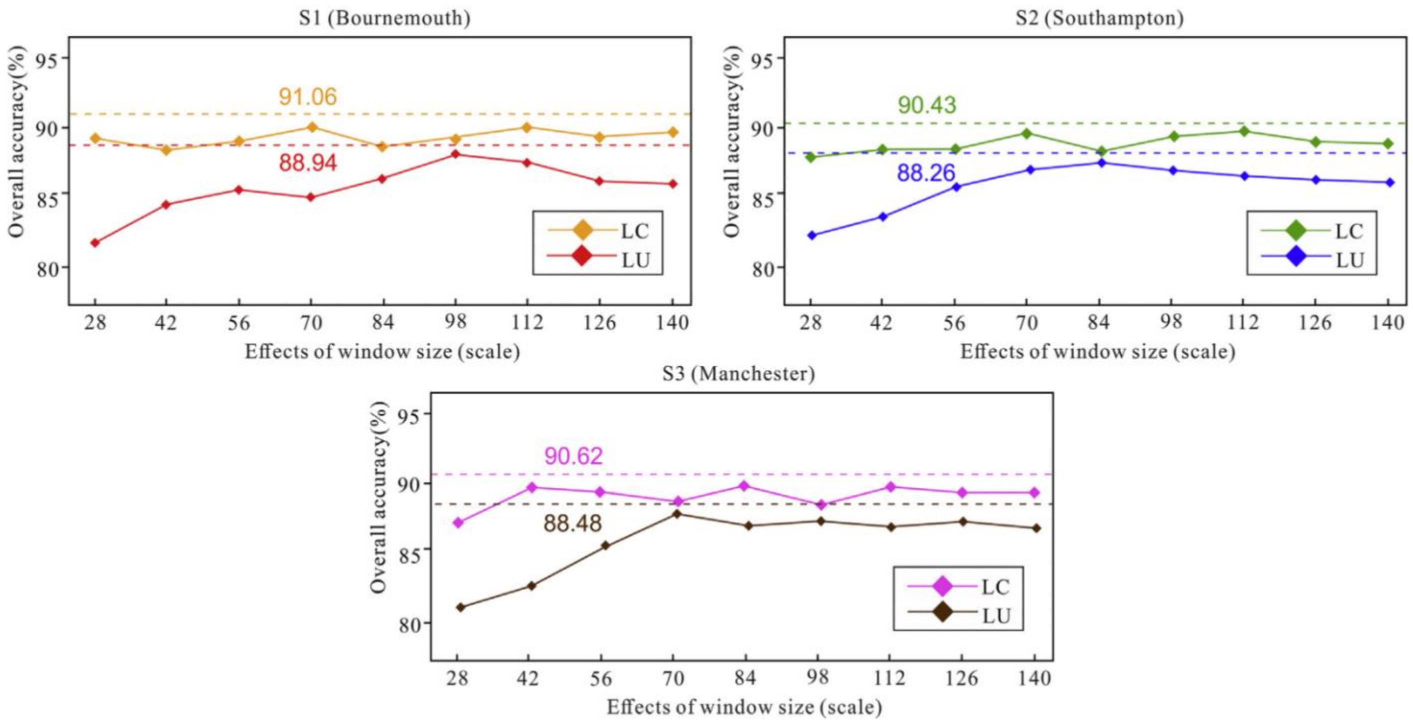


Figure 19 Using the SS-JDL (dashed lines) and the JDL approach, the influence of window size (scale) on the overall accuracy of the LU and LC classifications (solid lines) [11].

For relevant sample retrieval, the authors of [12]employed

$$E_j = -\sum_{K=1}^{K} P_{j,k} . \log(P_{j,k}) \qquad (41)$$

Where $P_{i,K}$ indicates the probability that patch x_j is a member of class k

The classification probability vector [12] is presented in

$$\boldsymbol{P}_{Sn}(\boldsymbol{z}) = \{P_{Sn,1}(z), P_{Sn,2}(z), \ldots\ldots, P_{Sn,K}(z)\}, \quad \boldsymbol{P}_{Sn}(\boldsymbol{z}) \in \boldsymbol{R}^K \qquad (42)$$

The goal of land-cover classification is to give land-cover category labels to pixels in an RS image. For accurate classification, the category and boundary information of the ground objects are both required. As a result, a hybrid architecture was presented by the authors of [12] It combines patch-wise classification and hierarchical segmentation with a majority voting technique, as illustrated in Figure 22.

A weighted fusion approach is used to integrate contextual information from multi-scale patches. The specificity measure [77] describes the degree of certainty with which a classification result can be established and is employed in (43) as the weight: [12]

$$W_{Sn}(z) = \sum_{K=1}^{K-1} \frac{1}{K} (\hat{p}_{Sn,K}(z) - \hat{p}_{Sn,K+1}(z)) \quad .(43)$$

Where $(\hat{p}_{Sn,K}(z)$ represents vector Psn(z)in the descending order Wsn(z) has a range values from 0 to 1, and the greater the value, the more certain the categorization. The weighted probability $\tilde{p}_k$ (z) [12] is presented in

$$\tilde{\boldsymbol{p}}_k(\boldsymbol{z}) = \frac{\sum_{n=1}^{N} W_{Sn}(z).p_{Sn,k}(z)}{\sum_{n=1}^{N} W_{Sn}(z)} \qquad (44)$$

Where $\tilde{\mathbf{p}}_k$ (z) ϵ [0,1] represents the probability that the reference pixel z is a member of class k. The reference pixel z is classified [12] as presented in

$$l(z) = \operatorname{argmax} \tilde{\boldsymbol{p}}_k(\boldsymbol{z}), \quad \boldsymbol{k} \in \{\boldsymbol{1},\ldots,\boldsymbol{K}\} \qquad (45)$$

where l(z) represents the label category of the pixel z. The label l(z) in the patch $\mathbf{x}_i$ is assigned per pixel.

Classification based on multi-scale contextual data is illustrated in Figure 23. It is used to utilize the attributes of the objects and their spatial distribution because of the difficulty of acquiring them from a single-scale observation field relevant object information. [12]

where n ∈ {1,.… , N }, $\mathbf{P}_{Sn}$, k(z) indicates the probability that z is a member of class k at the n-th scale.

The segmentation map obtained from the selective search method [79] was used to get accurate boundary information on the object, to clarify the preparatory classification map. A graph-based technique was utilized to generate a set of primary areas in various color spaces [80]. After that, an algorithm was used to combine small regions repetitively.

