An Effective Hyperparameter Tuned Deep Learning based Residual Network for Remote Sensing Image Scene Classification

N.Bharanikumar[1*], Dr. P. Dhanalakshmi[2]

[1*] Department of Computer and Information Science, Annamalai University, Annamalai Nagar 608002

[2] Department of Computer Science & Engg., Annamalai University, Annamalai Nagar,

Abstract

Due to the advancements in the remote sensing technologies in present times, massive quantity of high resolution aerial images are being generated and beneficial for different applications like ecological observation, traffic control, urban mapping, etc. Since the image classification process is a primary criteria for interlinking aerial images and their applications, several research works have been carried out to categorize the images into proper class labels. In this view, this paper presents a novel deep convolution neural network based Adaptive Delta (Adadelta) Optimized Residual Network with Logistic Regression (LR) Classifier, named ADRN-LR model for multi-label aerial image classification captured by UAVs. The presented ADRN-LR model involves image preprocessing to boost the quality of the image collected by UAVs. In addition, it make use of ResNet-152 model as a feature extractor, which is further tuned by the Adadelta technique. Besides, LR model is employed to classify the feature vectors generated by the ADRN model to determine the proper class labels of the remote sensing images. A detailed experimental validation process is carried out on the benchmark dataset and the results are investigated under several aspects. The obtained results ensured the effective classification outcome of the ADRN-LR model over the compared methods with the maximum accuracy of 0.9568, precision of 0.9589, recall of 0.9252, and F-score of 0.9390.

Keywords: Remote sensing, Image classification, Deep learning, Parameter tuning, Adadelta

1. Introduction

In last decades, Unmanned Aerial Vehicles (UAVs) or drones have accomplished the maximum concentration from many developers, especially from remote sensing communities for massive number of real-time domains like traffic surveillance [1], searching and rescue operation, optimal farming, and satellite imagery computation. The advanced developments like combination of camera sensors intend to create novel UAV domains like prediction, observation, and investigation of passive as well as active risk factors and intrusions in various scenarios such as forest fire, flooding disasters, road incidents, earthquake regions, and so on [2]. Moreover, since the UAV is small size equipment, rapid development is possible where it is in-the-loop of mission to make effective decisions for managing the accessible resources and to enhance the risk evaluation mechanism and elimination of threats. Hence, an exclusive set of issues has to be reported as UAV is operated on disaster-based regions in which remote communication for cloud is not feasible to be developed. Consequently, a maximum level of autonomy has become essential to assure the performance supremacy and real-time investigation. Fig. 1 illustrates the architecture of UAV.

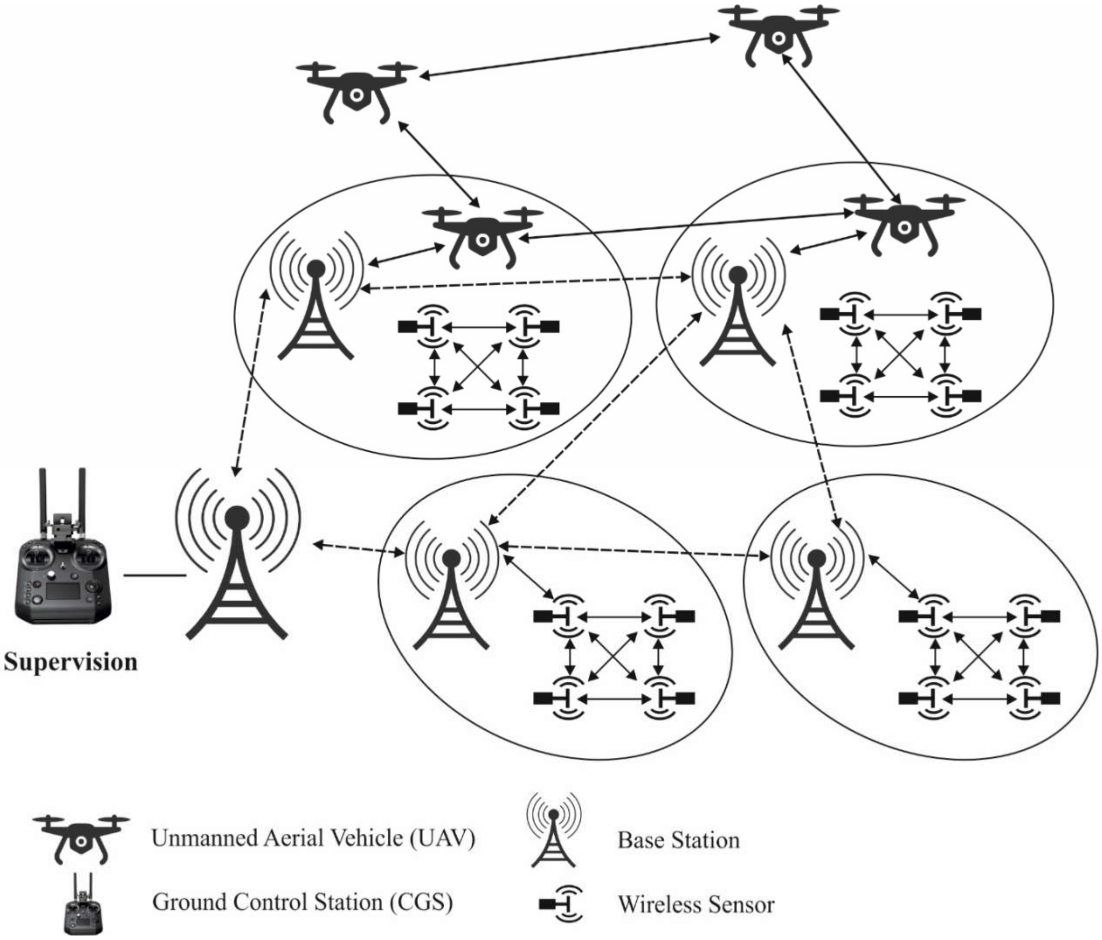


Fig. 1. Architecture of UAVs

At this point, independent UAV depends upon on-board sensors as well as microprocessors for computing the given performance with no requirement of central ground station. Moreover, the independent process of UAV by integrating path planning models with automatic on-board visual processing activates them to enclose massive regions within short span of time interval [3]. Therefore, on-board processing is emerged with prior set of constraints due to the inadequate processing resources as well as low-power issues which are considered to be low-payload abilities of UAVs. Followed by, processing efficiency of computer vision method is an important approach in activating independent UAV for hazard prediction in real-time scenarios. Ultimately, Deep Learning (DL) methods like Convolutional Neural Networks (CNNs) were utilized extremely as an eminent scheme for massive number of computer vision domains like image or video analysis, prediction, and classification where standard outcomes for various domains. Therefore, DL method is highly beneficial in immediate response and rescue management modules for retrieving private data regularly which activates better deployment as well as response at the time of time-based scenarios and withstands in-the-loop decision-making operations.

