A Multilevel Feature Selection Framework for Land-Use Classification

By

Hameed Ali

CIIT/FA19-REE-025/Wah

MS Thesis

In

Electrical Engineering

COMSATS University Islamabad
Wah Cantt- Pakistan

Spring, 2022

Abstract

A Multilevel Feature Selection Framework for Land-Use Classification

With technological advancements in remote sensing (RS), amounts of data gathered through satellites and other remote sensing sources have increased. The data obtained through RS has numerous applications including agriculture, change detection, monitoring of land-use, etc. RS is also an important and widely used technique to observe the earth's natural resources and environmental changes.

Extraction of correct and meaningful information from this big data using conventional means is becoming difficult. Therefore, the use of machine learning (ML) techniques, which make understanding and analysis of the data easier for end-user, is on the rise. However, with these advantages, some challenges are also faced.

The classification frameworks are highly dependent on the available features - extracted either using the conventional or deep-learning methods. Therefore, the extracted features should be providing the most discriminant information in order to achieve a higher classification rate. Since remotely sensed images have assorted appearances, generation of discriminative feature representation becomes difficult. Therefore, Satellite images classification accuracy can be boosted by effective feature representation. In this research, a system is proposed for the classification of satellite images. In the proposed system a discriminative feature representation is generated by combining distinct information of deep features. Pre-trained convolutional neural networks (CNNs) DenseNet-201, Inception-ResNet-v2, and NASNet-Mobile have been used for extracting the features through transfer learning. For this goal, a single strategy in which fully connected layers are efficiently used for the representation of different levels of image features. Secondly, different features selected from different architectures are fused to obtain a new feature set. Thirdly, entropy and ReliefF algorithms are used for the selection of relevant and discriminant features. Various classifiers are used for classification. The effectiveness of the proposed approach is validated by two benchmark datasets; WHU-RS19 and UC Merced.

Table of Contents

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

Acc.	Accuracy
AE	Auto-Encoder
Auth.	Author
CNN	Convolutional Neural Network
DBN	Deep Belief Network
DL	Deep Learning
En. Sub. Disc.	Ensemble Subspace Discriminant
FC	Fully Connected Layer
FPR	False Positive Rate
FPR	False Positive Rate
FS	Feature Selection
HIS	Hyperspectral Imaging
Linear Disc.	Linear Discriminant
LULC	Land-use Land-cover
Med. NN	Medium Neural Network
MG SVM	Medium Gaussian SVM
ML	Machine Learning
MSI	Multispectral Imaging
NLP	Natural Language Processing
NPV	Negative Predictive Value
PPV	Positive Predictive Values
PPV	Positive Predictive Values
Q SVM	Quadratic SVM
RNN	Recurrent Neural Network
RS	Remote sensing
SD	Subspace Discriminant

TL	Transfer Learning
TN	True Negatives
TNR	True Negative Rate
TP	True Positives
Tr. Time	Training Time
Wide NN	Wide Neural Network

Chapter 1
Introduction

1.1 Remote Sensing (RS)

Remote sensing (RS) can be defined as a system that acquires information from a distance without any physical contact using sensors. Normally this information is gathered from the earth's surface through satellites and aircraft. The earth is observed through sensors mounted on satellites and aircraft which detect and record reflected or emitted energy. Valuable information about earth systems is provided through remote sensing. Satellites provide detailed global observations due to their wide field of view and arrangement of these satellites in orbits in a way that the whole earth's surface can be observed in any given time period.

In RS the entire earth or its regions are studied. Due to the advances in RS technologies, the volume of data in the earth observations is on the rise [1], resultantly, offering new opportunities in many applications [2].

The data obtained through RS is used in numerous application domains such as agriculture, climate change, disaster recovery, and urban development [1, 3]. Other application areas include change detection, object recognition, land-use land-cover classification (LULC), etc. RS is also an important and widely used technique to observe the earth's natural resources and environmental changes.

1.1.1 Working of RS

Typically, RS working consists of the following 4 stages:

1. Upwelling energy is recorded by the sensor on the remote sensing platform.
2. The raw data is downlinked to a ground station and transformed into images.
3. Errors in brightness and geometry are removed, and map coordinates are added.
4. Data ready to be interpreted either by a human or with computer assistance (we now have imagery of high integrity, ready for use (the focus of this thesis).

Figure 1 shows a typical remote sensing system consisting of a sensor that can detect and measure energy transmitted from the Earth's surface through the atmosphere. The data is then transmitted to the ground station where it is processed to image products ready for

application by the users. Useable information is extracted from this processed image data obtained.

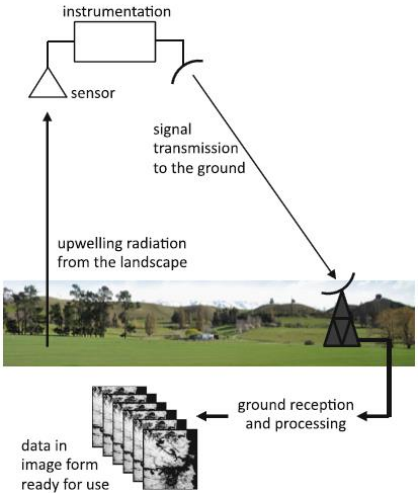


Figure 1: A typical RS system

1.1.2 History

History of the field can be traced back to 1859 when photographs were taken from a balloon. Active RS instruments such as radar, sonar, and thermal infrared systems were used in WW2. The space era of RS started in the 1960s when remote sensors were used in satellites for earth observations. Landsat-1, launched in 1972, was the first satellite dedicated to collecting information about the earth's surface.

1.1.3 Types of Remote Sensing (Based on sensors)

RS can be classified as active RS and passive RS based on the sensors being used for the exploration of objects. Passive sensors don’t have an energy source to illuminate targets being observed but rather measure reflected or emitted energy from the objects being observed. Reflected sunlight is the most common source of radiation that passive sensors measure. An example of a passive sensor/instrument is a photographic camera. In contrast, an active sensor contains its source of energy through which it scans its target area and measures the reflected energy from the scanned area, e.g. a camera with a flash and RADAR, etc.

1.1.4 Types of Remote Sensing (Based on the spectrum)

In RS, thermal, optical, or Synthetic Aperture Radar (SAR) is usually used as an imaging sensor. The optical sensor is sensitive to the radiations reflected/emitted from the surface of the earth, ranging from visible to mid-infrared of the EM spectrum. Both, optical and thermal imaging sensors are passive. SAR is an active sensor and is sensitive in the microwave range of the EM spectrum [2].

RS imaging can be classified into Multispectral imaging (MSI) and Hyperspectral imaging (HSI). HSI & MSI images have 3D structures with 2D representing spatial dimensions (x,y) while the 3rd dimension represents the spectral content [4]. They are differentiated by the number of spectral bands they contain. In MSI remote sensing near-infrared, visible, and short-wave infrared images are acquired and usually, these satellite images contain 2 to 13 bands. It is in use since the 1970s. On the other hand, in hyperspectral remote sensing, a wider electromagnetic spectrum is analyzed instead of just assignment of primary colors to each pixel. Hundreds of bands are contained in HSIs with a narrower spectral bandwidth of 10-20 nm. HSIs contain more data compared to MSIs. MSI and HSI are further demonstrated in Figure 2.

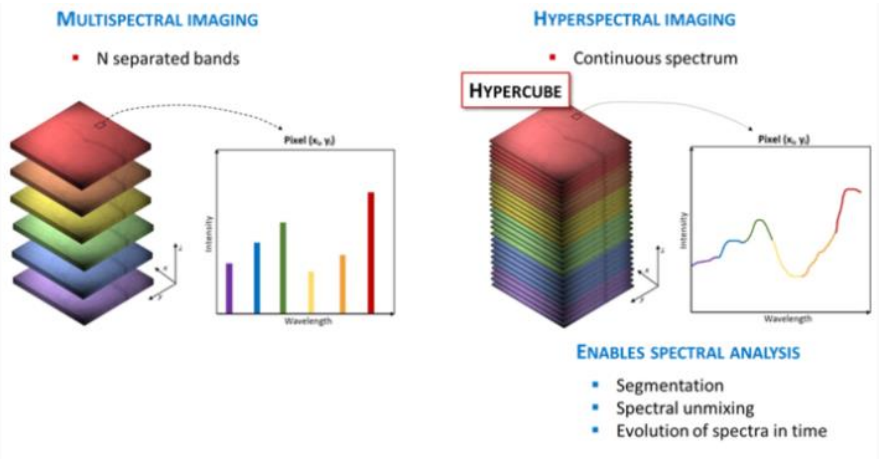


Figure 2: Difference between MSI and HSI

MSI and HSI data is a source of rich information that provides opportunities for many applications but this also offers some challenges [2]:

1) Due to high dimensionality, processing a large amount of data becomes computationally expensive.

2) Limited labeled training data availability.

1.2 Land-use Land Cover Classification

The use of RS data in LULC classification dates back to the 1940s when Francis J Marschner used aerial photography to map the whole US based on land uses [5]. With the start of the Landsat program in the 1970s and the use of multispectral scanners (MSS), studies relating to RS imagery data classification raised to a new high [6]. LULC images classification in the RS domain is important and has been applied in many applications including change detection, environment monitoring, and urban planning [1]. LULC mapping can't be denied in the planning of cities, better management of land, and conservation of natural resources [7]. RS has greatly helped in the mapping of LULC.

1.3 Classical ML Techniques for RS images Classification

ML techniques have been used extensively in the context of RS. 60000 papers were published from 2004 to 2015. Classification algorithms were applied in 10000 while the regression model was applied in 3000 papers. Applications in which ML methods are used in the RS context include classification of LULC, object detection, etc [8]. Random Forest (RF), Artificial Neural Networks (ANN), Support vector machine (SVM), and decision trees (DT) are the most common ML algorithms used in RS [9, 10]. Different ML techniques are being used for LULC classification e.g. SVM and RF but these techniques usually do not achieve the required accuracy for LULC classification [7].

Authors in [11] conducted a study and the accuracies of six ML classifiers were examined for LULC mapping. The classifiers examined included RF, spectral angle mapper (SAM), Mahalanobis distance (MD), SVM, fuzzy adaptive resonance theory-supervised predictive mapping (Fuzzy ARTMAP), and ANN. It was concluded that all these classifiers had almost similar accuracies; however, RF had the highest. DL methods are neither discussed nor compared in the study.

Machine learning (ML) has been in use in RS image analysis for many years. ML algorithms such as random forests (RF), decision trees (DT), support vector machines (SVM), and others, have been in use since the early years for MSI and HSI classification tasks. Most of the classification approaches for their training prior to classification rely, on extracted features of an image. These features can be classified as low-level, mid-level, and high-level. Feature extraction (FE) techniques used for low-level features included histogram-oriented gradient (HOG) and scale-invariant feature transform (SIFT), and bag of visual word (BoVW) has been an important technique for mid-level features extraction.

Extraction of features is independent of ML classifiers in these classical ML methods. Due to the very high dimensionalities of MSI and HSI data, extracting the best feature representations becomes difficult and time-consuming.

To achieve the best performance in terms of overall accuracy, computational time, and memory, these classical methods rendered limitations.

1.4 DL Techniques in RS

Due to technological advancements and global/regional coverage of the earth has resulted in huge amounts of data through RS, processing this big data is challenging as fast and efficient image processing techniques are required. Classical ML techniques render limitations at this point and DL comes to play its role.

Deep learning (DL) is a subfield of ML that employs a deep neural network. Since 2006, DL has emerged as a new area of ML research. DL techniques have efficiently solved many complex tasks in different areas like signal processing, image processing, and natural language processing (NLP).

ML community has diverted its attention to DL techniques as they are successful in many applications. Although DL techniques were introduced earlier but could not be used efficiently. One of the reasons was the unavailability of large datasets as DL requires huge amounts of data for training. Secondly, high computing power was required. Nowadays, DL techniques are on the success due to meeting these challenges and progress in DL research [12]. Three main characteristics of DL are: 1) extracts features itself from data, 2) learns hierarchical features, and 3) is more generalizable [13].

DL has been successful in various RS applications since its rise, resultantly, the RS community has shifted its attention from other ML techniques to DL [2, 10]. Due to the successes of DL in many RS imagery data analyses including object detection, LULC classification, scene classification, segmentation, etc, the use of DL has grown significantly since 2014 [14]. DL methods such as Generative Adversarial Networks (GAN), CNNs, and Recurrent Neural (RNN) have been used for RS image data classification [1]. DL is successful when used with "Big data", hence it can mine the hidden information from such complex huge data [2].

Authors in [13] have described Auto-encoder (AE), CNNs, Deep Belief Network (DBN), RNN and Deconvolutional Neural Network (DeconvNet) as contemporary mainstream DL architectures. These architectures are introduced shortly as:

AE: This network learns features from unsupervised data. Dimensionality reduction, required in different RS applications, was one of its earlier applications [13].