A majority voting method was used, by authors of [12], by integrating the category and boundary information. The most often used label T($\mathbf{y}_f$) is given to all pixels in $\mathbf{y}_f$ as illustrated in

$$T\, y_f) = \underset{r\epsilon\{1,\ldots,k\}}{argmax} \sum_{m=1}^{M} sing(l_m = r) \qquad (46)$$

Where sign ( ), represents (true) =1, (false) = 0, and r represents the potential class label.

GID offers a wide range of coverage as well as high spatial resolution. There are two primary branches to it. A large-scale classification set has 150 annotated pixels at the pixel level and a fine land-cover classification set which contains 30,000 multi-scale image patches[12].

GID images were portioned separately, by the authors of [12] into required patch sets using multi-scale sliding windows. The window sizes were chosen regarding the spatial resolution of the data.

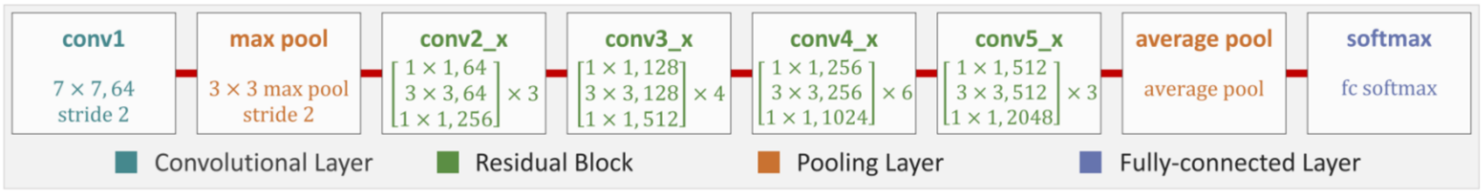


Figure 20 ResNet-50 structure[12]

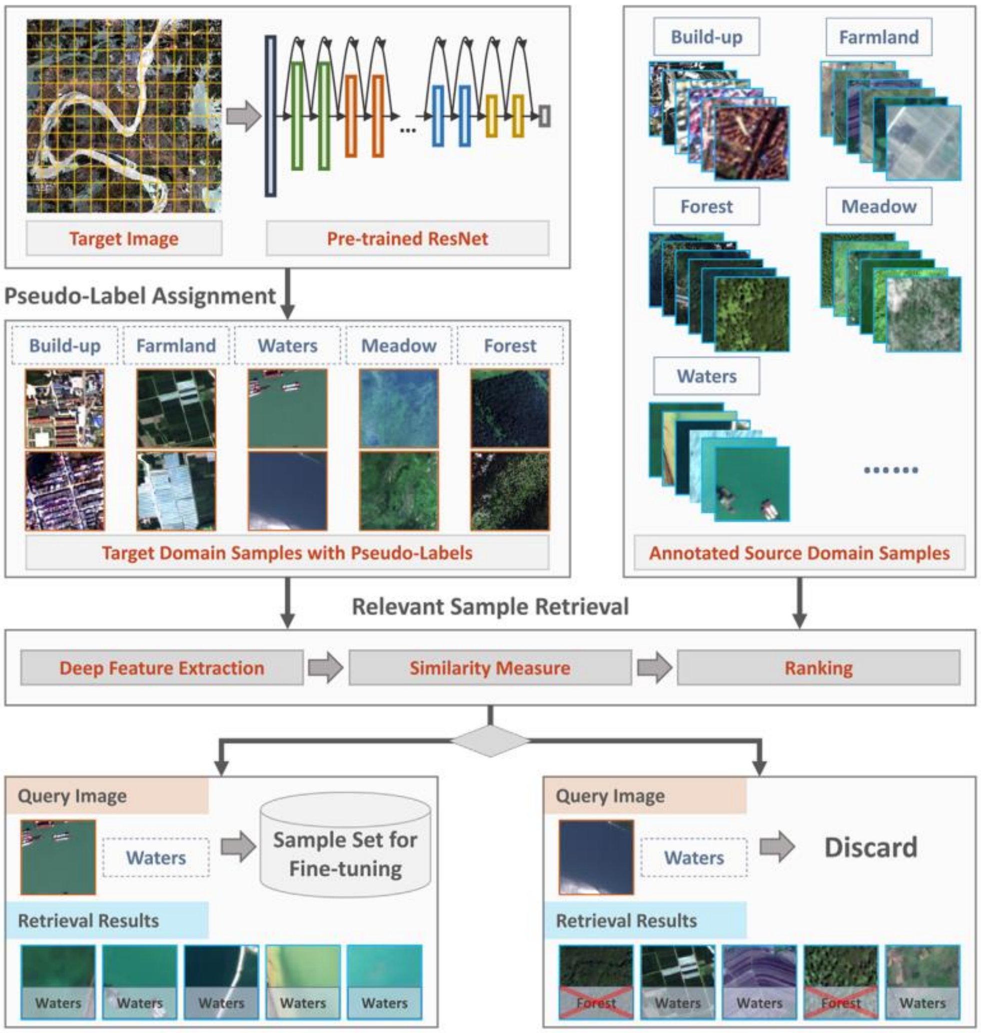


Figure 21 Sample selection for model fine-tuning[12]