Advanced works have showcased that process of how DL methods overcome classical Machine Learning (ML) model using hand-based characteristics under the application of Transfer Learning (TL) in which a pre-trained CNN was employed as a feature extractor and numerous layers are included on top for computing classification process [4]. Although CNN is more effective in classification operations, the inference robustness on implanted low-power channels like on-board UAVs is concealed by high-processing cost which is presented in [5], particularly while assuming the requirement for implementing various vision operations on similar environment. Thus, in various applications like local embedded computation near a sensor has been decided in a cloud because of the privacy factors where no connectivity is possible. Furthermore, tiny networks are used for providing better accuracy and working function for niche applications in which dense information is not accessible and processing limitations are induced. In addition, apart from the processing efficiency, it is more robust in training iterations which can be upgraded over-the-air. Therefore, by applying a tiny CNN were amicable for closer sensor modeling to compute aerial scene categorization on board, a UAV is highly significant and sensible for various models.

This study develops an effectivedeep convolution neural network-based Adaptive Delta (Adadelta)Optimized Residual Network with Logistic Regression (LR) Classifier, named ADRN-LR model formulti-label aerial image classification captured by UAVs. The presented ADRN-LR model initially performs the image preprocessing to boost the quality of the image collected by UAVs. Also, ResNet-152 model is utilized to extract the useful feature vectors. In addition, the parameters of the ResNet-152 model undergo tuning using the Adadelta technique. Furthermore, LR model is applied to categorize the feature vectors created by the ADRN model to regulate the proper class labels of the remote sensing images. An extensive simulation analysis takes place to ensure the goodness of the ADRN-LR model.

2. Literature Survey

Diverse approaches were presented in last decades for predicting the disasters present in an image like image-processing related thresholds for computing pixel-level categorization [6], Gaussian Mixture Models (GMM) that requires empirical tuning, and Support Vector Machines (SVM) that are assumed to be slow in real-world domains [7]. In developers have presented a cloud-related DL model for fire prediction with UAVs [8]. This prediction operation is operated under the application of custom CNN (VGG16) which undergoes training for discriminating fire and non-fire photographs of 128 × 128 resolution. Also, it is processed by sending the video clips from UAV to workstation along with NVIDIA Titan Xp GPU, in which this model has been implemented. Obviously, minimum connectivity has massive complexities in using this model. Hence, the projected technique has attained a better accuracy. In [9], a system presented for predicting objects of interest in avalanche disaster under the application of pre-trained inception network for feature extraction and linear SVM for classification. Moreover, it presents an image segmentation scheme as pre-processing approach which depends upon the fact in object of interest is varied color when compared with the background for isolating the image into regions with the help of a sliding window. Moreover, post-processing has been employed to enhance the solution of classification method on Hidden Markov Models (HMM) [10] intends to predict the fire existence using DL method. Particularly, 2 pre-trained CNNs have been applied and related such as VGG16 and Resnet50 which are assumed to be fundamental structures used for training fire prediction systems. It is applied by including fully connected (FC) layers once the feature extraction is completed and gain maximum classification accuracy. Followed by, in [11] a method for wildfire prediction is performed under the application of UAV scheme. The entire technology is composed of a CNN named as Fire_Net with 15 layers as same as VGG16 network of convolutional, max-pooling, and FC layers for examining fire in 128 × 128 resolution images. It can be attained by a region proposal technique which extracts image areas from large resolution images where it is classified using Neural Network (NN).

3. The Proposed ADRN-LR Model

Fig. 2 shows the working procedure involved in the presented ADRN-LR model. The figure states that the remote sensing images need to be preprocessed in the initial stage to raise its quality. Next, the features in the preprocessed images are extracted by the ResNet-152 model in which the parameters in it are tuned by the Adadelta technique. At last, LR model is employed to determine the different class labels of the input test images.

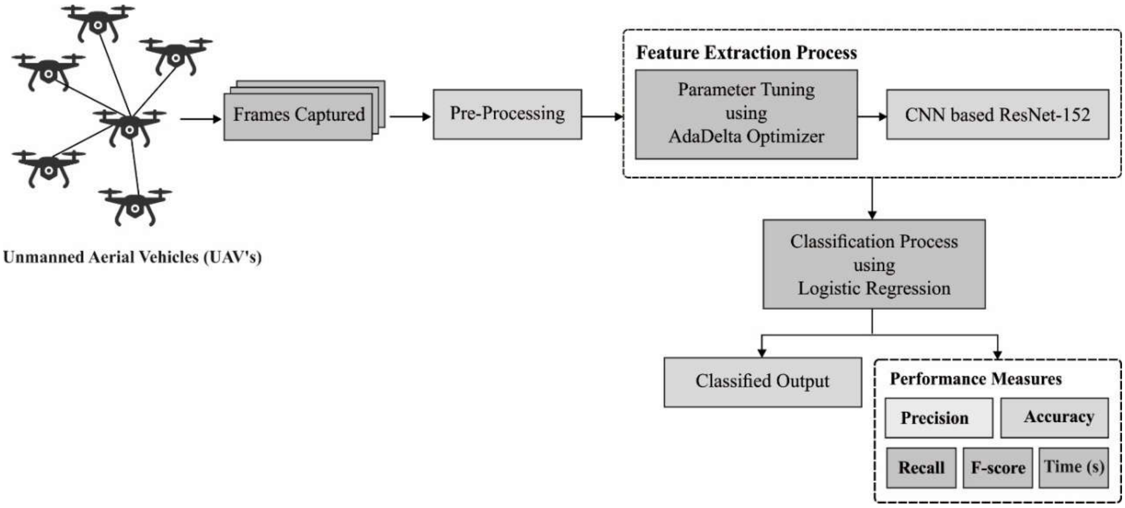

Unmanned Aerial Vehicles (UAV's)
Frames Captured
Pre-Processing
Feature Extraction Process
Parameter Tuning using AdaDelta Optimizer
CNN based ResNet-152
Classification Process using Logistic Regression
Classified Output
Performance Measures
Precision
Accuracy
Recall
F-score
Time (s)

Fig. 2. Architecture of Proposed ADRN-LR Method

3.1. Preprocessing

The remote sensing images collected by the UAVs might not be clear in some scenarios. When the object color is not static, image segmentation approaches enable the segmentation of frames into target areas and irrelevant objects. It enables the frame region to be processed and also discard the frames with no probable objects. Using a sliding window, in the HSVcolor space, a frame undergo scanning in the preprocessingstage, and all windows are verified by the saturation window component thresholding $th_{sat}(V)$. The thresholding is carried out using Eq. (1)

$$th_{sat}(V) = 1.0 - \frac{0.8V}{255} \tag{1}$$

where V is the intensity component rate. Once the saturation component rate exceeds to $th_{sat}(V)$, the pixels are computed based on the object.