DBN: It is a combination of probability and graph theory and is a form of the Probabilistic Graphical Model (PGM). More precisely, it is a deep-directed acyclic graph (DAG) [13].

RNN: In RNN the connections are formed in directed cycles. As opposed to AE and CNN, RNN has memory. Its main applications include time-series and speech analysis [13].

CNNs: It is inspired by the human visual cortex. Typically, it contains multiple layers of convolution followed by pooling and finally followed by fully connected layers. CNNs are leading in images classification and Computer Vision (CV) community has shown special interest in CNNs. LeCun developed the first CNN 'LeNet' in 1990 for reading zip codes. Other CNN models include VGG16, VGG19, GoogleNet, etc. Till now CNNs have better results on different benchmarks CV datasets [13].

CNNs have the ability to extract high-level features, due to which CNNs methods utilization is on the rise in image detection, classification, and segmentation [15]. Useful technological advancements have made the acquisition of high-resolution satellite images easier, resultantly, CNNs are widely used in LULC classification [7]. As compared to other DL methods, CNNs have achieved better classification accuracies in image classification.

Since there is no free lunch [16], applying CNNs in RS image classification offers some challenges e.g. a large amount of labeled training data is required which is difficult in the case of RS. Computational cost is high for training a CNN model from scratch [7].

1.5 Data Sources

RS data is made available timely and publicly by many satellite and aerial imagery providers [2]. Most popular among these providers are IPMUS Terra and NOAA, USGS, NASA Earth data, NEO, and Copernicus open Access hub. Satellite images used in the literature are mostly obtained from Sentinel-1, Sentinel-2, Landsat-8, Landsat-7, PROBA-1, WorldView-3, WorldView-2, EO-1, QuickBird, and SPOT-6 satellites [2].

1.6 Challenges

Despite the availability of a huge amount of data and DL techniques for LULC image classification, some challenges are rendered related to limited resolution, ground truth, and nature of the data that impacts classification performance. Other challenges include high dimensionality, data acquisition noise, and data redundancy [2]. Another challenge is that RS images have fused boundaries which makes the generation of discriminative feature representation difficult for the classification tasks [17].

1.7 Problem Statement

The classification frameworks are highly dependent on the available features - extracted either using the classical or deep methods. Therefore, the extracted features should be providing the most discriminant information in order to achieve a high classification rate.

A few problems in this domain that need our focus are:

- Existence of irrelevant and redundant features
- Limited training data availability
- Curse of dimensionality

1.8 Major Contributions

Instead of training models from scratch, TL is used for feature extraction. Thus, the limited training data availability problem is addressed. Secondly, the features fusion approach is used to fuse the extracted features from different pre-trained CNN models to obtain a new

feature set. Thirdly, two-level features selection strategy is devised for the selection of the most discriminative features.

1.9 Thesis Organization

The thesis is organized as follows: the first chapter is an introduction to RS, LULC classification, and the use of classical and deep learning approaches for land-use classification. In chapter 2, a brief literature review about deep learning techniques used for remotely-sensed images classification, feature fusion, feature selection, and reduction are presented. In chapter 3, the proposed scheme is discussed. In chapter 4, experimental results and analysis are presented. The conclusion of this work is presented in chapter 5.

Chapter 2
Literature Review

Different techniques have been discussed in the literature to improve the classification accuracy of LULC images.

Authors in [18] proposed a hybrid hot encoding VGG19 (HEVGG19) method for the classification of LULC changes. The proposed model is then trained with ResNet50 through TL by replacing the last layer. Stratified cross-validation was used for the evaluation of model performance. The images have been classified using MLR classifier. Feature Pyramid Network (FPN), a feature extractor has been used for object detection.

Authors in [10] reviewed the application of CNNs in RS data analysis. CNN was used for vehicle detection for the first time in 2014 using MS satellite images [19]. Since then application of CNN is on the rise in RS applications compared to other DL techniques. VGG and its variants are the most used models. Tensor Flow is the most used framework. With CNNs, classification accuracy is improved when multisource data is used and images with a spatial resolution of greater than 5m. Since huge amounts of data are required to train the network, researchers have used TL to handle this problem. Most of the studies have focused network architectures designs rather than time efficiency, and are now leading in RS imagery data analysis among the research community.

Authors in [3] have presented a novel dataset 'EuroSAT' for LULC classification as the existing datasets available for LULC are either small or depend on data sources that are not allowed publicly for applications of this domain. Sentinel-2 satellite images that cover 13 spectral bands have been used for the generation of the dataset. Furthermore, the images are MS, geo-referenced and its earth observation data is freely accessible. For benchmarking of the dataset, pre-trained CNN models ResNet-50 and GoogleNet were used for the classification of the introduced LULC classes. OA of 98.57% was achieved with ResNet-50 and 98.18% with GoogleNet.

Authors in [1] studied DL methods including CNNs, RNN, and GAN in analyzing RS data for LULC classification. They concluded that DL methods perform better than conventional ML methods, and among DL techniques, CNNs are the best performers in LULC classification.

Authors in [17] proposed a method for satellite images classification. In the proposed method discriminative feature representations were generated by combining distinct

information of deep features. The features were extracted from the selected layers of pre-trained CNN models VGG-16, AlexNet, VGG19, GoogleNet, and ResNet using TL. The selected features were fused prior to feature reduction. The features reduction process was carried out in two steps: 1) high-ranked features were selected using Shannon entropy and 2) for dimensionality reduction of fused features, NCA is used. The selected features prior to fusion and then after fusion were trained using SVM and KNN classifiers on benchmark datasets i.e. UC Merced, RS19, and AID. They have compared their proposed method with other existing methods and achieved maximum accuracy of 99.1 % while utilizing less than 5% of features on the RS dataset.

Authors in [7] have studied three techniques i.e. full-trained CNNs, pre-trained CNNs, and fine-tuning CNNs for LULC classification. In a full-trained approach, the CNN model has been trained from scratch. In the second technique, a pre-trained CNN model is used as a feature extractor and an SVM classifier is used for classification. In the third technique i.e. fine-tuning, the pre-trained CNN model is fine-tuned. They used two CNN models i.e. AlexNet and GoogleNet and two datasets i.e. UC Merced and Brazilian coffee scenes. 6-fold cross-validation is used for experimental data. Results showed that training from scratch is not always the best solution. Fine-tuning performed better than the other two approaches. The best accuracy (96.83%) was achieved using the fine-tuning approach with GoogleNet on the UC Merced dataset.

Authors in [20] proposed a framework for LULC classification which used an ensemble of CNNs. Sentinel-2 dataset images which consist of multispectral bands were distributed into different sets under different scenarios. The framework was designed in such a way that pre-trained AlexNet was fine-tuned with each set of bands separately, resulting in feature vectors F1…Fn (depending on the number of sets). Then these feature vectors were merged to form a single feature vector which was then classified using SVM, RF, or KNN (depending on the scenario). The highest accuracy of 96.83% was achieved with three sets and using an SVM classifier.

Authors in [13] reviewed theories, tools, and challenges offered by DL in RS. The authors have surveyed various papers and presented the challenges and theory, and the tools used.

Authors in [21] have compared the classification performance of four ML algorithms i.e. RF, SVM, DL, and Xgboost in boreal landscapes using Sentinel-2 images. It was concluded that the OA accuracy of all these algorithms was close to each other, however, SVM produced the highest accuracy. DL architecture implemented was a multilayered feed-forward DNN.

Table 1: Summary of Literature Review

Year	Auth.	Objective	Proposed framework	Dataset	Results
2018	[17]	Combining distinct information of deep features for generation of Discriminative feature representations	AlexNet, VGGNet-16, VGGNet-19, GoogleNet, ResNet, SVM, KNN	UCMerced, AID, RS19	99.1% (Highest accuracy) with KNN and UC Merced
2021	[18]	VGG19 has been used with hot encoding process. The proposed method (HEVGG19) is trained with ResNET50 through TL.	HEVGG19 and RESNET50 for TL, RF for classification	Sentinel-2, Google Earth	98.5%
2019	[3]	Presented a novel benchmark dataset EuroSAT covering 13 spectral bands for LULC classification	ResNet-50, GoogleNet, Sentinel-2 Satellite images	EuroSAT	98.57% with ResNet-50 and 98.18% with GoogleNet
2019	[7]	Three different modalities of CNNs i.e. Full-trained, pre-trained, and Fine-	AlexNet, GoogleNet	UC Merced, Brazilian	96.83%, Fine-tuned with

		tuning have been evaluated for LULC classification of high-resolution RS imagery		coffee scenes dataset	GoogleNet and UC Merced
2019	[20]	An ensemble of CNNs for features extraction for different spectral bands	AlexNet, SVM, RF	Sentinel-2	89.43%
2020	[21]	Comparison of LULC classification performance of four non-parametric ML algorithms in a boreal landscape	SVM, RF, Xgboost, DL	Sentinel-2 Images	SVM produced better results

2.1 Pre-trained CNN Models

Inception-ResNet-v2:

Inception-ResNet-v2 presented by [22] is one of the variants of the inception family of architectures. The model has been trained on imagNet database. It has 164 deep layers and 1000 different object categories e.g. pencil, keyboard, mouse, etc. can be classified with the model. The image input size for this network is 299x299.

DenseNet-201:

This CNN is 201 layers deep. The model has been trained on the imagNet database and as a result, a wide range of feature representations has been learned by the network for a wide range of images. 1000 different object categories e.g. pencil, keyboard, mouse, etc. can be classified with the model. The image input size for this network is 224x224. The model was proposed by [23].

NasNet-Mobile:

It was introduced by Google. This network has an input image size of 224x224. The model has been trained on imagNet database. 1000 different object categories e.g. pencil, keyboard, mouse, etc. can be classified with the model.

2.2 Datasets

Classification accuracy of supervised ML algorithms strongly depends on high-quality datasets availability with having enough classes [3]. In their review, [1] described AID, UC Merced, EuroSAT, and NWPU-RESISC45 as the mostly used datasets for the classification of LULC images. These datasets are briefly discussed below.

EuroSAT: The EuroSAT dataset was introduced by [3]. The dataset contains Sentinel-2 satellite images. These images cover 13 spectral bands and consist of 10 classes, 27000 labeled, and geo-referenced images. The images have a spatial resolution of 10m/pixel.

AID: The Aerial Image Dataset (AID) was introduced by [24], and is an RS dataset for aerial scene classification. There are 30 classes with 220-420 images/class. The images are 600 x 600 pixels with a varying spatial resolution of 0.5m/pixel to 8m/pixel. The number of images in the dataset is 10000. The images were selected from Google Earth Imagery and are multisource i.e. taken from different RS sensors. To increase the intraclass diversities, the images are chosen from different countries and regions at different times, seasons, and imaging conditions.

2.3 Features Extraction

Converting input data into a set of features is called feature extraction [12]. CNN models have been in use for feature extraction [17]. Various convolutional and pooling layers are used for the extraction of invariant, discriminant, and nonlinear deep features from images [25]. For this purpose, various techniques such as transfer learning (TL) and features descriptors are used [26]. Thus, Low-level features are determined in shallower layers and high-level features in the deeper layers [17].

2.4 TL

The challenge of limited training data availability can be overcome by using 1) TL and 2) Data Augmentation [13]. Despite CNNs' features extraction capability, it is not feasible to train CNN from scratch with small datasets [15]. Authors in [27] observed that image features learned in initial layers are general and are independent of the datasets on which the model is trained. Progressing through the layers, features are transitioned from generality to specificity on the dataset trained [27]. These transitions have resulted in TL [15].

Data Augmentation improves the diversity of the dataset by adding variations to each image in the dataset. It improves the accuracy at the cost of computational time as demonstrated by the authors in [15].

2.5 Features Selection

FS is a process in ML in which available subsets from the data are optimally selected for modeling [28]. Features in the data can be categorized as relevant, irrelevant, and redundant [28, 29]. FS methods play important role in efficient data reduction which is useful in finding accurate ML algorithms [30, 31]. The dimensionality of data can be reduced by selecting only a subset of features to create a model. Depending on the constraints such as excluded or required features, FS algorithms search for a subset of features that optimally model. However, these methods should be efficient computationally and sensitive to patterns of association so that useful and informative features are retained [32]. The main advantages of FS include 1) improved prediction performance by identifying the most discriminant features, 2) reduction of computational cost by removing redundant and irrelevant features, and 3) understanding the data generation process in a better way [29]. Despite being relevant and informative, using too many features can degrade prediction performance. FS techniques can be categorized as Filter, wrapper, and embedded techniques [33]. Common filter methods include Information Gain, Correlation, Chi-Square, ReliefF, and ReliefC to name a few [30].

In FS, redundant and irrelevant features are removed to obtain an optimal feature set. Such a feature set improves classification algorithms' efficiency and accuracy [34].

There is no FS algorithm that is universally accepted as the best for all tasks [32]. It means that selection of an algorithm depends on the application.