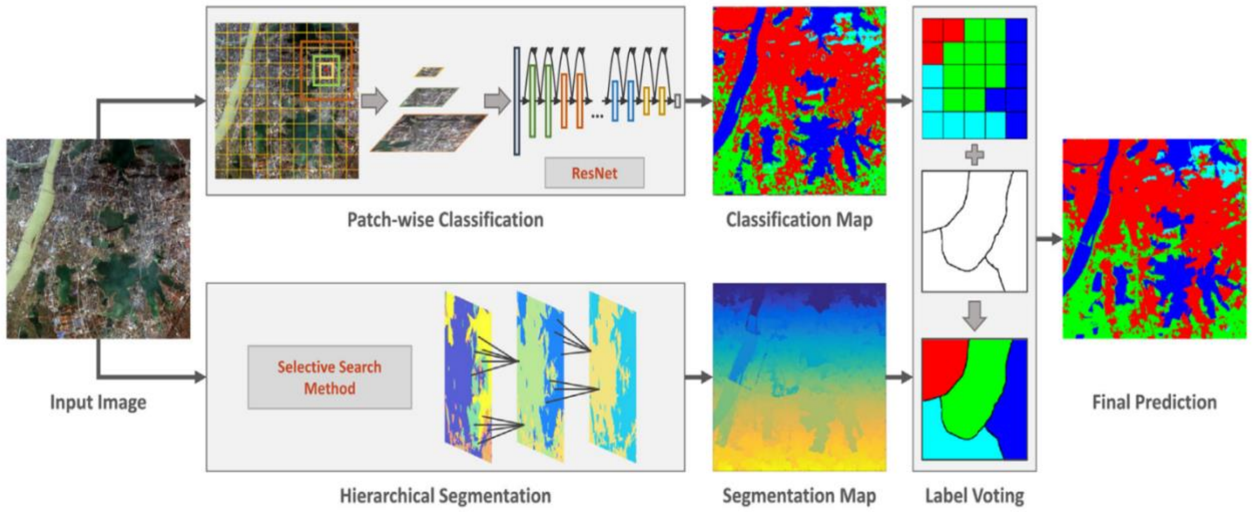


Figure 22 Land-cover classification architecture[12]

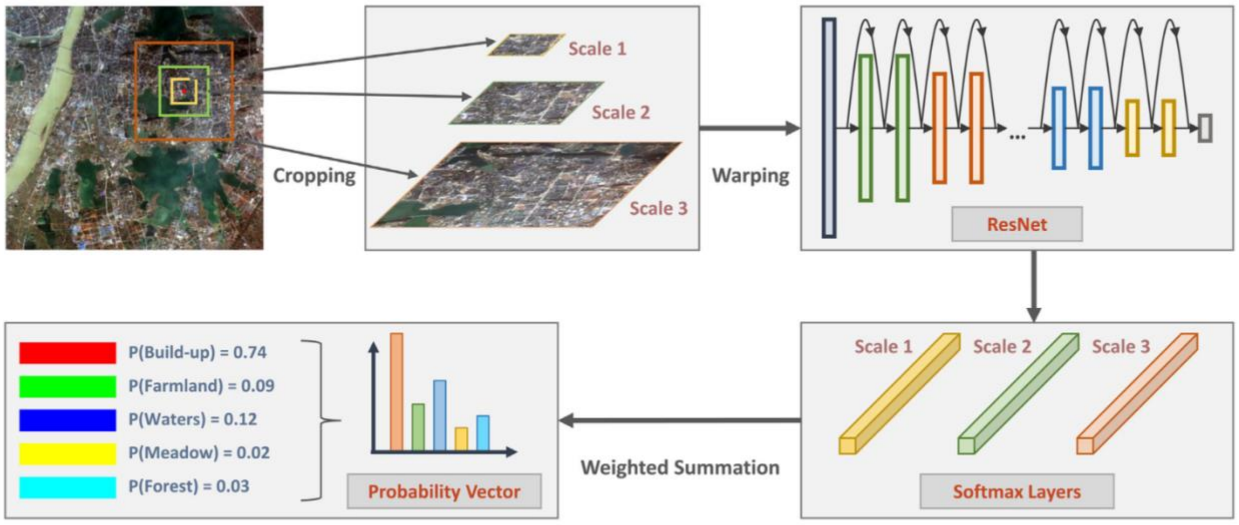


Figure 23 Multi-scale contextual information aggregation[12]

ImageNet [81] is used to initialize the parameters of ResNet-50 and a Gaussian distribution initializes the softmax layer. Training hyper-parameters were set to be: patch size, epoch number, momentum value, and learning rate were 32, 15, 0.9, and 0.1 respectively. The learning rate is divided by 10 and used as a new given value to update parameters when the error rate stops decreasing.

Multiple object-based classification methods were compared with the algorithm introduced by the authors [12]. To turn the image into a single, homogenous item, a specific search method is used. In particular, four distinct features are utilized, such as “spectral feature”, “gray-level co-occurrence matrix (GLCM)”[82], “differential morphological profiles (DMP)”[83], and “local binary patterns (LBP)”[84]. In addition, a multi-feature fusion strategy is used to assemble the aforementioned features via normalization and vector concatenation. “Maximum likelihood classification (MLC)”, “random forest (RF)”, “support vector machine (SVM)”, and “multi-layer perceptron (MLP)” are utilized as classifiers.

The settings of the comparison algorithms are set to their optimal values. 15,000 multi-scale patches were randomly selected from GID's training set to train comparison classifiers. The target data is classified directly after training. "Kappa coefficient (Kappa)", "overall accuracy (OA)", and "user's accuracy" [85] are used as assessment measures to evaluate the algorithm introduced by the authors of [12] by analyzing experimental results. On the validation set of GID and multisource data, the method was put to the test. The classification accuracy in the test images is evaluated on all labeled pixels (except the background). The number of pixels from class (a) that are predicted to belong to class (b) is expressed as Pab, the total number of pixels that belong to class (a) is expressed as ta =ΣbPab and the total amount of pixels member in a class (b) expressed as tb =ΣaPab

Kappa is a metric that determines how well the prediction and the ground truth agree [12]. It is expressed in

$$Kappa = \frac{P_o - P_c}{1 - P_c} \quad (47)$$

Where

$$P_o = \frac{\sum_a P_{aa}}{\sum_a t_a} \quad (48)$$

$$P_c = \frac{\sum_k (\sum_b P_{kb} \cdot \sum_a P_{ak})}{\sum_a t_a \cdot \sum_a t_a} \quad (49)$$

And $k \in [1, K]$, and K represents the of categories amount.

Overall accuracy (OA) is the proportion of properly categorized pixels in the entire image divided by the total number of pixels [12]. It is expressed in

$$OA = = \frac{\sum_a P_{aa}}{\sum_a t_a} \quad (50)$$

The fraction of properly identified pixels in all pixels projected to class b is the user's accuracy of class b [12]. It is expressed in

$$User's\ accuracy = \frac{P_{bb}}{t_b} \quad (51)$$

The values of Kappa, OA, and client exactness extend from 0 to 1, with better-displaying esteem and superior classification execution.

PT-GID and OA had the highest Kappa, with 0.924 and 96.28 percent in 5 classes and 0.605 and 70.04 percent in 15 classes. RF + Fusion produces the best results, with Kappa and OA of 0.641 and 78.45 % on five courses and 0.237 and 33.70 % on fifteen classes, respectively. This shows that standard classifiers and features are incapable of generalizing data shifts.