3.2. Feature Extraction

Once the preprocessing of the remote sensing images seems completed, the next process is to extract a useful set of feature vectors using ADRN technique.

3.2.1. Overview of CNN

Basically, CNNis a DL method which has been extensively applied in image-based operations such as image analysis, image classification, image captioning and so on. Such networks are defined as a integration of convolution, pooling, and FC layers. The 3 blocks have been applied for developing CNN method by changing block count, inclusion, or deletion of a block. The architecture is composed of different hyper-parameters with novel approaches such as Drop out, Batch normalization,and so on.

Convolution layer

This layer varies with NN where a pixel is linked to upcoming layer alongwith a weight and bias, however, the complete image is divided into tiny regions ($n \times n$ matrix) as well as weights and bias are employed in them. Such weights and bias are assumed as filters or kernels which is convoluted with tiny region of input image that provides feature maps. Such filters are elegant 'features' that is explored in an input image of a convolution layer. The count of attributes essential for convolution task which is minimum as same filter traversed through the entire image for single feature. The count of filters, size of a local region, stride, and padding are assumed as hyper parameters of conv. layer. According to the size and genre of input image, where hyper parameters are tuned for accomplishing optimal outcomes.

Pooling layer

To diminish the spatial dimension of an image and attribute count which limits the processing, the pooling layer has been applied. This layer computes a permanent function across an input, therefore no parameters have been established. There are two pooling layers namely, average, stochastic and max pooling. Initially, max pooling is a prominently applied pooling method where a$n \times n$ window is a slideover a stride value s and a position with maximum value in $n \times n$ region, consequently limiting the input size. This layer offers translational invariance where a small difference in a position is applicable for image analysis. However, the location details are dropped as input size is mitigated.

Fully connected (FC) layer

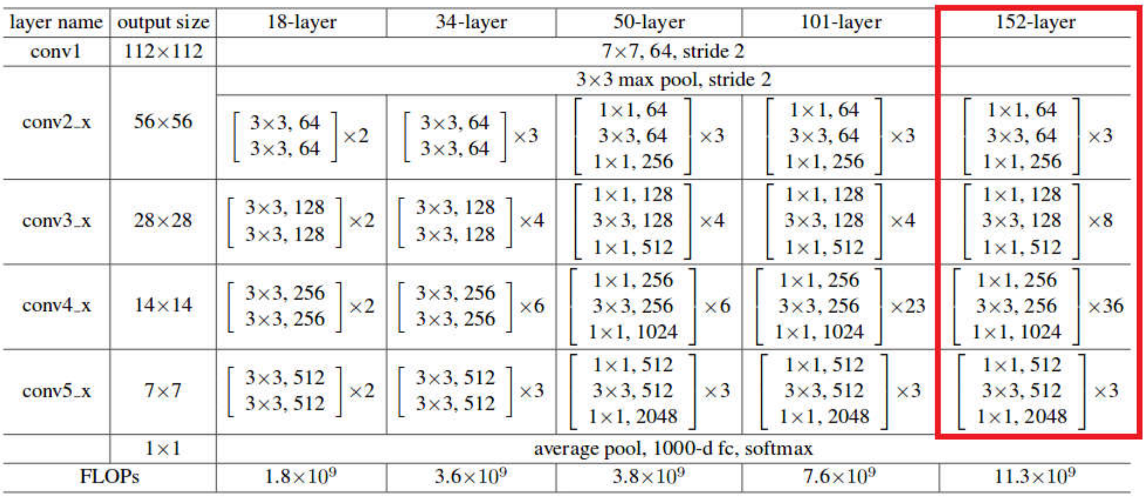
Here, flattened result of consequent pooling layer is induced as input for FC layer. It performs as a classical NN in which a neuron of existing layer is related to a projected layer. Therefore, count of variables is superior to convolution layer.

Activation function

Diverse activation functions were applied over structures of CNN. Non-linear activation functions like Rectified Linear Unit (ReLU), Leaky Rectified Linear Unit (LReLU), Parametric Rectified Linear Unit (PReLU), and Swish have ensured effective outcomes than traditional sigmoid functions. Such non-linear functions were guided in increasing the training. Here, various activation functions and identified ReLU for identifying effectual outcomes.

3.2.2. ResNet-152 model

ResNet methods are relied on deep structures which have implied better convergence behaviors as well as compelling accuracy. ResNet was created by numerous stacked residual elements and layers. Therefore, enormous process could be varied on the basis of various structures. Followed by, residual units are comprised of convolutional, pooling, and FC layers. ResNet is same as VGG net; however, ResNet is extremely deeper when compared with VGG. ResNet-152 was applied for the classification of remote sensing images. Initially, ResNet-152 is defined as a residual DL system which intends to resolve the issues of diminishing gradients which exist in back-propagation (BP) of CNN. It collection of ResNet techniques have different depths of imagery classification. Enhancing the network depth leads to improved system accuracy and over-fitting is referred.



layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix}3\times3, 64\\3\times3, 64\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 64\\3\times3, 64\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 64\\3\times3, 64\\1\times1, 256\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 64\\3\times3, 64\\1\times1, 256\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 64\\3\times3, 64\\1\times1, 256\end{bmatrix}\times3$
conv3_x	28×28	$\begin{bmatrix}3\times3, 128\\3\times3, 128\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 128\\3\times3, 128\end{bmatrix}\times4$	$\begin{bmatrix}1\times1, 128\\3\times3, 128\\1\times1, 512\end{bmatrix}\times4$	$\begin{bmatrix}1\times1, 128\\3\times3, 128\\1\times1, 512\end{bmatrix}\times4$	$\begin{bmatrix}1\times1, 128\\3\times3, 128\\1\times1, 512\end{bmatrix}\times8$
conv4_x	14×14	$\begin{bmatrix}3\times3, 256\\3\times3, 256\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 256\\3\times3, 256\end{bmatrix}\times6$	$\begin{bmatrix}1\times1, 256\\3\times3, 256\\1\times1, 1024\end{bmatrix}\times6$	$\begin{bmatrix}1\times1, 256\\3\times3, 256\\1\times1, 1024\end{bmatrix}\times23$	$\begin{bmatrix}1\times1, 256\\3\times3, 256\\1\times1, 1024\end{bmatrix}\times36$
conv5_x	7×7	$\begin{bmatrix}3\times3, 512\\3\times3, 512\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 512\\3\times3, 512\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 512\\3\times3, 512\\1\times1, 2048\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 512\\3\times3, 512\\1\times1, 2048\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 512\\3\times3, 512\\1\times1, 2048\end{bmatrix}\times3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