2.6 Features Fusion

Utilizing various features from different layers, which results in a new feature set, is called feature fusion [17]. Since a single-layer approach may inherit some shortcomings which will resultantly affect classification accuracy [26]. Therefore, utilizing the feature fusion approach will overcome this limitation. The main objective of feature fusion is that the

most relevant and comprehensive features should be passed on to the classifier. Feature fusion and FS should be moved together to achieve better classification accuracy [17].

2.7 Cross-validation

It is a model assessment technique used for evaluating the performance of ML algorithms in predicting new data. In this technique, training data is partitioned into training and testing subsets. The training subset is used to train the algorithm and the complementary subset is used to evaluate the performance of the model.

2.8 Performance Evaluation Measure

A confusion matrix will be used for the performance evaluation of the proposed methodology. Classification accuracy will be determined from the equation given below.

$$\text{Accuracy} = \frac{True\ Positive + True\ Negative}{Total\ samples}$$

Chapter 3
Proposed Methodology

The proposed methodology for the classification of remotely sensed images is shown in Figure 3. Pre-trained CNNs DenseNet-201, Inception-ResNet-v2, and NASNet-Mobile have been used for features extraction through TL. This step is followed by the fusion of different extracted features. Since the features fusion step results in increased dimensionality, the two-level features selection approach is employed in the next step. Various classifiers are used at each step and classification accuracy and training times are compared. Each of these steps is elaborated in detail in the following sections.

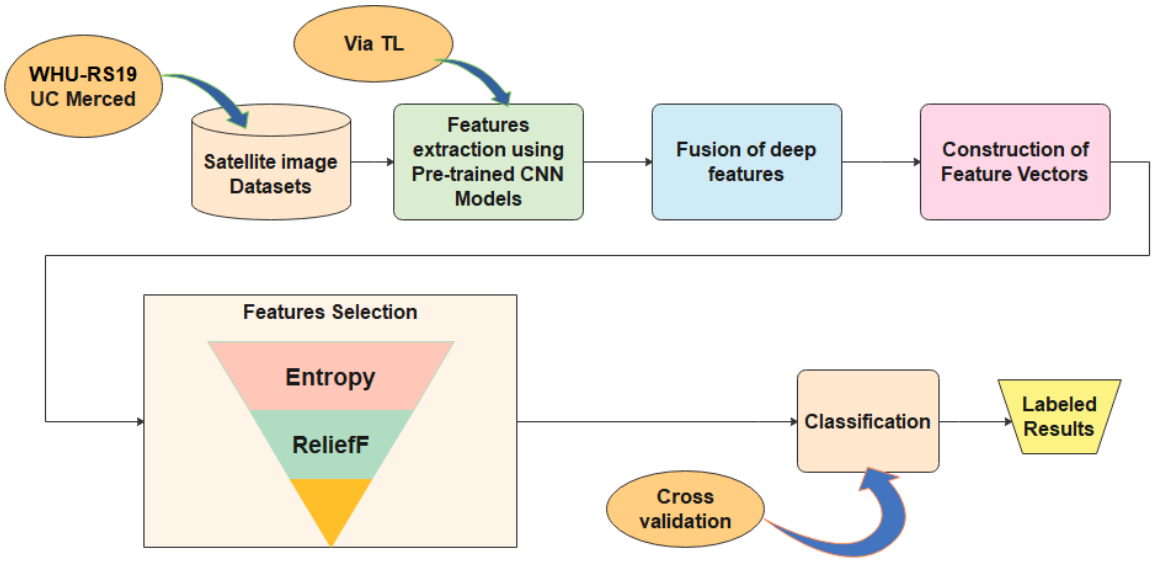


Figure 3: Flowchart showing the proposed methodology

3.1 Feature Layers

Features of any image can be extracted using a pre-trained CNN via TL. Different CNNs layers provide different features. Transferred weights are kept frozen on their initial values during the training phase. Layers choice is also important e.g. FC layers, activation layers etc. in this study, single FC layer is selected from each pre-trained model. Selected layers and their notations are given in table.

Table 2: Selected layers and their proposed notations

Pre-trained Model	Layers used	Feature vector notation
DensNet-201	Avg_pool	FV1
Inception-ResNet-v2	Avg_pool	FV2
NASNet-Mobile	Global_average_pooling2d_1	FV3

TL:

TL is the application of skills and knowledge learned in solving one set of problems (source) to a different set of problems (target) [17]. It can be very helpful when the training dataset is comparatively very smaller than the dataset of the source domain [35].

Given source domain $D_S = \{(d_1^s, e_1^s), \dots, (d_i^s, e_i^s), \dots, (d_n^s, e_n^s)\}$ with a learning task and source labels are $D_L = L_D, L_S, (d_m^s, e_m^s) \epsilon R$.

The target domain $S_t = \{(d_1^t, e_1^t), \dots, (d_i^t, e_i^t), \dots, (d_m^t, e_m^t)\}$ with the learning task and target labels $S_L = L_S, L_T, (d_n^t, e_n^t) \epsilon R$. Where $n \ll m$ (m, n) is the training data sizes and $e_1^D \; and \; e_1^S$ is the labels of training data. The main purpose of TL is the improvement of learning ability of the target predictive function D_T by transferring knowledge from D_S to D_T. CNN models shown in Table 2 are trained on the ImageNet dataset [36] and are re-used on satellite image datasets shown in Table 3. The TL approach is further elaborated in Figure 4.

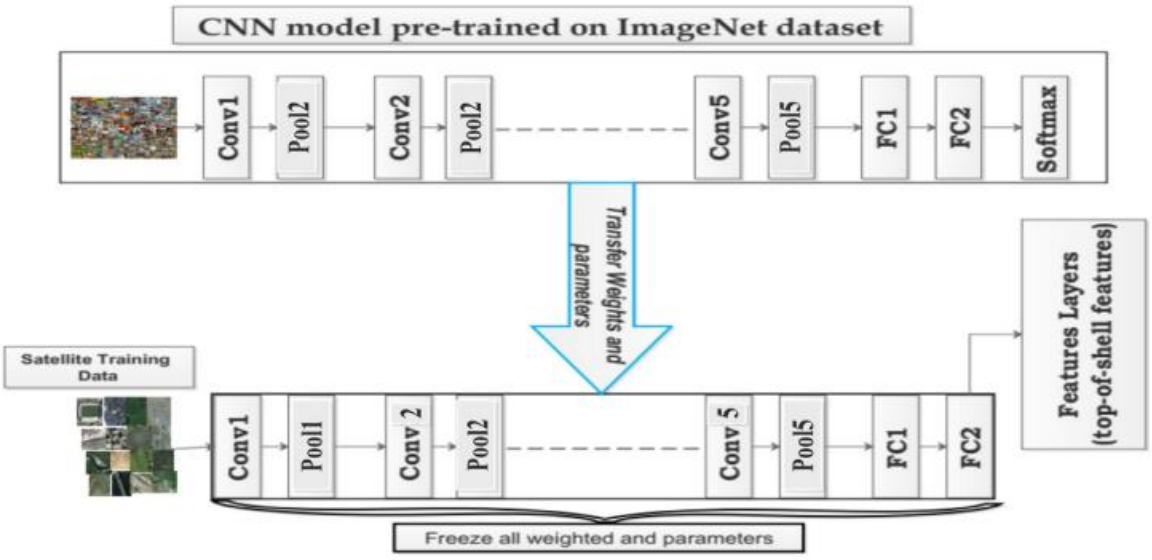


Figure 4: Features extraction from pre-trained CNN via TL

3.2 Fusion Mechanism:

Since features extracted in different layers from different CNN models exhibit different visual characteristics. Therefore, by utilizing these multiple features, classification accuracy would be improved. The main objective at this point is to study the behavior of the classifiers when the features from different models are combined. In this work, the extracted features are directly concatenated thus forming a joint feature space.

Let us consider that different layers are selected from different CNN models, then the final feature vector is given by FV $= \{FV_i^k\}$ or $\{FV_j^m\}$, where

$i = j \in \{1,2,3\}$ represent different selected layers from pre-trained CNNs

$j = m \in \{1,2,3\}$ represent different CNN models

The fused feature vector thus generated is given by

$FV^\gamma = \{FV_i^k\}$ or $\{FV_j^m\}$ which is having M dimensions.

Since feature fusion results in higher dimensionality, better performance compared to using single layer is not guaranteed due to redundancy in features from multiple layers. Therefore, feature reduction becomes important for achieving better classification accuracy.

3.3 Feature Selection Techniques

A two-level features selection strategy is proposed to achieve the best classification accuracies with a minimum number of features which resultantly reduces the computational burden. The proposed strategy utilizes Entropy followed by the ReliefF algorithm. This strategy results in a reduced number of features and better accuracy. The feature selection techniques used in this work are discussed below:

Entropy:

Entropy is a measure of randomness in a signal and reflects disorder of a system. This ability can be exploited to select the most relevant features. Ranks are assigned to the features and are arranged according to their contribution [37].

Considering a training set given by: $\mathrm{Tr} = \{(x_1, y_1), \dots, (x_i, y_i), \dots, (x_M, y_M)\}$ having M labels where $X = \{x_i\}_{i=1}^{M} \in R^c$ is a c-dimensional FV and $Y = \{y_i\}_{i=1}^{M}$ shows class labels with $y_i \in \{1,,2,3,\dots C\}$. Then entropy for this is given by

$$E(X) = -\sum_{n=1}^{\infty}(x_a)\log(6(x_a))$$

Where $\sigma(x_a)$ is the probability of observing a specific x_a of X.

Dominant features can be selected using entropy. After ranks are assigned to all the features, they are arranged according to their contribution, and only the high-ranked features are selected while the remaining are discarded/neglected. Thus, the existing features are down-sampled with preserving the original information for the second features reduction step.

Relief: Relief is a FS algorithm based on the filter-method and was developed by Rendell and Kira in 1992. It was designed for binary classification problems having numerical or discrete features. This algorithm calculates a score for each feature and based on the score, top-ranked features can then be selected. The output of the algorithm for each attribute is -1 to 1. The relief algorithm is rarely used in practice, but since then many variants of the algorithm have been developed which are more efficient.

Relief is capable of detecting feature dependencies. Instead of searching through feature combinations, the nearest neighbor concept is used for the derivation of feature statistics

that indirectly determine the interactions. Since RBAs are filter-based, they are fast. The selected features can be used with different algorithms, thus saving computational time when different modeling techniques are to be used. feature sets of different sizes can be selected [32].

It calculates features score or weights. A score is calculated for each feature that can be used to estimate feature relevance to the target. Each feature's relevance to the target is estimated by statistical means which is called feature weight or score that ranges from -1 (worst) to +1 (best) [32].

The original Relief algorithm was limited to binary classification; therefore, it is rarely used in practice. However, it has resulted in many more efficient variants. These include ReliefA, ReliefB, ReliefC, ReliefD, ReliefE, ReliefF, RRelieifF, Relieved-F, Iterative Relief, I-RELIEF, TuRF, EReliefF etc. [32]. Collectively, these algorithms are termed Relief based algorithms (RBSa). Relevant feature selection is addressed in these algorithms; however, these do not discriminate among redundant features.

ReliefF: The mostly utilized variant of RBA is ReliefF. Multiclass problem is addressed in ReliefF algorithm and is not limited to binary classification. This algorithm can also handle noisy and incomplete data. Since the multiclass problem is addressed in this algorithm, therefore, it has been used in this work.

3.4 Datasets

UC Merced: The University of California Merced (UC Merced), the dataset was introduced by [38] which consists of 21 land use and land cover classes: airplane, buildings, dense residential, beach, freeway, intersection, harbor, medium density residential, chaparral, mobile home park, agricultural, parking lot, overpass, river, golf course, sparse residential, runway, tennis courts, baseball diamond, storage tanks, and forest. Each class contains 100 images of 256 x 256 pixels with a spatial pixel resolution of 0.3048m. The images were extracted manually from large images which were downloaded from the United States Geological Survey (USGS) National Map. The images were taken from an aircraft and are in RGB color space [3].