The efficiency of the technique was validated for multisource data by employing two deep models to classify each target image. ResNet-50 is the first model, that was pre-trained on the source domain. The ResNet-50 fine-tuned using FT-U_{tg} is the second model. The accuracy of FT-U_{tg} is higher than that of other approaches, according to the results.

This experiment demonstrated that, if the target domain and source domain have the same spectral response, the learned information from the source domain samples can help with target domain perception. In contrast, if the target and source domains are obtained from various imagers, the source domain's supervision information is unreliable for the target domain.

The JL-1(2) and its 15-category ground truth, as well as model fine-tuning findings, demonstrate that the algorithm sample selection technique can obtain dependable samples from the target domain.

The cropped image from ZY-3(1) with a size of 1000x1250 pixels and its ground truth approves the rigidity and applicability of the methodology introduced by the authors of [12] for a variety of HRRS images.

Because of the influence of patch size, segmentation technique, and transfer learning scheme thresholds, herein after some analysis of these factors impacts classification results.

As a result, the system relies on image patches made from non-overlapping grid partitions., and the same label is dedicated to all pixels in a patch. The smallest patch of size 56x56 yields the best results. When the patch size is too large, the object features in patches are lost. The approach of multi-scale information fusion has the greatest kappa of 0.924% and an OA of 96.28% in comparison with the best results obtained by single-scale methodologies. These results demonstrate that ground objects in HRRS photos appear to have a wide range of meaningful data at various sizes. Combining picture data from different sizes also helps characterize the spatial distributions of ground items[12].

Five different preliminary segmentation sizes were tested to check the effect of the segmentation scale. The mean OA values per segmentation scale are illustrated in Figure 24.

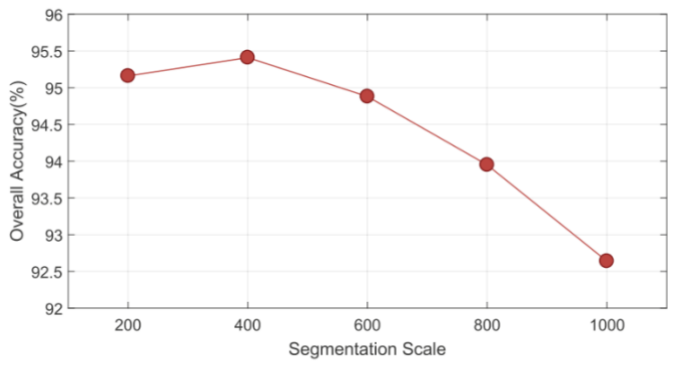


Figure 24: sensitivity analysis for segmentation scale[12]

It is obvious that the best results occur with segmentation scale 400. A comparison of selective search and multisource segmentation is demonstrated in Table 7. The selective search and OA values are slightly higher than for multi-resolution segmentation[12].

For a number of years, it has been challenging to detect objects in satellite imagery. The development of successful machine learning techniques and improvements in hardware systems have enabled more accuracy in the detection of various artifacts from very high-resolution satellite images. Over the past few decades, satellite imagery has been utilized effectively for geological mapping, space planning, and weather forecasting. Low-resolution satellite photos are sufficient for these types of applications[13]. In [13], the authors aimed to utilize high-resolution images, with the prediction being performed by processing the images and creating correspondingly accurate data for the same. RRSC-C ISRO, Nagpur is providing high-resolution satellite images. Using image processing methods the authors of [13], developed a system to classify satellite images and extract information. Satellite images have been divided into usable and unused regions, and each of the classes has been further divided into four subclasses.

Table 7 Comparison of selective search and multi-resolution segmentation.

segmentation	Kappa	OA	User's Accuracy (%)				
method		(%)	built-up	farmland	Forest	meadow	water
Selective Search Method	0.915	95.41	80.02	89.47	76.83	78.87	89.12
Multi-resolution Segmentation	0.907	94.62	82.35	87.50	81.89	83.94	88.01

The proposed algorithm appears to empower classification performance

By dividing the images into several bands and utilizing them to recognize the objects to accomplish their objectives, the authors of [13] current method of classifying objects in an image. Python scripts may be used to convert the bands, and the modified picture can then be sent to the trained machine to analyze the satellite-received image and finish the task given to us based on image processing with the aid of Liss-IV.

As seen in Figure 25., a CNN is made up of several processing layers. Convolution filter families that recognize picture characteristics make up each layer. The CNN eventually creates a set of predicted probabilities, one for each class, by combining the detector outputs in fully linked "dense" layers after the series. As it trains, the network itself learns which features to identify and how to do so.

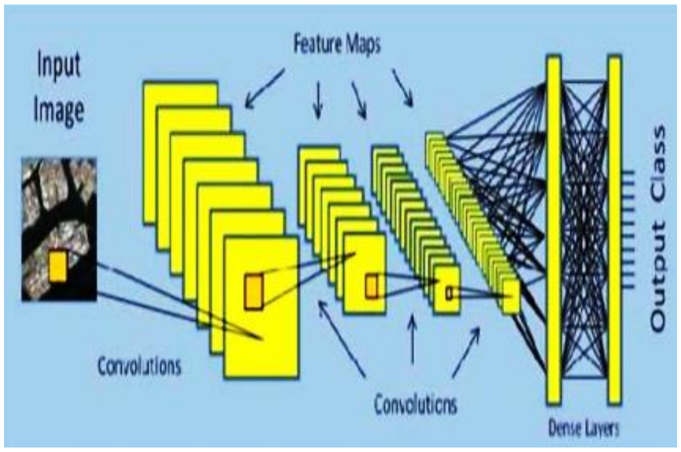


Figure 25 The methodology for classifying satellite images

By using a Keras-based deep U-Net solution for segmenting satellite images. 8 and commercial-grade satellite images from the Space Net collection make up the dataset. The information in these files pertains to five different categories of items, including Tracks, Slow H2O, and Fast H2O. The achieved accuracy of between 90% and 95%.

Packages and Libraries Included, 1) Python's Pandas package is used to manipulate data collections. 2) A tool for visualizing data, Matplotlib is a low-level graph charting framework written in Python. 3) The Python Imaging Library gives your Python interpreter the ability to process images. 4) EfficientNet has been re-implemented in PyTorch. As a result, it is simple to load weights from a TensorFlow checkpoint since it is compatible with the original TensorFlow implementation.