Fig. 3. Variants in CNN Based ResNet Model

Therefore, the issues with maximum depth is a signal required for weight modification that is emerged from the end of a system by comparing ground-truth as well as prediction becomes minimum at previous layers. It refers that the previous layers are unlearned or unknown. It is named as “diminishing gradient” issues, as a matrix of 2nd order derivatives in which nonlinear optimization is managed for tuning the weights are considered as zero. Additionally, training deep networks can be performed by developing a network using modules named as residual models. It is referred as degradation constraints. The different versions of ResNet are shown in Fig. 3 and the layered ResNet-152 structure is presented in Fig. 4.

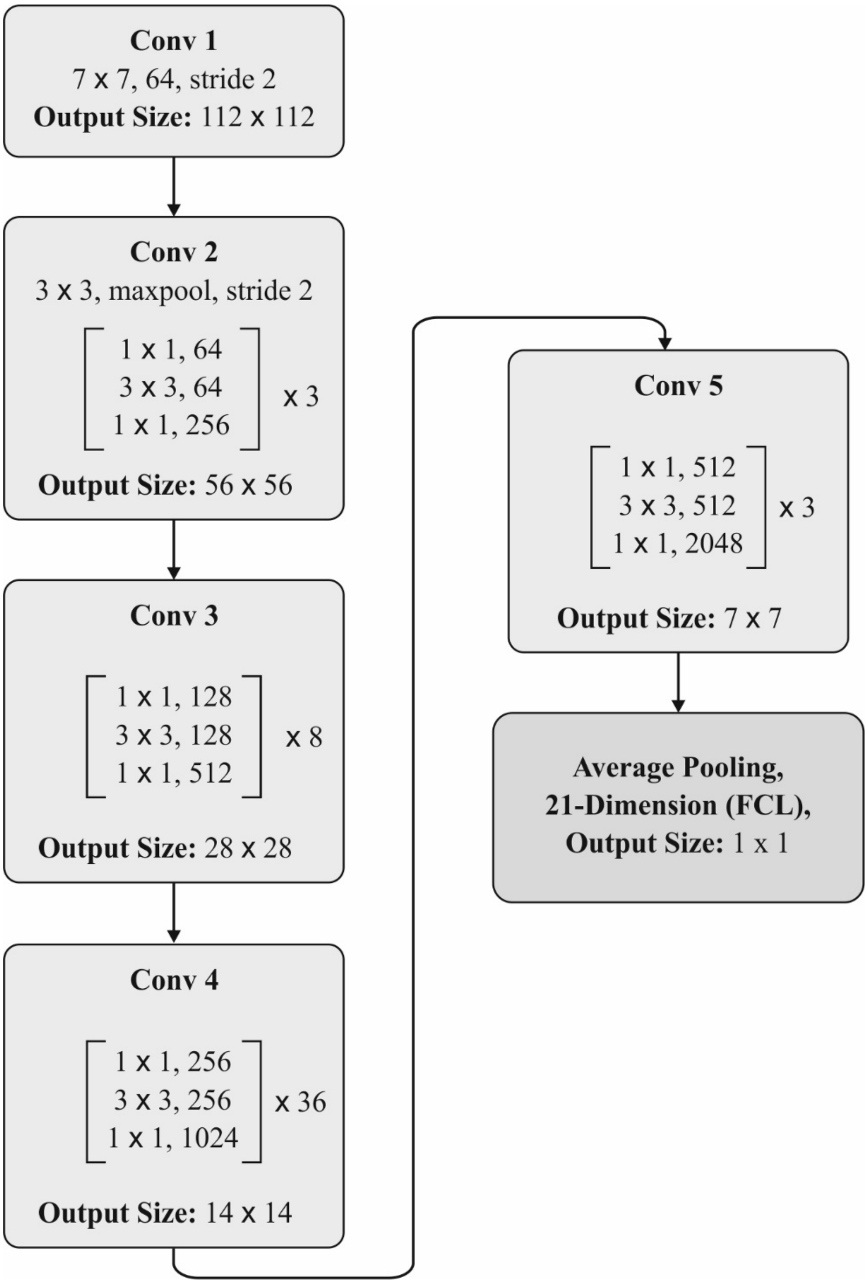


Fig. 4. Layers of ResNet-152

Traditional blocks are considered as main element of ResNet-152 scheme. Such networks utilize several filters (3 × 3-pixel size filter) for image classification. Such filters are shifted by strides over the actual image. Hence, the measures in filters are increased over the values of images. Finally, simulation outcome of these filters undergoes pooling and it is effectively down-sampled at the time of maintaining significant features.

3.2.3. Adadelta based Parameter Tuning

In order to tune the hyperparamters of the ResNet-152 model, Adadelta technique is employed and thereby maximum classification outcome can be attained. AdaDelta is an extended version of Adagrad that removes the inferior learning rate problem. The AdaDelta scheme is highly effective in extending the weights once the 1st order derivative is reached, which is time-consuming and cost-effective when compared with earlier models. Here, it is proficient for raucous gradient details, data modules, hyper-parameters, and model design solutions. It enhances the direction of steepest descent stated by using a false gradient.

$$\nabla x_t = -\eta g_t \tag{2}$$

where $g_t$implies a gradient at ith iteration $\frac{\delta f(x_t}{\delta f(x_t}$) and ηdenotes a learning rate that controls how wider the stage is moved towards a negative gradient. Selecting a learning rate and implementation of better learning rate estimated per value by applying the 1st order, it employs a modest amount of estimation per iteration in gradient descent (GD), which is defined as a downside for AdaDelta. It is identified by the hyper parameters which are not considered with optimal degree for changing the simulation results. The 2 major merits and demerits are:

- incessant rotting of learning rates in training time
- demands for automated selection of comprehensive learning rates.

AdaDelta with robust learning is applicable in clustered allotment of conditions where no demands for physical setting of a learning rate.