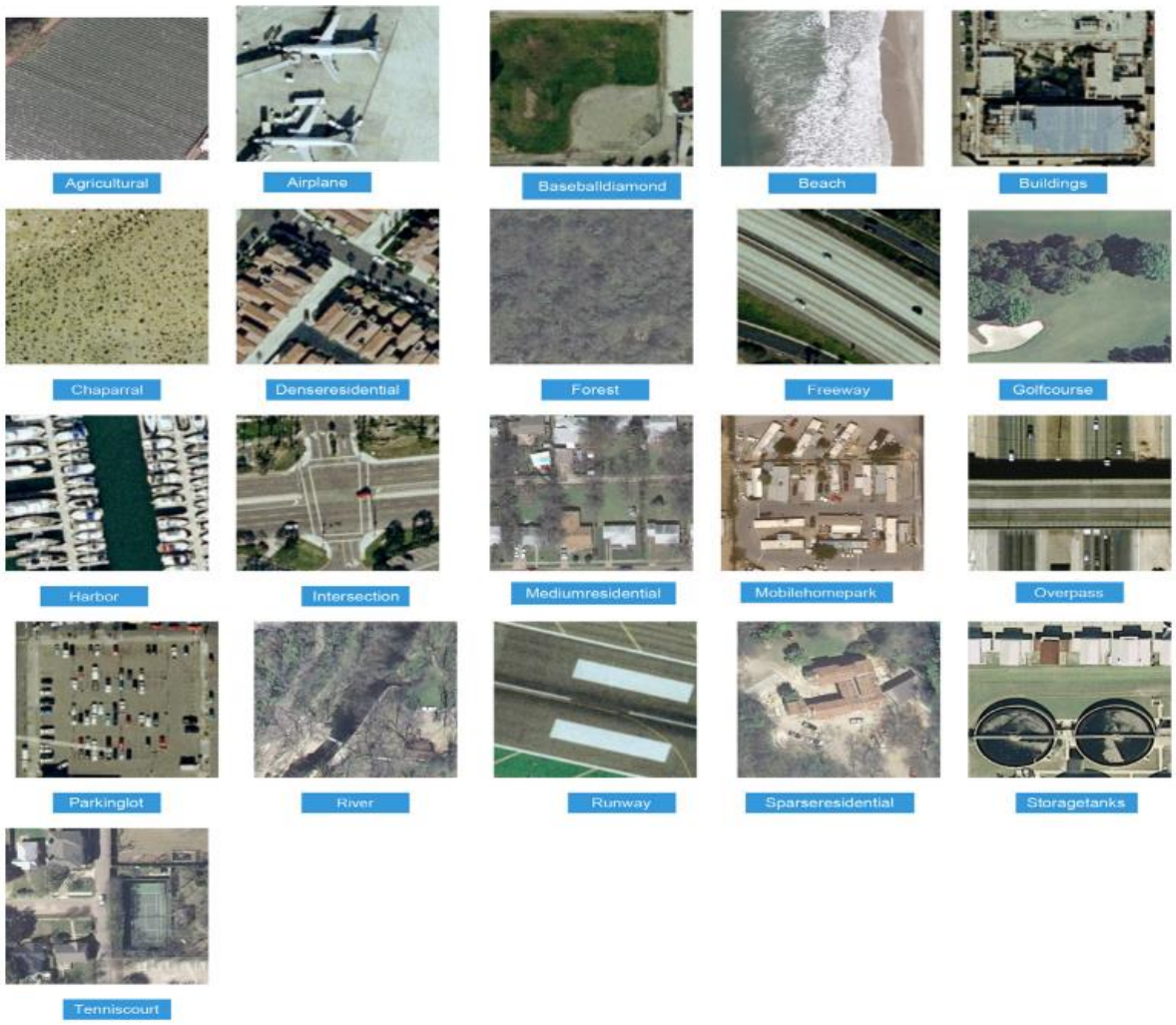


Figure 5: 21 Classes of UC Merced Dataset

WHU-RS19: This dataset contains 19 classes of high-resolution satellite images, a total of 1005 images with RGB spectral bands, and a spatial resolution of up to 0.5m. Image dimensions are 600 x 600 pixels. The classes include beach, commercial, river, bridge, desert, forest, farmland, meadow, port, parking, industrial, pond, park, railway station, mountain, viaduct, football field, residential, and airport. Due to variations in orientation, resolution, illumination, and scale, the dataset becomes challenging [39].

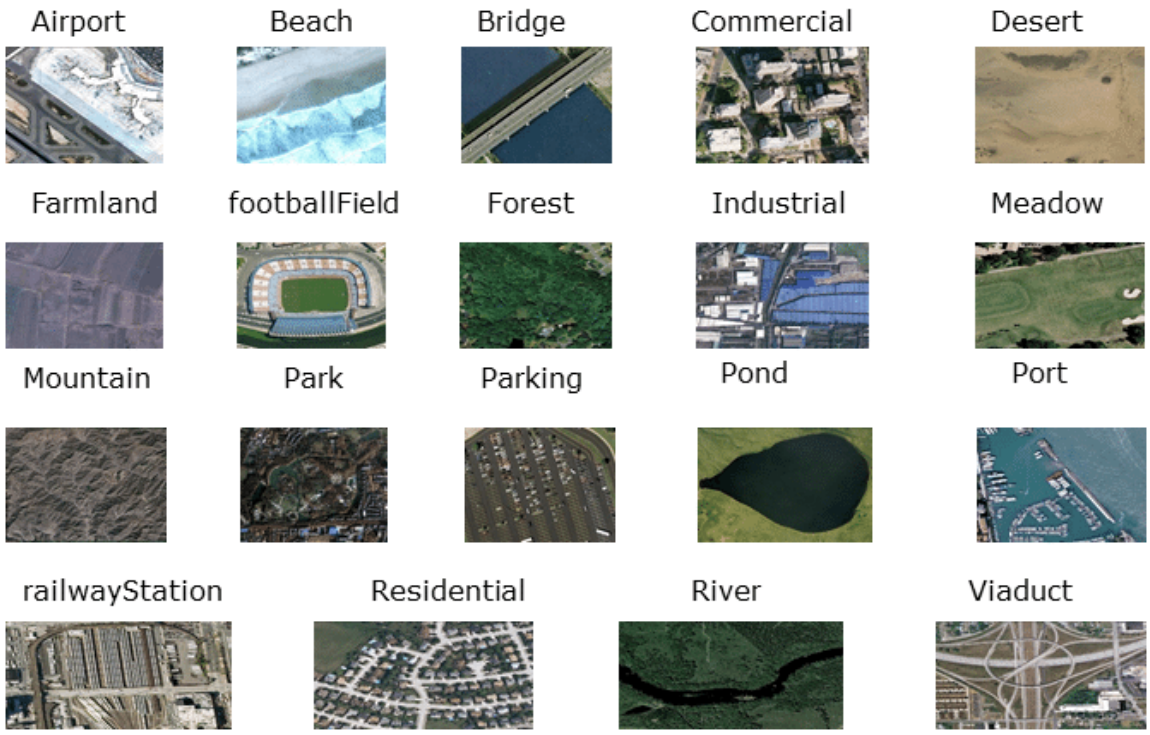


Figure 6: 19 Classes of WHU-RS19 Dataset

Table 3: Partitioning of datasets

Dataset	Size of dataset (Number of images)	Size of training set	Size of testing set
WHU-RS19	1005	704	301
UC Merced	2100	1470	630

Chapter 4
Results and Discussion

4.1 Simulation Results and Discussion

All the simulations are performed on two benchmark datasets i.e. WHU-RS-19 and UC Merced with a selected set of training and testing samples, as shown in Table 3. Several classifiers are used for classification which include Linear Discriminant, Ensemble Subspace Discriminant, Linear SVM, Quadratic SVM, Cubic SVM, Medium Gaussian SVM, Wide Neural Network, Medium Neural Network, and Cosine KNN. For extraction of selected layers from pre-trained CNN models via TL, a corei7 desktop computer with MATLAB 2021a, 16GB of RAM, and 8 GB of GPU is used. While the analysis was carried out with MATLAB 2021a on a corei7 desktop computer with 64GB of RAM.

Simulations are performed in the following sequence for both the datasets separately:

1. Classification results with single layer features.
2. Classification results with features fusion.
3. Classification results with the selection of features using Entropy.
4. Classification results with the selection of features using Entropy followed by ReliefF.

4.2 Results with WHU-RS19 Dataset

In this section simulation results with the WHU-RS19 dataset are discussed. It should be noted that class labels are represented as numeric class labels (1 through 19 for all the 19 classes in confusion matrix).

4.2.1 Results with Single Layer Features

It can be observed that features extracted from pre-trained DenseNet-201 (FV1) generated better results followed by Inception-ResNet-v2 (FV2), and then NASNet-Mobile (FV3), irrespective of the classifier. It is also evident that the Ensemble Subspace Discriminant classifier outperforms all other classifiers in terms of classification accuracy, but at the cost of maximum training time among the classifiers. The most efficient classifier in terms of computational cost is Linear Discriminant that takes the least training time of 6 seconds. Table 4 shows the classification accuracies of individual layers extracted from pre-trained CNN (shown in Table 2) and Figure 7 shows average classification accuracy of each extracted layer.

Table 4: Accuracy with different layers of pre-trained CNNs used as feature extractors on RS-19.

Classifier	**FV1 (Dimension = 704 x 3456)**		**FV2 (Dimension = 704 x 1536)**		**FV3 (Dimension = 704 x 1056)**	
	Acc. (%)	Tr. Time (sec)	Acc. (%)	Tr. Time (sec)	Acc. (%)	Tr. Time (sec)
Linear Disc.	*	*	94.7	8	87.4	6
En. Sub. Disc.	**98.7**	131	**95.7**	129	**93.4**	85
Linear SVM	96.7	60	90.7	44	92.7	36
Q SVM	96.7	68	91	50	92.7	40
Cubic SVM	96.7	69	91.4	59	92.7	44
MG SVM	96.7	86	88.7	70	88.4	47
Wide NN	97.3	81	93.7	91	92.4	59
Medium NN	96	80	92.7	85	89.4	52
Cosine KNN	96	53	86	65	87	36
Avg.	**96.9**	79	**91.6**	67	**90.7**	45

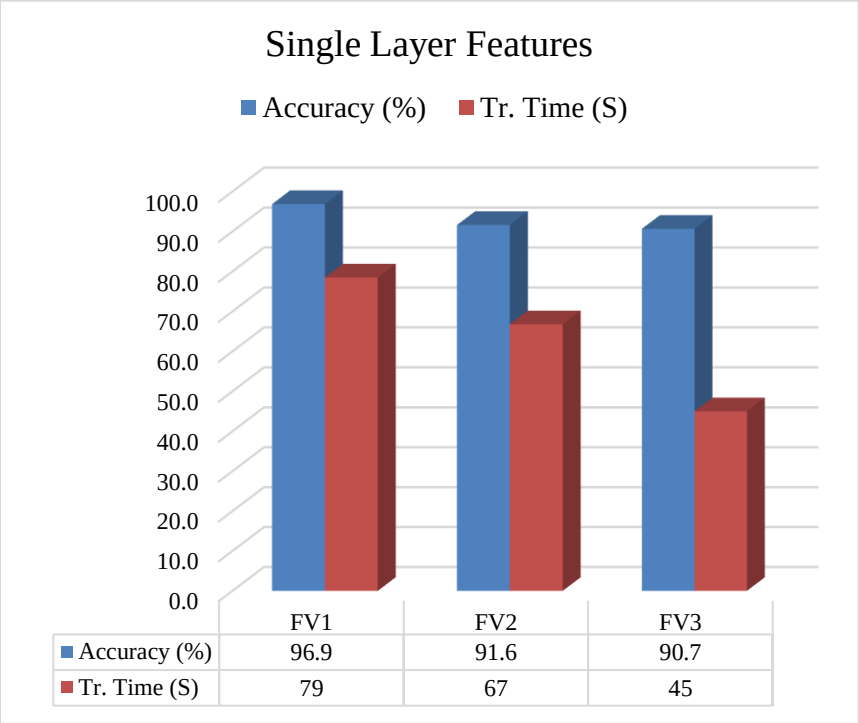


Figure 7: Average classification accuracies with single-layer features (WHU-RS19 dataset)

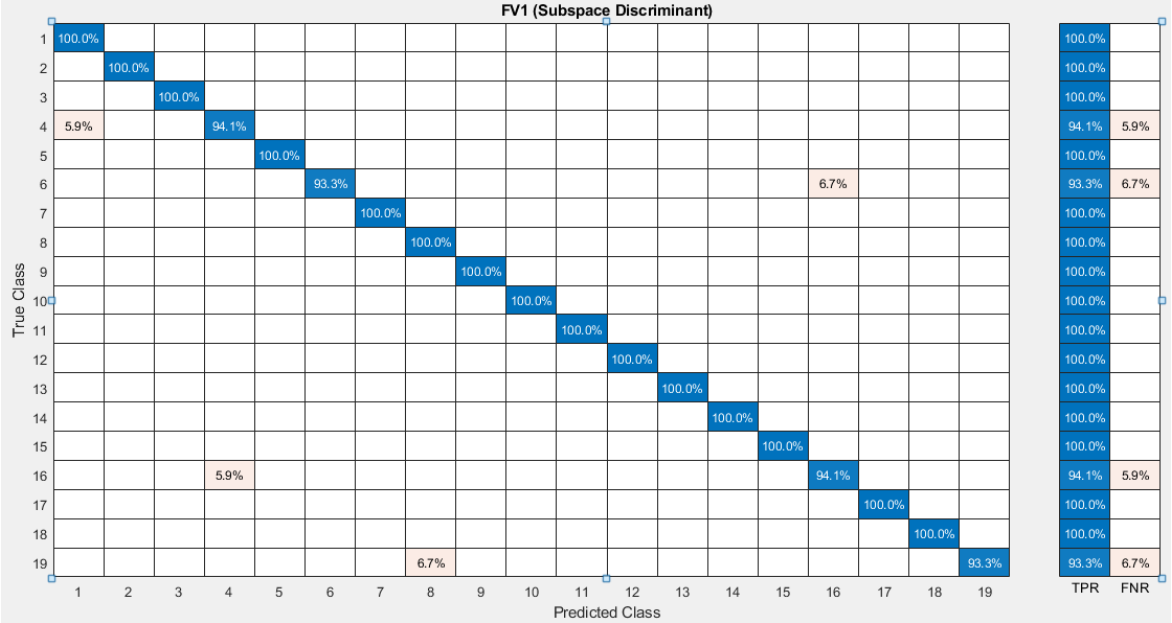


Figure 8: Confusion matrix of Ensemble Subspace Discriminant using extracted layer (FV1) of DensNet-201

The confusion matrix for FV1 and classifier Ensemble Subspace Discriminant is shown in Figure 8.