The currently developed system can further be trained to give more efficient results for different band images. The second thing which can be done is to make the website in which direction the image will be imported and the rest of the process is executed automatically. Moreover, accuracy may be further enhanced by improving the data utilized for classification.

4- Discussion and Comparison

In this section, we compare the literature reviews introduced in section (3). With this comparison, we want to highlight the key concepts, methodologies, algorithms, and results of each study. This is done to conclude the effects of these axes on the quality of service (QoS) of the discussed systems.

The results of the given literature are summarized in Table 8

Table 8 Results of the literature

literature	Results
[2]	• As the size of the systolic array is reduced, the DNN accuracy for the same defect rate improves. • SalvageDNN is a project that aims to fix irreversible flaws in DNN accelerators. • The procedure depends on saliency-aware mapping of DNN to a systolic array-based hardware accelerator that does not require retraining and has no influence on data flow at the hardware level.
[3]	• The proposed architecture gives better results in discriminative power in comparison with the augmented CNN architecture.
[4]	• The procedure did not need any optimization and reached comparable prediction accuracy to “PCANet”.
[5]	• A multitask deep learning algorithm for hyperspectral image classification was presented to address the problem of overfitting. • The proposed technique successfully improved the deep CNNs’ performance. • Time was saved in computation since a single multitask network was able to categorize two or more hyperspectral datasets in a single training session

[6]	• Deep learning as well as big data technology are the satellite data processing future.
[7]	• Full stage data-augmented framework could enhance the DCNN • Improved speculation ability on unseen test samples. • Test methods showed efficiency by comparing them with the latest algorithms. • Data augmentation throughout the training and testing stages could guarantee network convergence. • The data augmentation framework could enhance the globalization capability of deep learning models. • It might have no bearing on the model's convergence.
[8]	• Enhance the computing efficiency of IoT edge devices with limited resources. • The DNNs on GPPs were accelerated. • SPARCE enables GPPs to leverage sparsity efficiently. • SPARECE involved two keys micro-architectures improvements. The first key was SpRF, and the second key was the SASA table • could be used by reasonably altering the code age cycle to permit effective identification of the guidance command sequence and following programs of the SASA table. • Evaluation of SPARCE was done over 6 image-recognition DNNs for training and inference. • SPARCE is a potential strategy for accelerating DNNs on GPPs by using all types of static and dynamic sparsity.
[9]	• CNN is promising for shaping practical identification in geotechnics and many other related fields. • It is helpful for computing molecule shape features and classifying them depending on their shape by the available particle classification techniques. • The technique was designed to get 2D molecules and evaluate their shape features. • The algorithm could be used as an elective strategy for in situ shape evaluation as obtaining 3D molecule • enhancements still required to solve: 1- technique is not an end-to-end technique, so post-processing techniques are required. 2- If the input images are significantly different from the training images, the technique does not work well. • The speculation ability ought to be improved by utilizing more progressed neural networks that might be developed later on and by using more varied training images.
[10]	• An autonomous learning algorithm automatically generates a DCNN architecture for a specific image classification problem using a genetic algorithm and data. • The experimental results on the “MNIST”, “Fashion MNIST”, “EMNIST Digit”, “EMNIST Letter”, “CIFAR10”, and “CIFAR100” show that the proposed algorithm can generate DCNN architectures that are comparable to or even outperform state-of-the-art DCNN models.
[11]	• For the landscape scenes to be classified, scale plays a major role in the classification of the remote sensing images. • The accuracy was significantly increased for both LU and LC by predicating object-based CNN and MLP, respectively, and modeling unequivocally the connection between the anticipated LU and LC factors as a joint circulation. • addressing the conspicuous various-leveled connection between LU and LC in both the scale and the ontological sense. • The results showed excellent classification accuracy. • computational productivity in examination with the benchmark techniques, including the as of late proposed joint profound learning (JDL) technique. • The technique is easy to actualize and has extraordinary speculation capacity and functional utility with the default boundary settings. • The SS-JDL subsequently has the potential to alter image categorization in remote sensing and AI.
[12]	• An algorithm could be used to classify multi-source HRRS images. This algorithm has the aforementioned characteristics. 1) Training samples are selected automatically from the target domain depending on the information gathered from the deep model. 2) Multi-scale contextual information is used by the algorithm for classification. 3) patch-wise classification and hierarchical segmentation were merged by the algorithm. 4) The classification map obtains accurate category and boundary information at the same time. 5) the classification noise is reduced. • The algorithm appears to empower classification performance.
[13]	• For RRSC-C, ISRO, and Nagpur, machine learning-based feature extraction and object recognition utilizing high-resolution satellite pictures is being developed. • includes a classifier that aids in retrieving the data from the satellite sensor's various bands. • a system is used to identify things like landmasses, waterbodies, and forests. • An NVIDIA GPU processor located in the server room is supplied an image by the system. • The segmentation process uses the neural network model to provide predicted pixels for each class.

Table 8 illustrates different deep-learning models used in image classification and shape extraction. All these models are applied to satellite images and data sets. Some models are enhanced from old models to accelerate the models or to give better performance and accuracy.

DNNs, a branch of AI, have been applied in RS applications especially image classifications. This highly usage of DDNs give accuracy better than human, but unfortunately consume energy with more complex methodologies and architectures.

5. Points of future research

From the literature review in section (3) and the discussion in section (4), we can highlight the main points of the works as follows:

“Salvaging deep neural network accelerators with permanent faults through saliency-driven fault-aware mapping” [2]. “DeepSat V2, feature-augmented convolutional neural nets for satellite image classification” [3]. “FrequentNet , A New Deep Learning Baseline for Image Classification” [4]. “Multitask deep learning with spectral knowledge for hyperspectral image classification. ” [5]. “Next-Generation Artificial Intelligence Techniques for Satellite Data Processing” [6]. “A full-stage data augmentation method in deep convolutional neural network for natural image classification” [7]. “SPARCE, Sparsity-aware General-Purpose Core Extensions to Accelerate Deep Neural Networks” [8] “A particle shape extraction and evaluation method using a deep convolutional neural network and digital image processing” [9], “Autonomous deep learning: A genetic DCNN designer for image classification" [10], "Scale Sequence Joint Deep Learning (SS-JDL) for land use and land cover classification" [11], "Land-cover classification with high-resolution remote sensing images using transferable deep models" [12]. and " Satellite Image Classification and Analysis using Machine Learning with ISRO LISS IV" [13]

However, based on the given literature and discussion, we can suggest the following points for future research:

Enhance the generalization ability of particle shape extraction and evaluation methods using a deep convolutional neural network and digital image processing.