3.3. Classification

In this section, LRis employed as a classifier to determine the appropriate class labels of the remote sensing images using the extracted features. LR implies a multivariate analysis model that investigates the correlation among autonomous and dependent parameters for variable $\boldsymbol{y}$ and sequence of essential characteristics $\boldsymbol{x}$, as given in the following:

$$Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 +, \dots, \beta_n x_n (3)$$

$$p = \frac{1}{1 + e^{-p}} \tag{4}$$

where $p$means the possibility of existing parameter Y, and the measure of p is among 0 and 1. Here, the LR process the possible function of single instance is defined in:

$$h_\theta(x) = g(\theta^T x) = \frac{1}{1 + e^{-\theta^T x}} \tag{5}$$

$$\theta^T x = \theta_0 + \theta_1 x_1 + \cdots + \theta_n . x_n \tag{6}$$

where $\theta_0$refers the constant of a coefficient vector to be evaluated, $\theta_i$means a coefficient of argument x_i in function 1. In case of $h_\theta(x)$, the measure of a function implies the possibility of consuming 1 from parameter y. Hence, for input argument vectors and probability, $x$intends to generate 1 and 0 are:

$$p(y = 1|x; \theta) = h_\theta(x) \tag{7}$$

$$p(y = 0|x; \theta) = 1 - h_\theta(x) \tag{8}$$

Probability density likelihood expressions of $m$instances are represented by:

$$L(\theta) = \prod_{i=1}^{m} p\,(y_i|x_i;\theta) = \prod_{i=1}^{m} \left((h_\theta(x_i)^{y_i}1 - h_\theta(x_i)^{1-y})\right) \qquad (9)$$

The Log likelihood expressions are equated as:

$$\ln\big(L(\theta)\big) = \sum_{i=1}^{m}(y_i \ln\,(h_\theta x_i)) + (1-y_i)\ln(1-h_\theta x_i) \qquad (10)$$

$$h_{\hat{\theta}} = \frac{1}{1+e^{-\hat{\theta}^{Tx}}} \qquad (11)$$

By evaluating the y and $x$measures of m test instances, the determined measures of co-efficient vectors which enhance the function are recursively measured as θ. Here, function (9)could be applied in function (8). The possibility of y of 1 is calculated where x is an autonomous parameter vector in a sample m. As $h_{\hat{\theta}}$ is considered to apply logistic distribution, maximum-likelihood models could be employed for estimating the coefficients θ.

4. Experimental Validation

The presented ADRN-LR model has been experimented with using a benchmark remote sensing image dataset [12, 13], which comprises 21 classes of land images with 100 images under every class label. The size of every image is 256*256 pixels. The extraction of images takes place manually from the USGS National Map Urban Area Imagery collection with several urban regions over the country. The resolution of the pixels in the public domain images is 1 foot.The different sample images with the class label are shown in Fig. 5.

Table 1 and Fig. 6 exhibit the comparative results analysis of the ADRN-LR model with existing DL models on the applied dataset [14]. The resultant values implied the ineffective classification of the PlacesNet model with the least accuracy of 0.9144. Also, the VGG-VD19 model has offered certainly higher accuracy of 0.9315, which exceeds the performance of the PlacesNet. At the same time, six DL models such as VGG-VD16, VGG-F, AlexNet, CaffeNet, VGG-M, and VGG-S have showcased moderately closer performance with the accuracy of 0.9315, 0.9407, 0.9435, 0.9437, 0.9448, and 0.946 respectively. But the presented ADRN-LR model has reached to a higher classification accuracy of 0.9568 and outperformed all the existing methods.

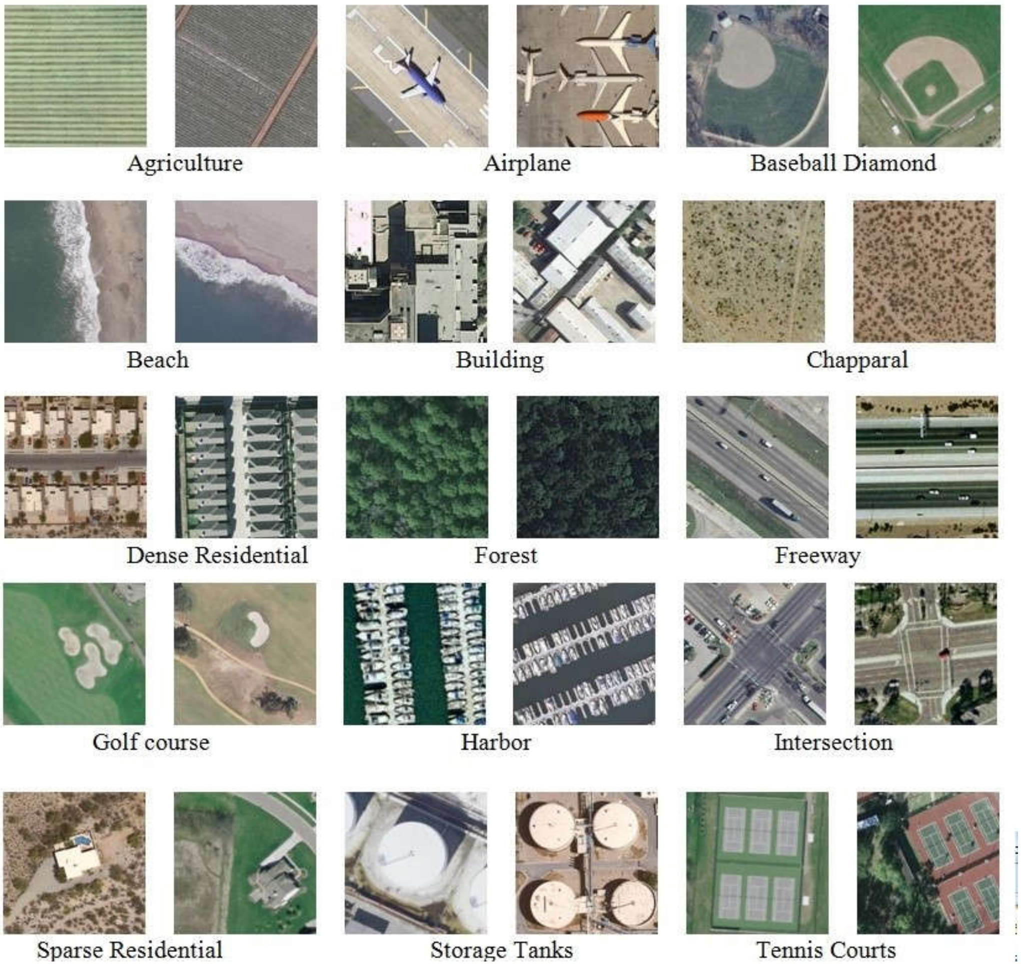


Fig. 5. Sample Images from UCM Dataset

Table 1 Accuracy Analysis of ADRN-LR with Various CNN Models

Methods	**Accuracy**
AlexNet	0.9437
CaffeNet	0.9443
VGG-F	0.9435
VGG-M	0.9448
VGG-S	0.9460
VGG-VD16	0.9407
VGG-VD19	0.9315
PlacesNet	0.9144
Proposed ADRN-LR	**0.9568**