After analyzing the classification performance of single-layer features with different classifiers, the classification performance of fused features will be analyzed.

4.2.2 Results with Features Fusion Technique

In this approach, the features which are extracted from different pre-trained CNNs, i.e. FV1, FV2, and FV3 are concatenated before feature selection. The features are fused in four different configurations i.e. FV1-FV2, FV1-FV3, FV2-FV3, and FV1-FV2-FV3. The resulting feature vectors are classified with different classifiers. The dimension of each feature vector, classification accuracies, and training times of each feature vector and classifier are shown in Table 5.

As is evident from Table 5, the highest accuracy of 99.3% is achieved with fused feature vector FV1-FV2 while using the Linear Discriminant classifier. Ensemble Subspace Discriminant classifier takes the highest training time while Linear Discriminant has the lowest training time. However, on average, the best classification accuracy is achieved

when FV1-FV2-FV3 are fused but it also takes the highest training time, as shown in Figure 9.

Table 5: Accuracy with different layers of pre-trained CNNs used as feature extractors on WHU-RS-19.

Classifier	**FV1-FV2 (Dimension = 704 x 3456)**		**FV1-FV3 (Dimension = 704 x 2976)**		**FV2-FV3 (Dimension = 704 x 2592)**		**FV1-FV2-FV3 (Dimension = 704 x 4512)**	
	Acc. (%)	Tr. Time (sec)	Acc. (%)	Tr. Time (sec)	Acc. (%)	Tr. Time (sec)	Acc. (%)	Tr. Time (sec)
Linear Disc.	**99.3**	35	**98.3**	30	**95.7**	17.577	*	*
En. Sub. Disc.	98.7	488	**98.3**	316	**95.7**	235.92	**98.3**	378.35
Linear SVM	96.7	155	97.7	133	93.7	77	98	129.6
Q SVM	97	170	97.7	153	94.4	83	97.7	150.8
Cubic SVM	97	214	97.7	183	94.4	92	97.7	152.6
MG SVM	96.7	240	96.7	180	92	77	96.7	200
Wide NN	97.3	343	95.7	207	**95.7**	160	97	291
Med. NN	96.3	323	97.3	205	93.7	147	97	282
Cosine KNN	94.4	257	95.7	94	90	23	95.3	207
Avg.	**97.0**	247	**97.2**	167	**93.9**	101	**97.2**	224

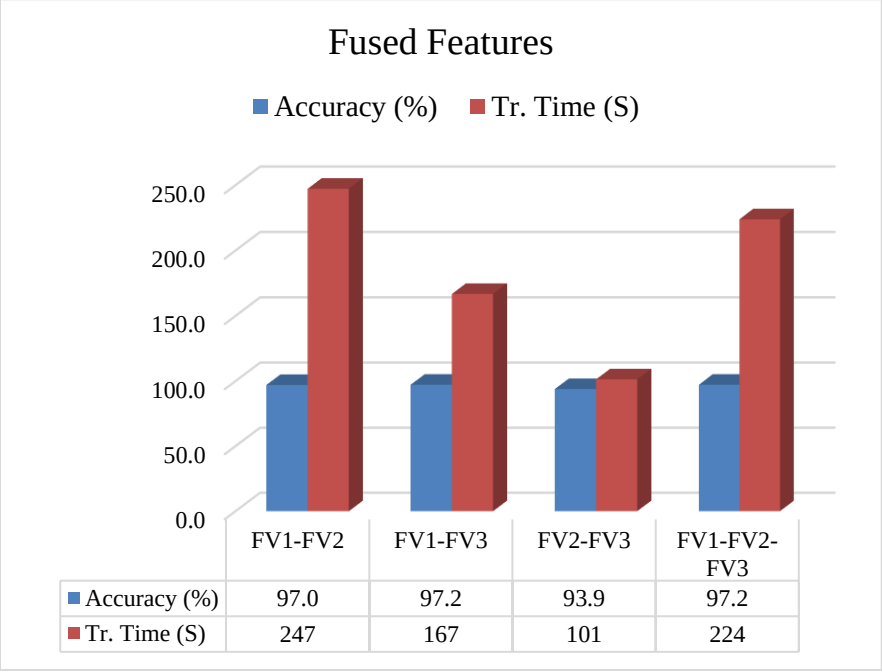


Figure 9: Average classification accuracies comparison when different features are fused (WHU-RS19 dataset)

4.2.3 Results with Features Selection Approach

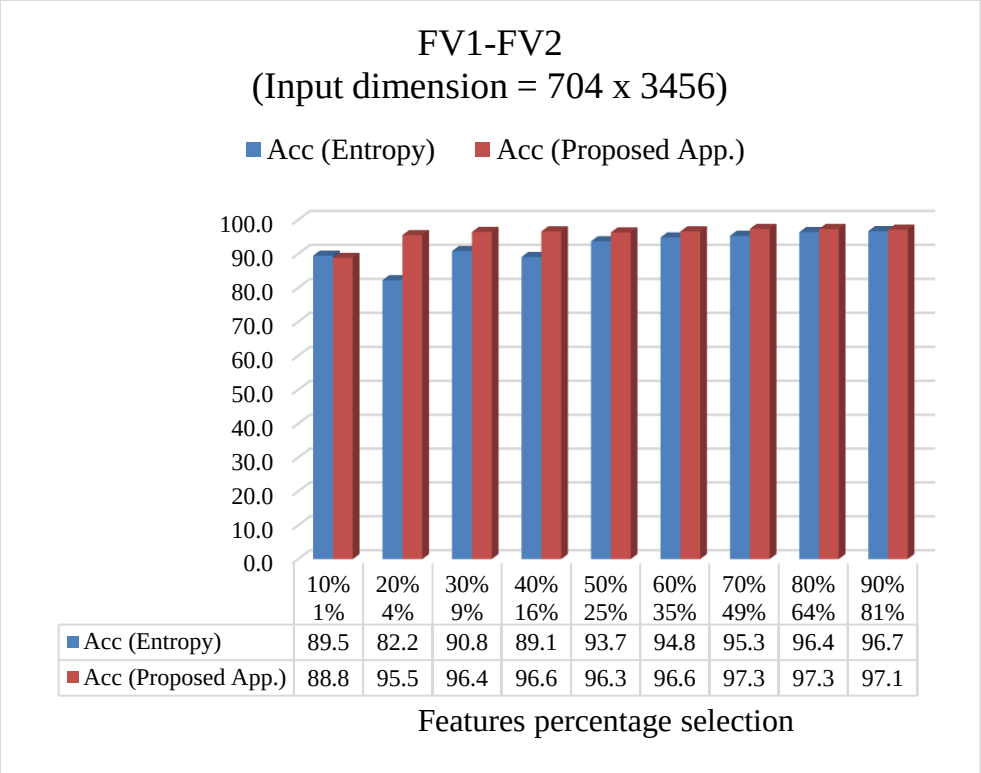
Since with features fusion approach, the number of features increases, resultantly, training time also increases while there is no significant difference in the classification accuracy. Therefore, the feature selection approach is investigated to study the effects of feature selection on classification accuracy and training time. In this work two strategies, Entropy and ReliefF are adopted for feature selection. Only Entropy is used for features selection in the first phase and in the second phase, which is the proposed approach, features are selected with Entropy followed by features selection with the ReliefF algorithm. In this way, optimal features can be selected with better classification accuracy and low computational cost. Results of each combination of fusion i.e. FV1-FV2, FV1-FV3, FV2-FV3, and FV1-FV2-FV3 will be separately discussed using both the strategies.

In the case of features selection with Entropy, features are selected from 10% to 90% with a step size of 10%. With the proposed approach of features selection, the sequence of features percentage selected is 1%, 4%, 9%, 16%, 25%, 26%, 35%, 49%, 64% and 81%. Furthermore, average classification accuracies of all the 9 classifiers used are considered for comparison between the two approaches, i.e. classification with features selection with Entropy and the features selection with the proposed approach (Entropy followed by ReliefF).

FV1-FV2:

Classification accuracies achieved after features are selected in different proportions using Entropy are shown in Table 6 while those with the proposed approach are shown in Table 7. Training times are also given in the tables. Average accuracies of both approaches are compared in Figure 10. The highest accuracy of 97.3% is achieved with the proposed approach with selection of 49% of the input features.

Linear Discriminant classifier achieved 99.3% accuracy in both the cases, i.e. with features selection with Entropy and the proposed approach. LD takes the minimum training time.



	10% 1%	20% 4%	30% 9%	40% 16%	50% 25%	60% 35%	70% 49%	80% 64%	90% 81%
Acc (Entropy)	89.5	82.2	90.8	89.1	93.7	94.8	95.3	96.4	96.7
Acc (Proposed App.)	88.8	95.5	96.4	96.6	96.3	96.6	97.3	97.3	97.1

Figure 10: Average classification accuracies of FV1-FV2 after features selection with Entropy and the proposed approach (WHU-RS19 dataset)

Table 6: Accuracy and Training Time of fused features (FV1-FV2) with selection of features using Entropy on WHU-RS19 dataset.

Input dimension = 704 x 3456	Percent. Selection	Output Dimension		Linear Disc.	En. Sub. Disc	Linear SVM	Q SVM	Cubic SVM	MG SVM	Wide NN	Med. NN	Cosine KNN	Avg.
	10%	**704 x 346**	Acc. (%)	91.7	**93**	89	88.4	88.7	88	91.7	87.7	87	**89.5**
			Tr. Time (sec)	2	35	23	26	26	26	27	26	20	23
	20%	**704 x 691**	Acc. (%)	17.9	**93**	90.4	89.7	89	88.7	94.4	91.4	85.7	**82.2**
			Tr. Time (sec)	4.7	72	34	37	43	45	52	47	27	40
	30%	**704 x 1037**	Acc. (%)	90	93	90.7	90.4	90	89.4	**94.4**	92.7	86.7	**90.8**
			Tr. Time (sec)	8	122	43	50	59	68	83	78	53	63
	40%	**704 x 1382**	Acc. (%)	93.7	72.8	90.7	90	91	88	**95.7**	93	86.7	**89.1**
			Tr. Time (sec)	10.5	148	10	64	78	90	87	71	61	69
	50%	**704 x 1728**	Acc. (%)	95.3	**95.7**	93.4	94	94	92	**95.7**	95	88	**93.7**
			Tr. Time (sec)	14	229	61	77	95	106	145	137	110	108
	60%	**704 x 2074**	Acc. (%)	**97**	96.7	94.4	95	94.7	93.7	96.3	95	90.7	**94.8**
			Tr. Time (sec)	17	110	91	103	125	147	167	159	148	119
	70%	**704 x 2419**	Acc. (%)	**98.3**	98	94.7	95.3	95.7	94	94.7	94.7	92.4	**95.3**
			Tr. Time (sec)	25	123	118	136	150	145	146	121	81	116
	80%	**704 x 2765**	Acc. (%)	**98.7**	**98.7**	96.7	96.7	96.7	94.7	96.3	96.3	92.7	**96.4**
			Tr. Time (sec)	27	141	134	149	169	168	112	94	75	119
	90%	**704 x 3110**	Acc. (%)	**99.3**	98	96.3	96.7	97	96	97.7	95.7	93.4	**96.7**
			Tr. Time (sec)	38	330	150	170	207	206	204	187	170	184