Enhance the classification accuracy and improve the land-cover classification with high-resolution remote sensing images using transferable deep models.

Use the full-stage data augmentation method to improve and enhance the models of different DNNs.

Try to find automatic and generic methods to be used for image classification and use multitasking methods to enhance the processing time.

Apply and implement the algorithm on different hardware and compare the performance.

Depends on methodologies that enable efficient processing of DNNs to enhance energy economy and throughput without losing accuracy with affordable hardware.

6. Conclusions:

This paper presented a literature survey for the classification of satellite images using different methods and architectures. These methods were applied to various datasets of different images. From these papers, we made comparisons among the methods used, and finally, we found some points for future research.

Finally, DNNs are still a significant field of study with a wide range of innovative uses and potential.

Conflict of Interest

The authors declare no conflict of interest.

References

[1] J. E. Ball, D. T. Anderson, and C. S. Chan, "Comprehensive survey of deep learning in remote sensing: theories, tools, and challenges for the community," *J. Appl. Remote Sens.*, vol. 11, no. 04, p. 1, 2017.

[2] M. A. Hanif, "Subject Areas : Author for correspondence : SalvageDNN : salvaging deep neural network accelerators with permanent faults through mapping," 2019.

[3] Q. Liu *et al.*, "DeepSat V2: feature augmented convolutional neural nets for satellite image classification," *Remote Sens. Lett.*, vol. 11, no. 2, pp. 156–165, 2020.

[4] Y. Li, Z. Wang, X. Lu, K. Song, and Y. Sun, "FrequentNet : A New Interpretable Deep Learning Baseline for Image Classification," 2020.

[5] S. Liu and Q. Shi, "Multitask Deep Learning With Spectral Knowledge for Hyperspectral Image Classification," *IEEE Geosci. Remote Sens. Lett.*, vol. PP, pp. 1–5, 2020.

[6] N. Sisodiya, N. Dube, and P. Thakkar, "Next-Generation Artificial Intelligence Techniques for Satellite Data Processing," *Remote Sens. Digit. Image Process.*, vol. 24, pp. 235–254, 2020.

[7] Q. Zheng, M. Yang, X. Tian, N. Jiang, and D. Wang, "A full stage data augmentation method in deep convolutional neural network for natural image classification," *Discret. Dyn. Nat. Soc.*, vol. 2020, 2020.

[8] S. Sen, S. Jain, S. Venkataramani, and A. Raghunathan, "S PAR CE : Sparsity aware General-Purpose Core Extensions to Accelerate Deep Neural Networks," *IEEE Trans. Comput.*, vol. PP, no. c, p. 1, 2018.

[9] Z. Liang, Z. Nie, A. An, J. Gong, and X. Wang, "A particle shape extraction and evaluation method using a deep convolutional neural network and digital image processing," *Powder Technol.*, vol. 353, pp. 156–170, 2019.

[10] B. Ma, X. Li, Y. Xia, and Y. Zhang, "Autonomous deep learning: A genetic DCNN designer for image classification," *Neurocomputing*, vol. 379, pp. 152–161, 2020.

[11] C. Zhang, P. A. Harrison, X. Pan, H. Li, and I. Sargent, "Scale Sequence Joint Deep Learning (SS-JDL) for land use and land cover classification," *Remote Sens. Environ.*, vol. 237, no. September 2019, p. 111593, 2020.

[12] X. Y. Tong *et al.*, "Land-cover classification with high-resolution remote sensing images using transferable deep models," *Remote Sens. Environ.*, vol. 237, no. July 2018, p. 111322, 2020.

[13] T. Jain, S. Pandey, S. Mishra, and S. Yadav, "Satellite Image Classification and Analysis using Machine Learning with ISRO LISS IV," no. May, pp. 1479–1487, 2022.

[14] A. Yusuf and S. Alawneh, "A Survey of GPU Implementations for Hyperspectral Image Classification in Remote Sensing," *Can. J. Remote Sens.*, vol. 44, no. 5, pp. 532–550, 2018.

[15] J. Zhang, T. Gu, K. Basu, and S. Garg, "Analyzing and mitigating the impact of permanent faults on a systolic array based neural network accelerator," in *Proceedings of the IEEE VLSI Test Symposium*, 2018, vol. 2018-April, pp. 1–6.

[16] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," 2009.

[17] M. D. Zeiler, "ADADELTA: An Adaptive Learning Rate Method," 2012.

[18] S. Basu, S. Ganguly, S. Mukhopadhyay, R. DiBiano, M. Karki, and R. Nemani, "DeepSat - A learning framework for satellite imagery," *GIS Proc. ACM Int. Symp. Adv. Geogr. Inf. Syst.*, vol. 03-06-Nove, pp. 1–22, 2015.

[19] 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, 2015.

[20] K. S. Choi, J. S. Shin, J. J. Lee, Y. S. Kim, S. B. Kim, and C. W. Kim, "In vitro trans-differentiation of rat mesenchymal cells into insulin-producing cells by rat pancreatic extract," *Biochem. Biophys. Res. Commun.*, vol. 330, no. 4, pp. 1299–1305, 2005.

[21] H. Larochelle, D. Erhan, A. Courville, J. Bergstra, and Y. Bengio, "An empirical evaluation of deep architectures on problems with many factors of variation," *ACM Int. Conf. Proceeding Ser.*, vol. 227, no. 2006, pp. 473–480, 2007.

[22] S. Dev, B. Wen, Y. H. Lee, and S. Winkler, "Machine Learning Techniques and Applications For Ground-based Image Analysis," pp. 1–12, 2016.

[23] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-Up

Robust Features (SURF),” *Comput. Vis. Image Underst.*, vol. 110, no. 3, pp. 346–359, 2008.

[24] K. Radhika and S. Varadarajan, “Joint Deep Learning for land cover and land use classification,” *Cogent Eng.*, vol. 5, no. 1, pp. 1–9, 2018.