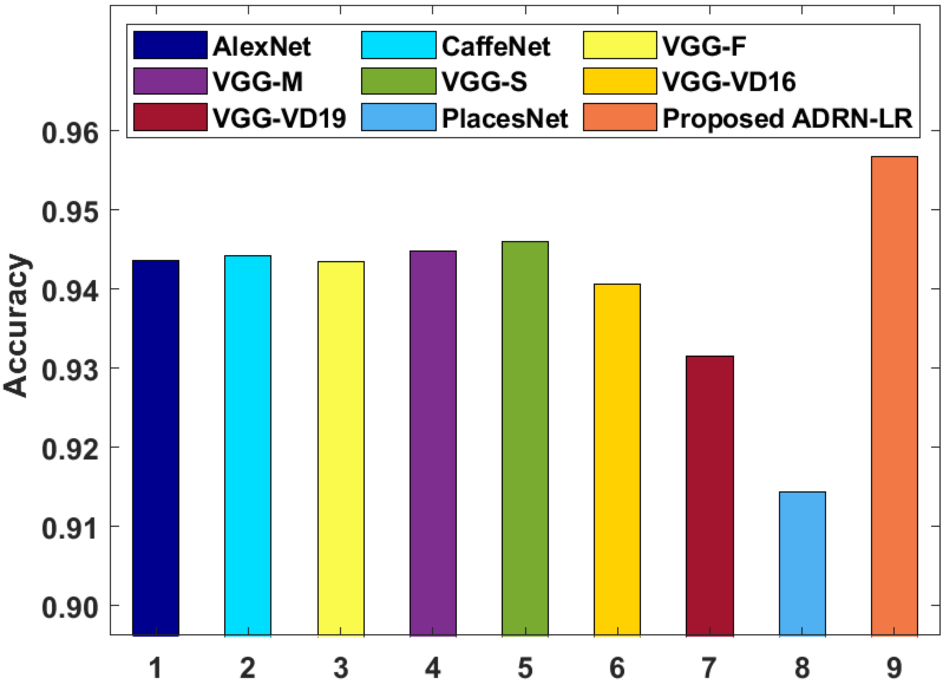


Fig. 6. Accuracy analysis of ADRN-LR model with different CNN models

To further validate the classification results of the proposed ADRN-LR model, an extensive comparative study is made with the existing CNN based models interms of precision, recall, and F-score, as shown in Table 2 and Fig. 7 [15].

Table 2 Performance Analysis of ADRN-LR with Various Models in terms of Precision, Recall, F-score

Model	**F-score**	**Precision**	**Recall**
VGGNet	0.7854	0.7906	0.8230
VGG-RBFNN	0.7880	0.7818	0.8391
CA-VGG-LSTM	0.7957	0.8064	0.8247
CA-VGG-BiLSTM	0.7978	0.7933	0.8399
GoogLeNet	0.8068	0.8051	0.8427
GoogLeNet-RBFNN	0.8154	0.7995	0.8675
CA-GoogLeNet-LSTM	0.8178	0.7852	0.8860
CA-GoogLeNet-BiLSTM	0.8182	0.7991	0.8706
ResNet-50	0.7968	0.8086	0.8195
ResNet-RBFNN	0.8058	0.7992	0.8459
CA-ResNet-LSTM	0.8136	0.7990	0.8614

CA-ResNet-BiLSTM	0.8147	0.7794	0.8902
Proposed ADRN-LR	**0.9390**	**0.9589**	**0.9252**

The table values denoted that the CA-ResNet-BiLSTM model has appeared as the poor performance by attaining a lower precision of 0.7794, recall of 0.8902, and F-score of 0.8147. Besides, the VGG-RBFNN model has tried to showcase slightly better outcomes by offering a precision of 0.7818, recall of 0.8391, and F-score of 0.788. In addition, the CA-GoogLeNet-LSTM model has exhibited even better outcomes over the earlier models by achieving a precision of 0.7852, recall of 0.886, and F-score of 0.8178. Similarly, the VGGNet model has demonstrated moderate performance with the precision of 0.7906, recall of 0.823, and F-score of 0.7854. Likewise, the CA-VGG-BiLSTM model has accomplished somewhat satisfactory results with the precision of 0.7933, recall of 0.8399, and F-score of 0.7978 whereas even higher precision of 0.799, recall of 0.8614, and F-score of 0.7978 has been attained by the CA-ResNet-LSTM model. Moreover, the CA-GoogLeNet-BiLSTM model has shown manageable classification outcome with the precision of 0.7991, recall of 0.8459, and F-score of 0.8182. Simultaneously, the ResNet-RBFNN model has tried to demonstrate reasonable performance and ended up with the precision of 0.7992, recall of 0.8459, and F-score of 0.8058. Concurrently, the GoogLeNet-RBFNN model has resulted to a somewhat satisfactory performance with the precision of 0.7995, recall of 0.8675, and F-score of 0.8154. Additionally, the GoogLeNet model has obtained even better precision of 0.8051, recall of 0.8427, and F-score of 0.8068 whereas the near acceptable classification outcome is depicted by the CA-VGG-LSTM model with the precision of 0.8064, recall of 0.8247, and F-score of 0.7957. Although the near optimal classification outcome is demonstrated by the ResNet model with the precision of 0.8086, recall of 0.8195, and F-score of 0.7968, the presented ADRN-LR model has shown superior performance with the precision of 0.9589, recall of 0.9252, and F-score of 0.939.