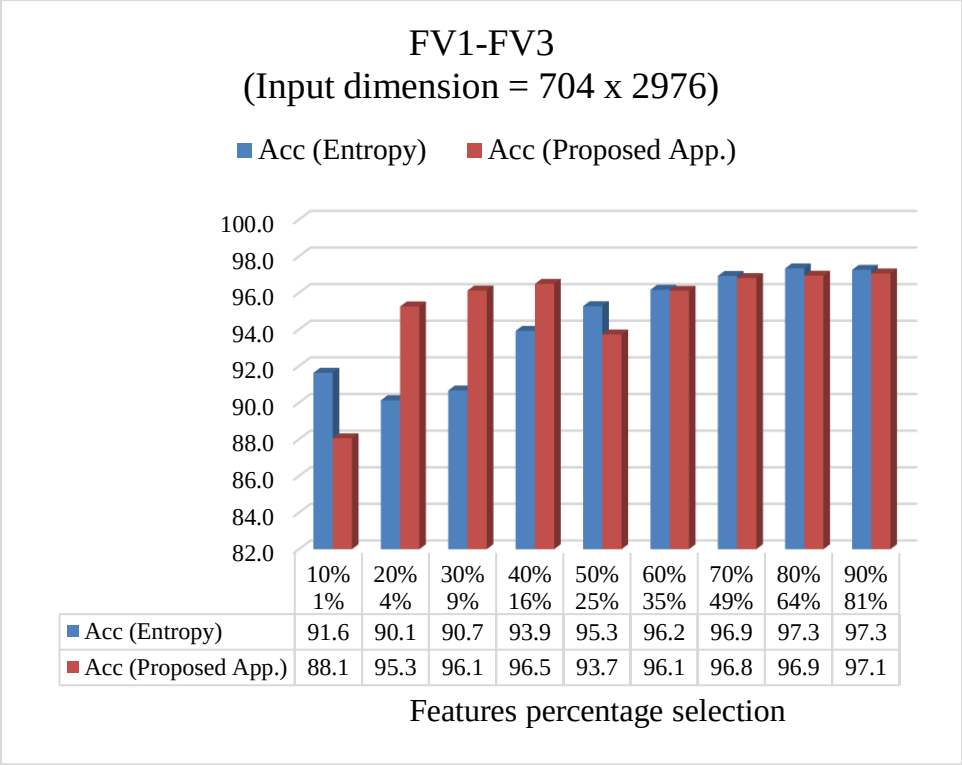
Table 7: Accuracy and Training Time of fused features (FV1-FV2) with selection of features using the proposed approach on WHU-RS19 dataset.

Input dimension = 704 x 3456	Percent. Selection	Output Dimension		Linear Disc.	En. Sub. Disc	Linear SVM	Q SVM	Cubic SVM	MG SVM	Wide NN	Med. NN	Cosine KNN	Avg.
	1%	**704 x 35**	Acc. (%)	91.4	88.7	90	88.7	88.7	**91.7**	87.4	84.7	87.7	**88.8**
			Tr. Time (sec)	0.8	17	15	15	16	16	17	16	4	13
	4%	**704 x 138**	Acc. (%)	**97**	**97**	95.7	96	95.7	96	95	95.7	91.4	**95.5**
			Tr. Time (sec)	0.8	23	18	18	19	18	21	19	9	16
	9%	**704 x 311**	Acc. (%)	**99**	98.3	95.7	96	96	96.3	97	96.3	93.4	**96.4**
			Tr. Time (sec)	2.7	31	22	24	25	26	26	25	14	22
	16%	**704 x 553**	Acc. (%)	94.4	**99**	97.3	97	97.3	97	97	95.7	94.7	**96.6**
			Tr. Time (sec)	3.8	57	27	30	33	37	43	38	33	34
	26%	**704 x 864**	Acc. (%)	92.7	**99**	97	96.3	96.3	97.3	98	95.3	94.7	**96.3**
			Tr. Time (sec)	5.3	71	37	44	49	48	43	37	24	40
	36%	**704 x 1244**	Acc. (%)	96.3	**98**	96.7	97	96.7	96.7	97	96.3	94.7	**96.6**
			Tr. Time (sec)	9	115	51	63	68	67	67	63	31	59
	49%	**704 x 1693**	Acc. (%)	98.7	**99.3**	97	97.3	96.7	97.7	97.3	97.3	94.7	**97.3**
			Tr. Time (sec)	13	90	67	80	97	95	111	104	69	81
	64%	**704 x 2212**	Acc. (%)	**99**	97.9	97	97	97	97.7	97.3	97.3	95.7	**97.3**
			Tr. Time (sec)	19	115	98	111	132	135	126	114	65	102
	81%	**704 x 2799**	Acc. (%)	**99.3**	98.7	96.7	97	97	97.3	96.7	96.3	95	**97.1**
			Tr. Time (sec)	31.5	245	130	150	153	150	158	143	59	136

FV1-FV3

Table 8 and Table 9 show classification accuracies with Entropy and the proposed approach respectively. When the averaged accuracies of both approaches are compared, the proposed approach shows better results with 96.5% with 9% feature selection. However, 97.3% accuracy is achieved with Entropy when 80% of features are selected. The averaged accuracies are shown in Figure 11.

Ensemble subspace Discriminant classifier shows better results but it also takes the highest training time. In comparison, the LD classifier also achieved better accuracies with less training time. These results are shown in Table 8 and Table 9.



	10% 1%	20% 4%	30% 9%	40% 16%	50% 25%	60% 35%	70% 49%	80% 64%	90% 81%
Acc (Entropy)	91.6	90.1	90.7	93.9	95.3	96.2	96.9	97.3	97.3
Acc (Proposed App.)	88.1	95.3	96.1	96.5	93.7	96.1	96.8	96.9	97.1

Figure 11: Average classification accuracies of FV1-FV2 after features selection with Entropy and the proposed approach (WHU-RS19 dataset)

Table 8: Accuracy and Training Time of fused features (FV1-FV3) with the selection of features using Entropy on WHU-RS19 dataset.

Input dimension = 704 x 2976	Percent. Selection	Output Dimension		Linear Disc.	En. Sub. Disc	Linear SVM	Q SVM	Cubic SVM	MG SVM	Wide NN	Med. NN	Cosine KNN	Avg.
	10%	**704 x 298**	Acc. (%)	91.4	**98**	91.4	91.4	91.7	90.7	91.7	89.7	88.7	91.6
			Tr. Time (sec)	2	33	20	22	24	22	30	28	19	22
	20%	**704 x 595**	Acc. (%)	77	93	92	91.7	92	90.7	92.4	**94**	88.4	90.1
			Tr. Time (sec)	5	62	29	35	40	41	44	42	33	37
	30%	**704 x 893**	Acc. (%)	79.4	92.4	93.7	93.4	93.7	90.7	**94**	91	87.7	90.7
			Tr. Time (sec)	6	104	38	47	54	54	75	67	45	54
	40%	**704 x 1191**	Acc. (%)	94.4	**95.3**	93.7	94	94.7	92.4	94.4	95	91.4	93.9
			Tr. Time (sec)	9	135	47	59	65	73	86	81	50	67
	50%	**704 x 1488**	Acc. (%)	96.3	95.7	95.3	95.3	95.7	92.4	**96.7**	95.3	94.7	95.3
			Tr. Time (sec)	10	181	55	69	78	91	113	104	84	87
	60%	**704 x 1786**	Acc. (%)	97.3	**97.7**	96.7	97	97	93.4	96.7	96	93.7	96.2
			Tr. Time (sec)	13	196	71	86	95	94	115	102	58	92
	70%	**704 x 2084**	Acc. (%)	98	**98.7**	98	97.7	97.3	94	97.3	96.3	95	96.9
			Tr. Time (sec)	17	198	93	112	124	128	113	106	76	107
	80%	**704 x 2381**	Acc. (%)	**98.3**	**98.3**	98	98	97.7	96	96.7	97.7	95.3	97.3
			Tr. Time (sec)	22	260	108	126	149	150	168	160	101	138
	90%	**704 x 2678**	Acc. (%)	**98.7**	98.3	98	98	98	96.3	97	95.7	95.3	97.3
			Tr. Time (sec)	29	245	132	151	153	149	164	159	64	138

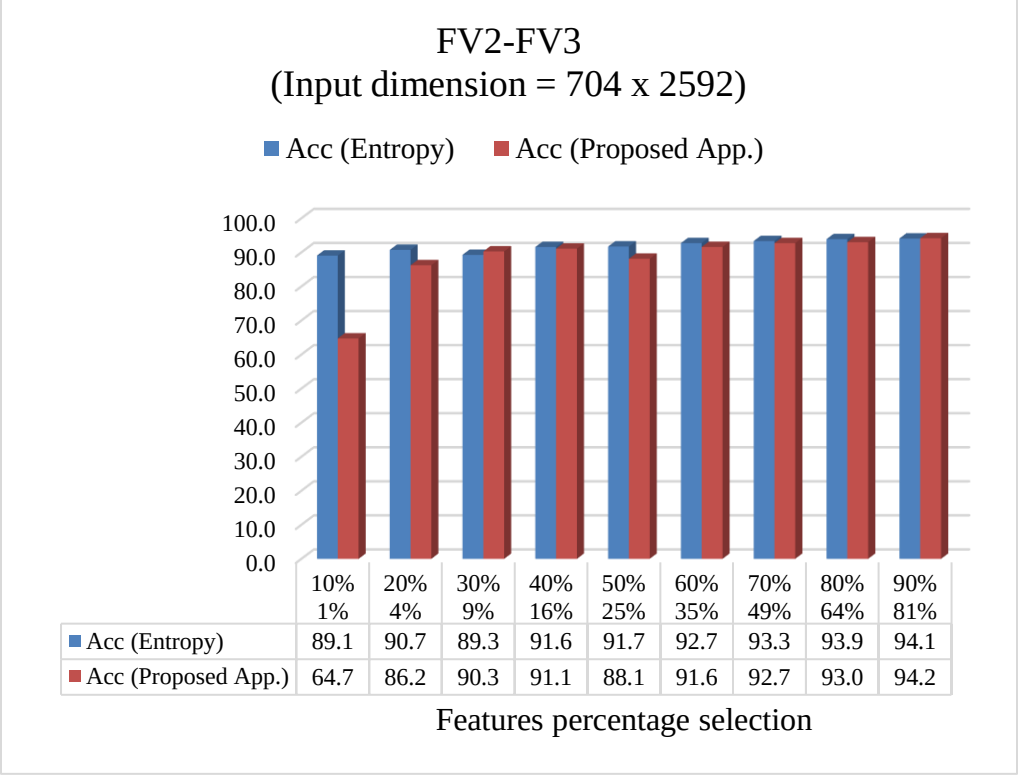
Table 9: Accuracy and Training Time of fused features (FV1-FV3) with the selection of features using the proposed approach on WHU-RS19 dataset

Input dimension = 704 x 2976	Percent. Selection	Output Dimension		Linear Disc.	En. Sub. Disc	Linear SVM	Q SVM	Cubic SVM	MG SVM	Wide NN	Med. NN	Cosine KNN	Avg.
	1%	**704 x 30**	Acc. (%)	**90**	87.4	88	89	88	90	87.4	87	85.7	**88.1**
			Tr. Time (sec)	1.2	16	14	14	15	15	16	16	3.5	12
	4%	**704 x 119**	Acc. (%)	**97**	96	95.3	95	94.7	96.3	96.3	94	92.7	**95.3**
			Tr. Time (sec)	1.2	19	14	16	18	16	18	16	6	14
	9%	**704 x 268**	Acc. (%)	96.3	**98.3**	95.7	95.7	95.7	97	96	96.7	93.7	**96.1**
			Tr. Time (sec)	1.7	29	21	22	23	24	30	28	15	22
	16%	**704 x 476**	Acc. (%)	96.7	**99**	96.3	96	96.3	96.7	97	96	94.4	**96.5**
			Tr. Time (sec)	3.4	47	26	30	32	33	36	34	23	29
	26%	**704 x 744**	Acc. (%)	75.4	**98**	96.7	96	96	96.7	96.3	94.4	94	**93.7**
			Tr. Time (sec)	8	87	41	48	53	61	65	62	50	53
	36%	**704 x 1072**	Acc. (%)	95.3	**98.3**	97	96.3	96.3	97	95.7	94.4	94.7	**96.1**
			Tr. Time (sec)	7	117	43	50	56	62	75	66	38	57
	49%	**704 x 1458**	Acc. (%)	97	**98.3**	97.7	97.3	97.3	97.3	96	95.3	95	**96.8**
			Tr. Time (sec)	13	197	59	73	84	99	90	96	60	86
	64%	**704 x 1905**	Acc. (%)	**98.3**	97.7	97	97	97	96.7	96	97.7	95	**96.9**
			Tr. Time (sec)	17	200	86	98	106	107	119	116	64	101
	81%	**704 x 2410**	Acc. (%)	**98.3**	**98.3**	97.3	97.3	97	96.7	97.3	96	95.3	**97.1**
			Tr. Time (sec)	21	320	105	123	147	155	214	203	144	159

FV2-FV3:

As shown in Figure 12, classification results of features selected with the proposed approach performed better as compared to features selection with Entropy. The highest average accuracy of 94.2% is achieved with the proposed method when 81% of features are selected. However, the results do not improve with lower percentages of feature selection.