[25] G. Bilgin, S. Ertürk, and T. Yildirim, “Unsupervised classification of hyperspectral-image data using fuzzy approaches that spatially exploit membership relations,” *IEEE Geosci. Remote Sens. Lett.*, vol. 5, no. 4, pp. 673–677, 2008.

[26] R. L. Cannon, J. C. Bezdek, J. V. Dave, and M. M. Trivedi, “Segmentation of a Thematic Mapper Image Using the Fuzzy c-Means Clustering Algorithm,” *IEEE Trans. Geosci. Remote Sens.*, vol. GE-24, no. 3, pp. 400–408, 1986.

[27] A. R. S. Marçal and L. Castro, “Hierarchical clustering of multispectral images using combined spectral and spatial criteria,” *IEEE Geosci. Remote Sens. Lett.*, vol. 2, no. 1, pp. 59–63, 2005.

[28] I. Khan, P. M. Roth, A. Bais, and H. Bischof, “Semi-supervised image classification with huberized Laplacian Support Vector Machines,” *ICET 2013 - 2013 IEEE 9th Int. Conf. Emerg. Technol.*, vol. 5, no. 3, pp. 336–340, 2013.

[29] W. Liu, S. Member, S. Li, Y. Zhou, and S. Member, “Hypergraph - Laplacian Regularization for Remote Sensing Image Recognition,” pp. 1–9.

[30] Z. Huigang *et al.*, “via Graph Laplacian Energy,” *IEEE Geosci. Remote Sens. Lett.*, vol. 10, no. 2, pp. 396–400, 2013.

[31] G. Camps-valls, S. Member, and T. V Bandos, “Semi-supervised Graph-based Hyperspectral Image Classification,” vol. XX, pp. 1–29, 2007.

[32] L. Bruzzone, M. Chi, and M. Marconcini, “A novel transductive SVM for semisupervised classification of remote-sensing images,” *IEEE Trans. Geosci. Remote Sens.*, vol. 44, no. 11, pp. 3363–3372, 2006.

[33] M. Chi and L. Bruzzone, “Semisupervised classification of hyperspectral images by SVMs optimized in the primal,” *IEEE Trans. Geosci. Remote Sens.*, vol. 45, no. 6, pp. 1870–1880, 2007.

[34] L. Gómez-Chova, L. Bruzzone, G. Camps-Valls, and J. Calpe-Maravilla, “Semi-supervised remote sensing image classification based on clustering and the mean map kernel,” *Int. Geosci. Remote Sens. Symp.*, vol. 4, no. 1, pp. 2–5, 2008.

[35] S. Saha, B. Banerjee, and S. N. Merchant, “Unsupervised domain adaptation without source domain training samples - A maximum margin clustering based approach,” *ACM Int. Conf. Proceeding Ser.*, 2016.

[36] M. Abadi and L. De Alfaro, *Lecture Notes in Computer Science: Preface*, vol. 3653. 2005.

[37] D. Quan, S. Wang, M. Ning, T. Xiong, and L. Jiao, “Using deep neural networks for synthetic aperture radar image registration,” in *2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, 2016, pp. 2799–2802.

[38] “Deep learning for satellite imagery via image segmentation - deepsense.ai.” [Online]. Available: https://deepsense.ai/deep-learning-for-satellite-imagery-via-image-segmentation/. [Accessed: 05-Oct-2020].

[39] “CIFAR-10 and CIFAR-100 datasets.” [Online]. Available: https://www.cs.toronto.edu/~kriz/cifar.html. [Accessed: 05-Oct-2020].

[40] X. Zhang, Y. Zou, and W. Shi, “Dilated convolution neural network with LeakyReLU for environmental sound classification,” *Int. Conf. Digit. Signal Process. DSP*, vol. 2017-Augus, 2017.

[41] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” *32nd Int. Conf. Mach. Learn. ICML 2015*, vol. 1, pp. 448–456, 2015.

[42] M. MORTENSEN, “Learning Deep Features for Discriminative Localization,” *Acad. Manag. Proc.*, vol. 2004, no. 1, pp. M1–M6, 2004.

[43] Y. Jia *et al.*, “Caffe,” pp. 675–678, 2014.

[44] V. Sangeetha and K. J. R. Prasad, “Deep Residual Learning for Image Recognition,” *Indian J. Chem. - Sect. B Org. Med. Chem.*, vol. 45, no. 8, pp. 1951–1954, 2006.

[45] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc.*, pp. 1–14, 2015.

[46] G. Zeng, Y. He, Z. Yu, X. Yang, R. Yang, and L. Zhang, “Preparation of novel high copper ions removal membranes by embedding organosilane-functionalized multi-walled carbon nanotube,” *J. Chem. Technol. Biotechnol.*, vol. 91, no. 8, pp. 2322–2330, 2016.

[47] N. Binkert *et al.*, “The gem5 simulator,” *ACM SIGARCH Comput. Archit. News*, vol. 39, no. 2, pp. 1–7, 2011.

[48] “OpenBLAS : An optimized BLAS library.” [Online]. Available: https://www.openblas.net/. [Accessed: 06-Oct-2020].

[49] T. F. Gonzalez, “ImageNet Classification with Deep Convolutional Neural Networks,” *Handb. Approx. Algorithms Metaheuristics*, pp. 1–1432, 2007.

[50] O. Ronneberger, P. Fischer, and T. Brox, “U-Net: Convolutional Networks for Biomedical Image Segmentation,” pp. 1–8.

[51] A. Fabija, “Segmentation of corneal endothelium images using a U-Net-based convolutional neural network,” 2018.

[52] R. Deng, C. Shen, S. Liu, H. Wang, and X. Liu, “Learning to predict crisp boundaries,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 11210 LNCS, pp. 570–586, 2018.

[53] D. P. Kingma and J. L. Ba, “Adam: A method for stochastic optimization,” *3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc.*, pp. 1–15, 2015.

[54] P. re Soille, *Morphological Image Analysis: Principles and Applications*. SpringerVerlag, Berlin, Heidelberg, 1999.

[55] C. Liu, B. Shi, J. Zhou, and C. Tang, “Quantification and characterization of microporosity by image processing, geometric measurement and statistical methods: Application on SEM images of clay materials,” *Appl. Clay Sci.*, vol. 54, no. 1, pp. 97–106, 2011.

[56] J. Zheng and R. D. Hryciw, “Traditional soil particle sphericity, roundness and surface roughness by computational geometry,” *Geotechnique*, vol. 65, no. 6, pp. 494–506, 2015.