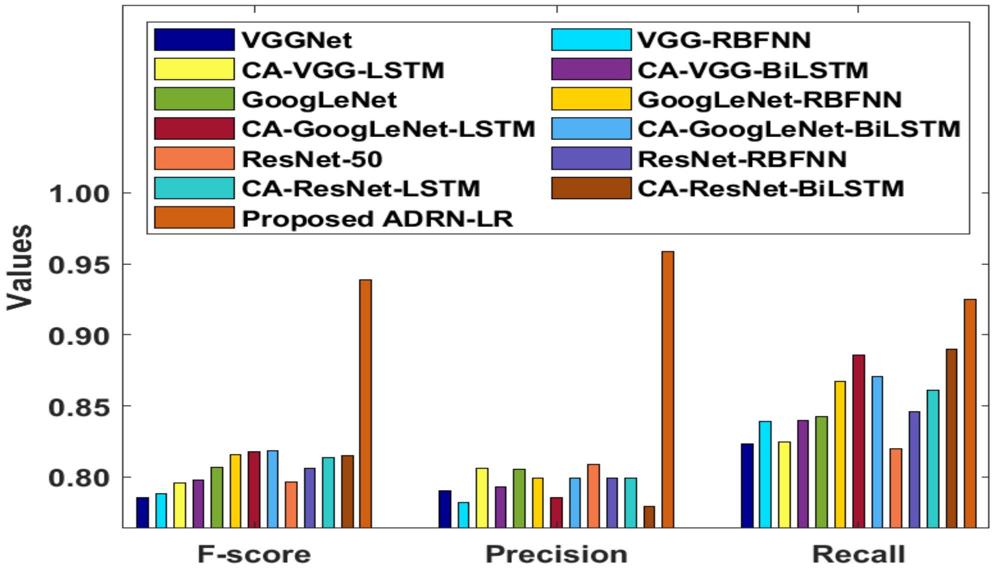


Fig. 7. Result analysis of ADRN-LR model

Table 3 and Fig. 8 exhibit the comparative results analysis of the ADRN-LR model with traditional models [16]. The resultant values implied the ineffective classification of the SCK model with the least accuracy of 0.7252. Also, the SPM model has offered certainly higher accuracy of 0.7400, which exceeds the performance of the SCK.

Table 3 Accuracy Analysis of ADRN-LR with Traditional models

Methods	**Accuracy**
SPM	0.7400
SCK	0.7252
SPCK++	0.7738
SC+Pooling	0.8167
SG+UFL	0.8664
CCM-BOVW	0.8664
PSR	0.8910
UFL-SC	0.9026
MSIFT	0.9097
COPD	0.9133
Dirichlet	0.9280
VLAT	0.9430
OverFeat	0.9091
Proposed ADRN-LR	**0.9568**

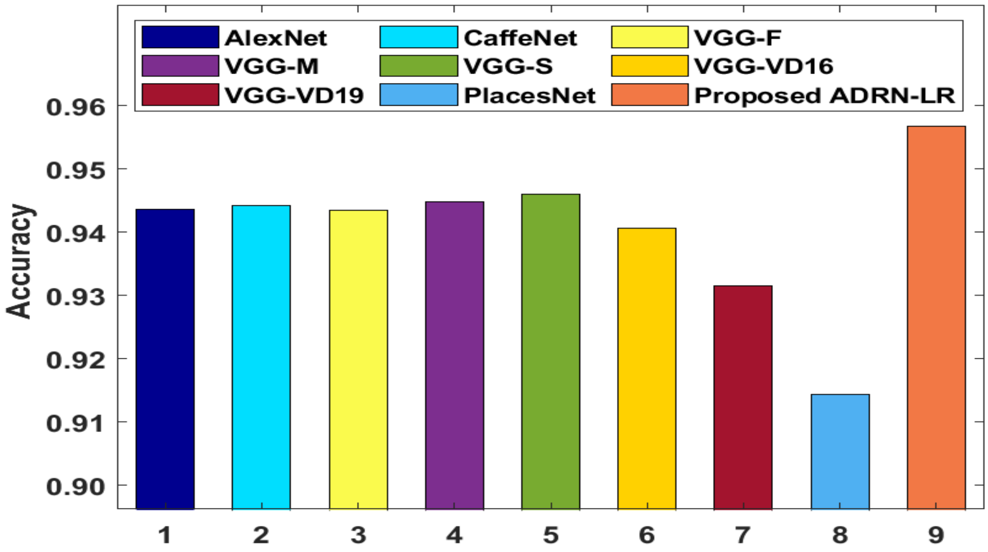


Fig. 8. Accuracy analysis of ADRN-LR model with traditional models

At the same time, the SPCK++ and SC+Pooling have showcased moderately closer performance with the accuracy of 0.7738 and 0.8167 respectively. Similarly, the SG+UFL and CCM-BOVW models have demonstrated the higher result with identical accuracy of 0.8664. In line with, the PSR, UFL-SC, MSIFT, and COPD models even superior results with accuracy of 0.8910, 0.9026, 0.9097, and 0.9133. Besides, the Dirichlet, VLAT, and OverFeat models have exhibited that slightly higher result with accuracy of 0.9280, 0.9430, and 0.9091 respectively. But the presented ADRN-LR model has reached to a higher classification accuracy of 0.9568.

Table 4 and Fig. 9 illustrates a running time analysis of the presented ADRN-LR model takes place with existing CNN models on the applied test dataset[17]. The obtained values inferred that the VGG-VD19 model needs an enormous amount of running time with 496s whereas the VGG-VG16 model has exhibited slightly better performance by requiring a running time of 423s. Followed by, it is observed that the VGG-M and VGG-S models have needed a moderate running time of 124s and 135s respectively. On continuing with, the CaffeNet and AlexNet models have also showcased closer running time of 85s and 86s. Though the VGG-F and PlacesNet models have obtained a reasonable and identical running time of 82s, the presented ADRN-LR model has needed only a minimum running time of 78s.

Table 4 Running Time (s) Analysis of presented method ADRN-LRwith various CNN Models

Methods	**Running Time (s)**
AlexNet	086
CaffeNet	085
VGG-F	082
VGG-M	124
VGG-S	135
VGG-VD16	423
VGG-VD19	496
PlacesNet	082
Proposed ADRN-LR	**078**