Detailed classification accuracies with all the classifiers and with training times for both approaches are shown in Table 10 and Table 11. 96.3% accuracy is achieved with the Ensemble subspace Discriminant classifier. LD also shows better results and with lower training time.



	10% 1%	20% 4%	30% 9%	40% 16%	50% 25%	60% 35%	70% 49%	80% 64%	90% 81%
Acc (Entropy)	89.1	90.7	89.3	91.6	91.7	92.7	93.3	93.9	94.1
Acc (Proposed App.)	64.7	86.2	90.3	91.1	88.1	91.6	92.7	93.0	94.2

Figure 12: Average classification accuracies of FV2-FV2 after features selection with Entropy and the proposed approach (WHU-RS19 dataset)

Table 10: Accuracy and Training Time of fused features (FV2-FV3) with the selection of features using Entropy on WHU-RS19 dataset.

Input dimension = 704 x 2592	Percent. Selection	Output Dimension		Linear Disc.	En. Sub. Disc	Linear SVM	Q SVM	Cubic SVM	MG SVM	Wide NN	Med. NN	Cosine KNN	Avg.
	10%	**704 x 260**	Acc. (%)	91.4	**92**	88	89	88.4	87.4	91.7	88	85.7	**89.1**
			Tr. Time (sec)	1.5	25	22	23	23	22	24	21	11	19.2
	20%	**704 x 518**	Acc. (%)	90.4	**92.7**	90.4	91	91	90.4	92.4	92.4	86	**90.7**
			Tr. Time (sec)	3.5	52	27	30	32	36	41	40	30	32
	30%	**704 x 778**	Acc. (%)	79	**93.7**	90.7	90	89.7	89	93.7	91.7	86	**89.3**
			Tr. Time (sec)	5	74	38	41	31	52	49	44	33	41
	40%	**704 x 1037**	Acc. (%)	92	94.4	91	91.4	89.7	90.4	**94.7**	93	87.7	**91.6**
			Tr. Time (sec)	7	94	43	50	57	63	56	52	34	51
	50%	**704 x 1296**	Acc. (%)	93	**94**	91.4	91.7	91.7	89	93.7	93	88	**91.7**
			Tr. Time (sec)	8	172	49	58	66	77	114	101	76	80
	60%	**704 x 1555**	Acc. (%)	94.4	94	91.4	92.4	92.7	90.4	95	**94.7**	89.4	**92.7**
			Tr. Time (sec)	11	183	61	72	83	94	117	112	78	90
	70%	**704 x 1814**	Acc. (%)	94.7	95	92.7	93.4	93.4	90.7	**95.3**	95	89.7	**93.3**
			Tr. Time (sec)	14	245	71	85	97	111	158	145	116	116
	80%	**704 x 2074**	Acc. (%)	95.3	95.3	94	93.7	93.7	92	95	**95.3**	90.7	**93.9**
			Tr. Time (sec)	17	272	90	101	121	136	179	163	133	135
	90%	**704 x 2333**	Acc. (%)	95.7	**96**	94	94.7	94.4	92	95	94.4	90.4	**94.1**
			Tr. Time (sec)	20	322	94	115	137	155	216	203	162	158

Table 11: Accuracy and Training Time of fused features (FV2-FV3) with selection of features using Entropy and ReliefF on WHU-RS19 dataset.

Input dimension = 704 x 2592	Percent. Selection	Output Dimension		Linear Disc.	En. Sub. Disc	Linear SVM	Q SVM	Cubic SVM	MG SVM	Wide NN	Med. NN	Cosine KNN	Avg.
	1%	**704 x 26**	Acc. (%)	**66.8**	63.5	66.4	69.4	67.1	65.8	64.5	61.5	57.5	**64.7**
			Tr. Time (sec)	0.4	18	16	17	15	19	16	17	3	13
	4%	**704 x 104**	Acc. (%)	88.7	88	85.7	87.4	86	85	**89.4**	82.1	83.7	**86.2**
			Tr. Time (sec)	0.87	18	15	15	17	15	17	16	6.6	13
	9%	**704 x 234**	Acc. (%)	92.7	**93.4**	90.7	91.7	91.4	86	92.7	88.7	85.4	**90.3**
			Tr. Time (sec)	1.5	28	20	21	22	24	29	26	17	21
	16%	**704 x 415**	Acc. (%)	90	**95.7**	92	91	90.7	87.4	93	91.7	88.4	**91.1**
			Tr. Time (sec)	2.86	49	24	27	29	32	38	34	28	29
	26%	**704 x 648**	Acc. (%)	62.1	**94.7**	90.7	91.4	91.4	88.7	94.7	91.7	87.7	**88.1**
			Tr. Time (sec)	5	73	34	37	41	44	52	47	39	41
	36%	**704 x 933**	Acc. (%)	90.4	**94.4**	92	92.4	91.7	89.4	94	92.4	87.7	**91.6**
			Tr. Time (sec)	7	107	40	48	51	54	69	61	48	54
	49%	**704 x 1270**	Acc. (%)	94	**94.7**	92	93.4	93	89.7	96	93	88.7	**92.7**
			Tr. Time (sec)	9	150	58	63	70	68	72	65	55	68
	64%	**704 x 1659**	Acc. (%)	**95.3**	**95.3**	93.7	93.4	92.7	90	95	93	88.7	**93.0**
			Tr. Time (sec)	13	181	65	79	91	98	108	103	71	90
	81%	**704 x 2100**	Acc. (%)	95.7	**96.3**	94.4	93.7	94	91	95.7	95.7	91	**94.2**
			Tr. Time (sec)	19	231	95	109	123	135	144	129	106	121

4.3 Results with UC Merced Dataset

In this section simulation results with the UC Merced dataset are discussed. It should be noted that class labels are represented as numeric labels (1 through 21 for all the 21 classes).

4.3.1 Results with single layers features

Table 12 shows the classification accuracies of individual layers extracted from pre-trained CNN (shown in Table 2). The classification performance of the individual layers extracted from pre-trained CNNs is above 90%. FV1 performs the best with an accuracy of 96% followed by FV2 with an accuracy of 94.8% and then FV3 with an accuracy of 91.3%. These accuracies are achieved by the Ensemble subspace classifier. The average accuracies are shown in Figure 13.

Table 12: Accuracy with different layers of pre-trained CNNs used as feature extractors on UC Merced dataset.

Classifier	**FV1 (Dimension = 1470** x **3456)**		**FV2 (Dimension = 1470** x **1536)**		**FV3 (Dimension = 1470** x **1056)**	
	Acc. (%)	Tr. Time (sec)	Acc. (%)	Tr. Time (sec)	Acc. (%)	Tr. Time (sec)
Linear Disc.	*	*	94	24	87.8	10
En. Sub. Disc.	**96**	370	**94.8**	181	**91.3**	108
Linear SVM	94.1	96	90.5	94	87.5	85
Q SVM	94.4	122	91.4	114	88.9	77
Cubic SVM	94.3	123	91.1	116	87.9	82
MG SVM	89.8	186	84.3	111	77.3	81
Wide NN	95.6	236	94	82	90.3	59
Medium NN	94.9	226	90.5	62	88.7	54
Cosine KNN	90.6	158	86	71	86	49
Avg.	**93.7**	190	**90.7**	95	**87.3**	67

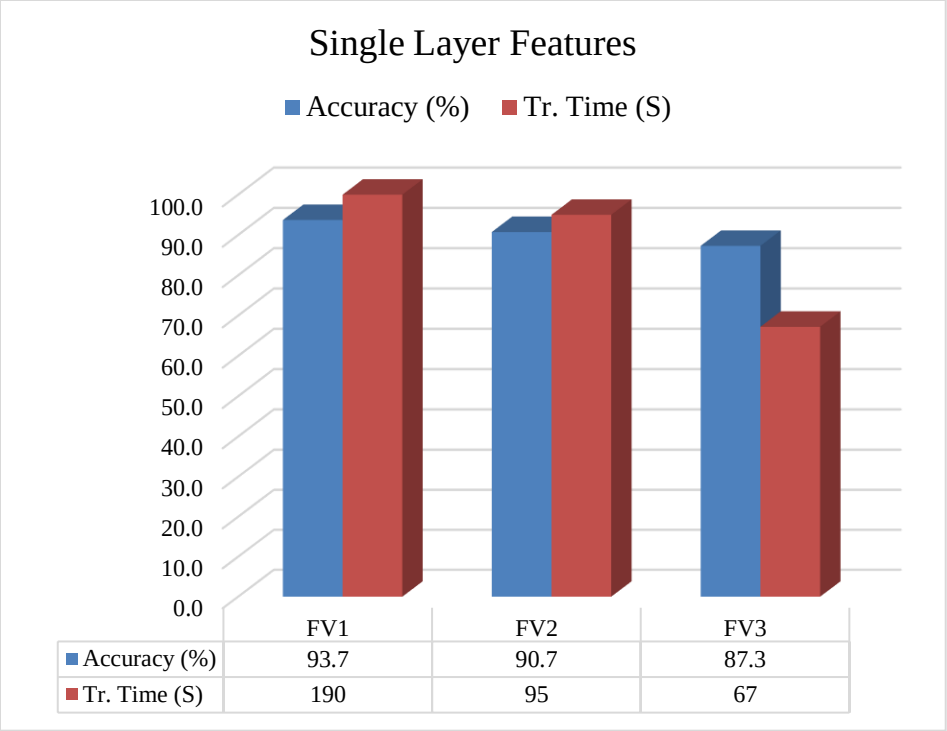


Figure 13: Average classification accuracies with single-layer features (UC Merced)

4.3.2 Results with Features Fusion Technique

In this approach, the features which are extracted from different pre-trained CNNs, i.e. FV1, FV2, and FV2 are concatenated before feature selection. The features are fused in four different configurations i.e. FV1-FV2, FV1-FV3, and FV2-FV3. The resulting feature vectors are classified with different classifiers. The dimension of each feature vector, classification accuracies, and training times of each feature vector and classifier are shown in Table 13. The results are graphically shown in Figure 14.

Table 13: Accuracy with different layers of pre-trained CNNs used as feature extractors on UC Merced dataset.

Classifier	FV1-FV2 (Dimension = 704 x 3456)		FV1-FV3 (Dimension = 704 x 2976)		FV2-FV3 (Dimension = 704 x 2592)		FV1-FV2-FV3 (Dimension = 704 x 4512)	
	Acc. (%)	Tr. Time (sec)	Acc. (%)	Tr. Time (sec)	Acc. (%)	Tr. Time (sec)	Acc. (%)	Tr. Time (sec)
Linear Disc.	*	*	*	*	84.8	37	*	*
En. Sub. Disc.	96	516	95.9	397	93.7	479	96.8	593
Linear SVM	95.1	190	94.6	169	92.1	147	94.9	265
Q SVM	95.4	234	95.2	201	92.5	177	95.2	303
Cubic SVM	95.2	237	94.9	205	92.4	188	95.6	312
MG SVM	91.3	271	88.9	192	83.5	174	88.1	299
Wide NN	**97.3**	200	97.1	141	**95.6**	234	**97.1**	144
Med. NN	94.8	155	95.9	108	93.2	215	96.7	106
Cosine KNN	91.9	157	92.5	81	89.8	145	92.2	137
Avg.	**94.6**	245	**94.4**	187	**90.8**	200	**94.6**	270