[57] P. Vangla, N. Roy, and M. L. Gali, “Image based shape characterization of granular materials and its effect on kinematics of particle motion,” *Granul. Matter*, vol. 20, no. 1, pp. 1–19, 2018.

[58] N. Zhihong, L. Zhengyu, W. Xiang, and G. Jian, “Evaluation of granular particle roundness using digital image processing and computational geometry,” *Constr. Build. Mater.*, vol. 172, pp. 319–329, 2018.

[59] J. R. and C. de Boor, “A Practical Guide to Splines.,” *Math. Comput.*, vol. 34, no. 149, p. 325, 1980.

[60] “GitHub - keras-team/keras: Deep Learning for humans.” [Online]. Available: https://github.com/keras-team/keras. [Accessed: 03-Apr-2021].

[61] C. Zhang, P. A. Harrison, X. Pan, H. Li, I. Sargent, and P. M. Atkinson, “Joint Deep Learning for land cover and land use classification,” *Remote Sens. Environ.*, vol. 237, pp. 1–50, 2020.

[62] C. Zhang *et al.*, “An object-based convolutional neural network (OCNN) for urban land use classification,” *Remote Sens. Environ.*, vol. 216, no. April, pp. 57–70, 2018.

[63] C. Zhang *et al.*, “Joint Deep Learning for land cover and land use classification,” *Remote Sens. Environ.*, vol. 221, no. November 2018, pp. 173–187, 2019.

[64] J. R. Jensen and K. Lulla, “Introductory digital image processing: A remote sensing perspective,” *Geocarto Int.*, vol. 2, no. 1, p. 65, 1987.

[65] P. Gong, D. J. Marceau, and P. J. Howarth, “A Comparison of Spatial Feature Extraction Algorithms for Land-Use Classification with SPOT HRV Data,” 1992.

[66] P. Casals-Carrasco, S. Kubo, and B. Babu Madhavan, “Application of spectral mixture analysis for terrain evaluation studies,” *Int. J. Remote Sens.*, vol. 21, no. 16, pp. 3039–3055, 2000.

[67] S. Giada, T. De Groeve, D. Ehrlich, and P. Soille, “Information extraction from very high resolution satellite imagery over Lukole refugee camp, Tanzania,” *Int. J. Remote Sens.*, vol. 24, no. 22, pp. 4251–4266, Nov. 2003.

[68] Y. Tarabalka, J. Chanussot, and J. A. Benediktsson, “Segmentation and classification of hyperspectral images using minimum spanning forest grown from automatically selected markers,” *IEEE Trans. Syst. Man, Cybern. Part B Cybern.*, vol. 40, no. 5, pp. 1267–1279, Oct. 2010.

[69] Y. Zhong, J. Zhao, and L. Zhang, “A hybrid object-oriented conditional random field classification framework for high spatial resolution remote sensing imagery,” *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 11, pp. 7023–7037, 2014.

[70] L. Ma, M. Li, X. Ma, L. Cheng, P. Du, and Y. Liu, “A review of supervised object-based land-cover image classification,” *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 130. Elsevier B.V., pp. 277–293, 01-Aug-2017.

[71] W. Zhao and S. Du, “Learning multiscale and deep representations for classifying remotely sensed imagery,” *ISPRS J. Photogramm. Remote Sens.*, vol. 113, pp. 155–165, 2016.

[72] Y. Zhong, S. Wu, and B. Zhao, “Scene semantic understanding based on the spatial context relations of multiple objects,” *Remote Sens.*, vol. 9, no. 10, 2017.

[73] F. Hu, G. S. Xia, J. Hu, Y. Zhong, and K. Xu, “Fast binary coding for the scene classification of high-resolution remote sensing imagery,” *Remote Sens.*, vol. 8, no. 7, 2016.

[74] H. Yu, W. Yang, G. S. Xia, and G. Liu, “A color-texture-structure descriptor for high-resolution satellite image classification,” *Remote Sens.*, vol. 8, no. 3, pp. 1–24, 2016.

[75] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2016-Decem, pp. 770–778, 2016.

[76] K. Wu and K.-H. Yap, “Fuzzy SVM for Content-Based Image Retrieval: a pseudo-label support vector machine framework,” *IEEE Comput. Intell. Mag.*, no. May, pp. 10–16, 2006.

[77] X. Ya, L. Xuejun, L. Carin, and B. Krishnapuram, “Multi-task learning for classification with Dirichlet process priors,” *J. Mach. Learn. Res.*, vol. 8, pp. 35–63, 2007.

[78] W. Ge and Y. Yu, “Borrowing treasures from the wealthy: Deep transfer learning through selective joint fine-tuning,” *Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017*, vol. 2017-Janua, pp. 10–19, 2017.

[79] J. R. R. Uijlings, K. E. A. Van De Sande, T. Gevers, and A. W. M. Smeulders, “Selective search for object recognition,” *Int. J. Comput. Vis.*, vol. 104, no. 2, pp. 154–171, 2013.

[80] P. F. Felzenszwalb and D. P. Huttenlocher, “Efficient graph-based image segmentation,” *Int. J. Comput. Vis.*, vol. 59, no. 2, pp. 167–181, 2004.

[81] J. Deng, W. Dong, R. Socher, L.-J. Li, Kai Li, and Li Fei-Fei, “ImageNet: A large-scale hierarchical image database,” no. May 2014, pp. 248–255, 2010.

[82] R. Haralick, … K. S.-I. T. on, and undefined 1973, “Textural features for image classification,” *ieeexplore.ieee.org*.

[83] J. A. Palmason, J. A. Benediktsson, and J. R. Sveinsson, “Classification of hyperspectral ROSIS data from urban areas Based on Extended Morphological Profiles,” *RAST 2005 - Proc. 2nd Int. Conf. Recent Adv. Sp. Technol.*, vol. 2005, no. 3, pp. 63–69, 2005.

[84] T. Ojala, M. Pietikäinen, and T. Mäenpää, “Gray scale and rotation invariant texture classification with local binary patterns,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 1842, pp. 404–420, 2000.

[85] P. Olofsson, G. M. Foody, M. Herold, S. V. Stehman, C. E. Woodcock, and M. A. Wulder, “Good practices for estimating area and assessing accuracy of land change,” *Remote Sens. Environ.*, vol. 148, pp. 42–57, 2014.