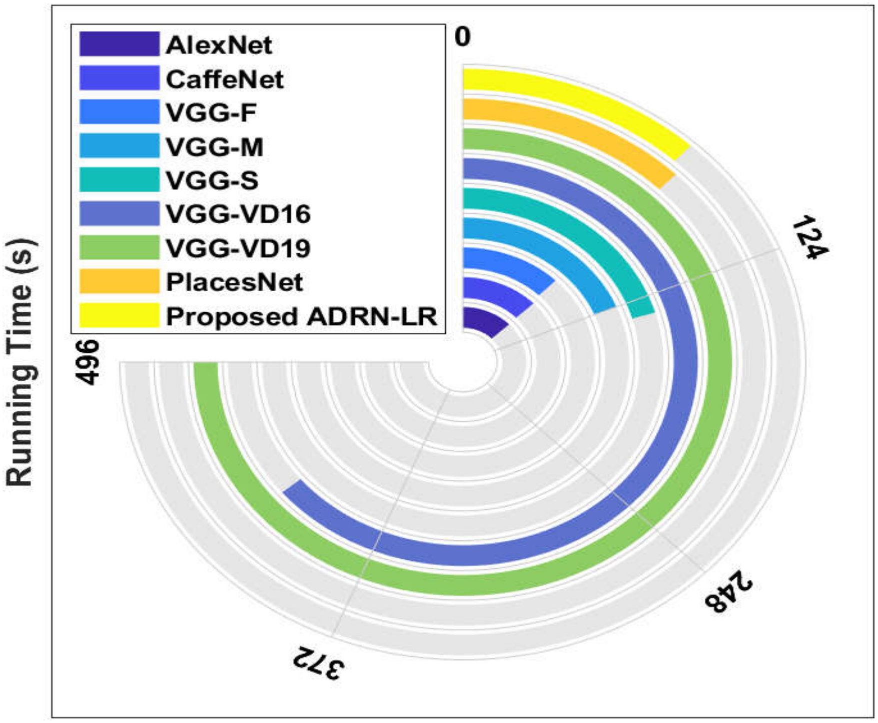


Fig. 9. Running time analysis of ADRN-LR model

From the above-mentioned tables and figures, it is evident that the ADRN-LR model can be considered as an effective remote sensing image scene classification tool. The presented ADRN-LR model has reached to optimal classification performance with minimum running time under all the applied test images.

5. Conclusion

This study has introduced an effective ADRN-LR model for multi-label aerial image classification captured by UAVs.Primarily, the remote sensing images need to be preprocessed in the initial stage to raise its quality. Next, the features in the preprocessed images are extracted by the ResNet-152 model in which the parameters in it are tuned by the Adadelta technique. LR is employed as a classifier to determine the appropriate class labels of the remote sensing images using the extracted features.A detailed experimental validation process is carried out on the benchmark dataset and the results are investigated under several aspects. The obtained results ensured the effective classification outcome of the ADRN-LR model over the compared methods with the maximum accuracy of 0.9568, precision of 0.9589, recall of 0.9252, and F-score of 0.9390. The results assured that the ADRN-LR model can be employed as an appropriate tool for real-time processing.

References

[1] C. Kyrkou, S. Timotheou, P. Kolios, T. Theocharides, and C. G. Panayiotou, “Optimized vision-directed deployment of UAVs for rapid traffic monitoring,” in Proc. IEEE Int. Conf. Consum. Electron., Jan. 2018, pp. 1–6.

[2] A. Al-Kaff, D. Martn, F. Garca, A. de la Escalera, and J. M. Armingol, “Survey of computer vision algorithms and applications for unmanned aerial vehicles,” Expert Syst. Appl., vol. 92, pp. 447–463, 2018.

[3] P. Petrides, C. Kyrkou, P. Kolios, T. Theocharides, and C. Panayiotou, “Towards a holistic performance evaluation framework for drone-based object detection,” in Proc. Int. Conf. Unmanned Aircr. Syst., Jun. 2017, pp. 1785–1793.

[4] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, “CNN features off-the-shelf: An astounding baseline for recognition,” in Proc. IEEE Conf. Comput. Vision Pattern Recognit. Workshops, 2014, pp. 512–519.

[5] Y. Wang, Z. Quan, J. Li, Y. Han, H. Li, and X. Li, “A retrospective evaluation of energy-efficient object detection solutions on embedded devices,” in Proc. Des., Autom. Test Eur. Conf. Exhib., Mar. 2018, pp. 709–714.

[6] C. Yuan, Z. Liu, and Y. Zhang, “UAV-based forest fire detection and tracking using image processing techniques,” inProc. Int. Conf. Unmanned Aircr. Syst., Jun. 2015, pp. 639–643.

[7] B. C. Ko, K.-H. Cheong, and J.-Y. Nam, “Fire detection based on vision sensor and support vector machines,” Fire Saf. J., vol. 44, no. 3, pp. 322– 329, 2009.

[8] S. Kim,W. Lee, Y.-s. Park, H.W. Lee, and Y. T. Lee, “Forest fire monitoring system based on aerial image,” in Proc. 3rd Int. Conf. Inf. Commun. Technol. Disaster Manag., Dec. 2016, pp. 1–6.

[9] M. B. Bejiga, A. Zeggada, A. Nouffidj, and F. Melgani, “A convolutional neural network approach for assisting avalanche search and rescue operations with UAV imagery,” Remote Sens., vol. 9, no. 2, 2017, Art. no. 100.

[10] J. Sharma, O.-C. Granmo, M. Goodwin, and J. T. Fidje, “Deep convolutional neural networks for fire detection in images,” in Eng. Appl. Neural Netw., G. Boracchi, L. Iliadis, C. Jayne, and A. Likas, Eds. Cham, Switzerland: Springer, 2017, pp. 183–193.

[11] Y. Zhao, J. Ma, X. Li, and J. Zhang, “Saliency detection and deep learningbased wildfire identification in UAV imagery,” Sensors, vol. 18, no. 3, 2018, Art. no. 712.

[12] Yi Yang and Shawn Newsam, "Bag-Of-Visual-Words and Spatial Extensions for Land-Use Classification," ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (ACM GIS), 2010.

[13] http://weegee.vision.ucmerced.edu/datasets/landuse.html

[14] Rajagopal, A., Joshi, G.P., Ramachandran, A., Subhalakshmi, R.T., Khari, M., Jha, S., Shankar, K. and You, J., 2020. A Deep Learning Model Based on Multi-Objective Particle Swarm Optimization for Scene Classification in Unmanned Aerial Vehicles. IEEE Access, 8, pp.135383-135393.

[15] Hua, Y., Mou, L. and Zhu, X.X., 2019. Recurrently exploring class-wise attention in a hybrid convolutional and bidirectional LSTM network for multi-label aerial image classification. ISPRS journal of photogrammetry and remote sensing, 149, pp.188-199.

[16] Qi, K., Guan, Q., Yang, C., Peng, F., Shen, S. and Wu, H., 2018. Concentric circle pooling in deep convolutional networks for remote sensing scene classification. Remote Sensing, 10(6), p.934.

[17] Rajagopal, Aghila, A. Ramachandran, K. Shankar, Manju Khari, Sudan Jha, Yongju Lee, and Gyanendra Prasad Joshi. "Fine-tuned residual network-based features with latent variable support vector machine-based optimal scene classification model for unmanned aerial vehicles." IEEE Access 8 (2020): 118396-118404.