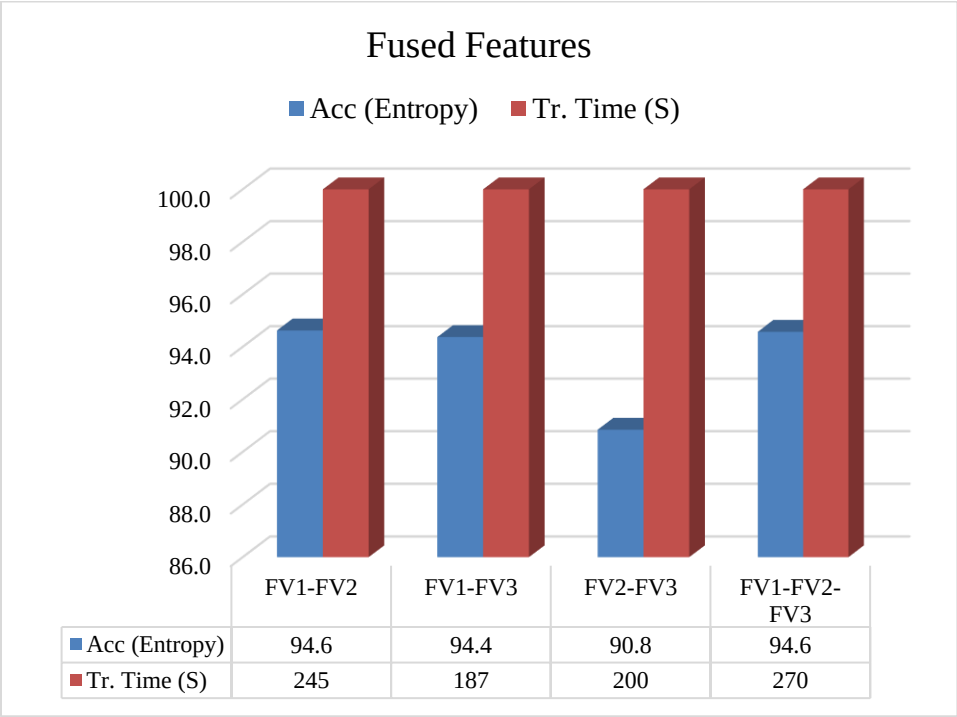


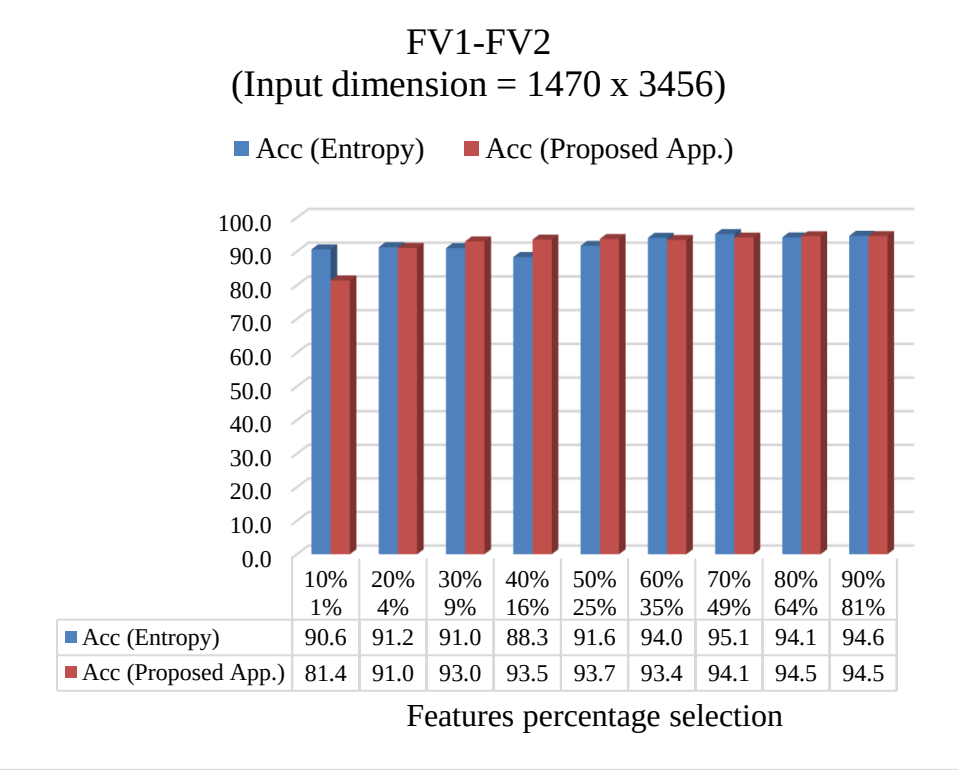
Figure 14: Classification accuracies comparison when different features are fused (UC Merced)

4.3.3 Results with Features Selection Approach

In this section results of each combination of features fusion i.e. FV1-FV2, FV1-FV3 and FV2-FV3 are separately discussed using both the strategies (feature reduction).

FV1-FV2:

Figure 15 shows that the classification accuracies achieved with the proposed approach are better than the features selected with Entropy are classified. The highest accuracy of 94.5% is achieved with the proposed approach and with the selection of 64% of the features. The proposed approach also performs better compared to features selection with Entropy, for lower percentages of features selection. When looking at classifier level accuracies, En. Sub. Disc. Classifier achieved the highest accuracy of 97.7% with 70% features selection through Entropy. 94.5% accuracy is achieved with Wide NN with 9% features selection through the proposed approach. These are shown in Table 14 and Table 15.



	10% 1%	20% 4%	30% 9%	40% 16%	50% 25%	60% 35%	70% 49%	80% 64%	90% 81%
Acc (Entropy)	90.6	91.2	91.0	88.3	91.6	94.0	95.1	94.1	94.6
Acc (Proposed App.)	81.4	91.0	93.0	93.5	93.7	93.4	94.1	94.5	94.5

Figure 15: Average classification accuracies of FV1-FV2 after features selection with Entropy and the proposed approach (UC Merced dataset)

Table 14: Accuracy and Training Time of fused features (FV1-FV2) with the selection of features using Entropy on UC Merced dataset.

Input dimension = 1470 x 3456	Percent. Selection	Output Dimension		Linear Disc.	En. Sub. Disc	Linear SVM	QSVM	Cubic SVM	MSVM	Wide NN	Med. NN	Cosine KNN	Avg.
	10%	**1470 x 346**	Acc. (%)	**92.2**	**92.2**	91.6	91.6	91.1	88.9	91.3	90	86.3	**90.6**
			Tr. Time (sec)	4	64	45	52	56	57	54	51	35	46
	20%	**1470 x 691**	Acc. (%)	93.3	**94.4**	91.1	91.7	91.3	89.8	93	90.8	85.7	**91.2**
			Tr. Time (sec)	11	103	66	83	87	88	61	55	38	66
	30%	**1470 x 1037**	Acc. (%)	89.8	**96**	91.4	92.2	91.6	90.6	92.9	88.4	85.7	**91.0**
			Tr. Time (sec)	32	171	122	142	147	148	73	58	48	105
	40%	**1470 x 1382**	Acc. (%)	63.5	**95.7**	91.4	91.6	91.4	89.7	93.5	91.6	86.6	**88.3**
			Tr. Time (sec)	42	229	147	167	172	168	83	54	63	125
	50%	**1470 x 1728**	Acc. (%)	86.2	**96.2**	92.7	93.2	93.3	90.8	94.1	91.6	86.7	**91.6**
			Tr. Time (sec)	52	333	193	220	223	217	76	61	70	161
	60%	**1470 x 2074**	Acc. (%)	93.5	**95.9**	94.6	94.9	95.1	93.7	95	93.7	89.7	**94.0**
			Tr. Time (sec)	57	526	260	305	314	306	136	106	96	234
	70%	**1470 x 2419**	Acc. (%)	95.1	**97.7**	95.2	96	95.6	93.8	96.8	94.6	91.3	**95.1**
			Tr. Time (sec)	71	601	263	301	307	296	108	89	96	237
	80%	**1470 x 2765**	Acc. (%)	95.7	95.1	94.4	94.9	94.8	88.7	**96.7**	95.1	91.9	**94.1**
			Tr. Time (sec)	76	714	281	323	336	334	143	107	124	271
	90%	**1470 x 3110**	Acc. (%)	96.7	95.7	95.1	95.4	95.2	90.5	**97.3**	94.8	90.8	**94.6**
			Tr. Time (sec)	91	795	305	344	359	357	156	125	131	296

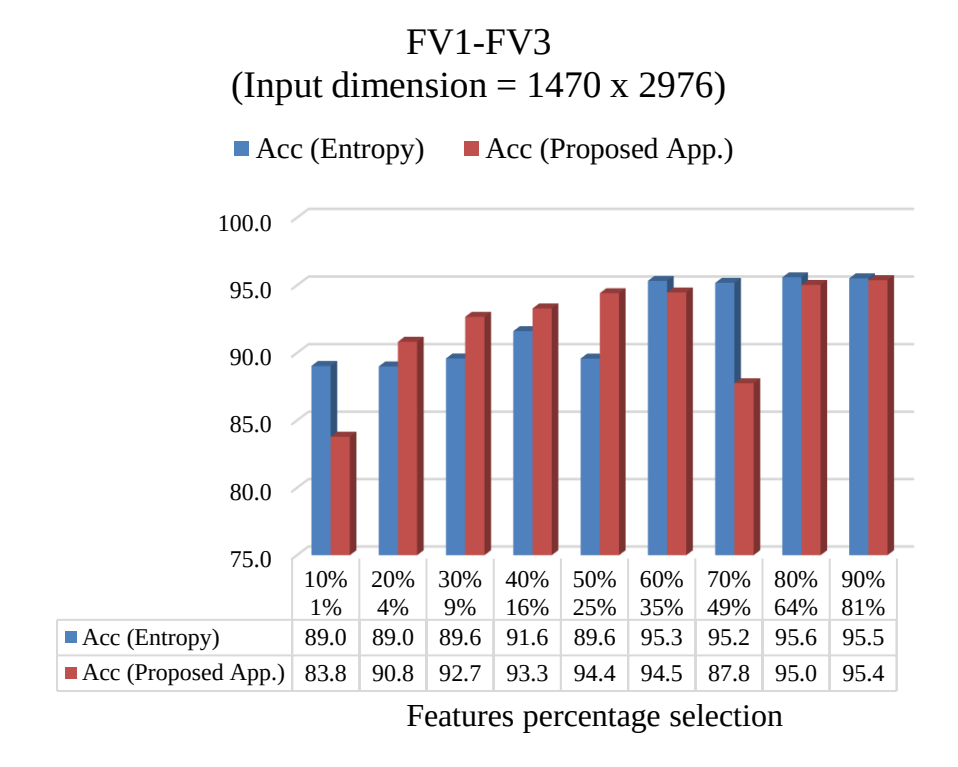
Table 15: Accuracy and Training Time of fused features (FV1-FV2) with the selection of features using the proposed approach on UC Merced dataset

Input dimension = 1470 x 3456	Percent. Selection	Output Dimension		Linear Disc.	En. Sub. Disc	Linear SVM	QSVM	Cubic SVM	MSVM	Wide NN	Med. NN	Cosine KNN	Avg.
	1%	**1470 x 35**	Acc. (%)	78.1	75.2	81.6	85.2	84.4	83.7	84.4	79.8	80.2	**81.4**
			Tr. Time (sec)	1.2	22	18	19	22	21	25	23	4.2	17
	4%	**1470 x 138**	Acc. (%)	92.4	**91.3**	92.2	92.9	92.9	90.3	91.3	88.4	87.5	**91.0**
			Tr. Time (sec)	0.9	33	23	24	26	27	37	35	17	25
	9%	**1470 x 311**	Acc. (%)	94.4	93.2	93.5	93.7	93.7	91.1	**95.4**	93.3	88.3	**93.0**
			Tr. Time (sec)	2.2	54	31	36	38	45	49	43	26	36
	16%	**1470 x 553**	Acc. (%)	95.2	94.6	94.1	95.1	94.1	91.1	94.4	93.8	88.9	**93.5**
			Tr. Time (sec)	7.8	65	49	58	59	60	37	34	25	44
	26%	**1470 x 864**	Acc. (%)	94.6	**95.7**	94.4	94.8	94.6	89.5	95.6	93.7	90.2	**93.7**
			Tr. Time (sec)	17.7	110	73	87	89	90	51	43	38	67
	36%	**1470 x 1244**	Acc. (%)	87.6	**96.7**	94.8	94.9	94.6	90	95.7	94.3	91.7	**93.4**
			Tr. Time (sec)	37	193	118	139	148	147	75	53	62	108
	49%	**1470 x 1693**	Acc. (%)	87.1	**97.5**	95.1	**95.7**	**95.7**	91.1	96.8	95.9	92.1	**94.1**
			Tr. Time (sec)	55	330	184	214	217	214	98	82	64	162
	64%	**1470 x 2212**	Acc. (%)	94.1	97.5	94.6	95.1	95.1	90.6	**95.6**	95.7	92.2	**94.5**
			Tr. Time (sec)	62	510	226	252	261	258	123	85	98	208
	81%	**1470 x 2799**	Acc. (%)	**96.7**	94.6	95.2	95.4	95.2	90.2	96.3	95.6	91.6	**94.5**
			Tr. Time (sec)	84	726	299	333	341	337	133	108	121	276

FV1-FV3:

When FV1 and FV2 are fused, the proposed approach shows better results at lower percentages of features selection as shown in Figure 16**.** The highest average accuracy of 95.6% is achieved with the selection of 80% of the features through Entropy.

Table 16 and Table 17 show accuracies and training times of all the classifiers with both approaches, respectively. En. Sub. Disc. Classifier achieved the highest accuracy of 97.3% with the proposed approach and selection of 49% features. This classifier takes the maximum time training time among the mentioned classifiers.



	10% 1%	20% 4%	30% 9%	40% 16%	50% 25%	60% 35%	70% 49%	80% 64%	90% 81%
Acc (Entropy)	89.0	89.0	89.6	91.6	89.6	95.3	95.2	95.6	95.5
Acc (Proposed App.)	83.8	90.8	92.7	93.3	94.4	94.5	87.8	95.0	95.4

Figure 16: Average classification accuracies of FV1-FV3 after features selection with Entropy and the proposed approach (UC Merced dataset)

Table 16: Accuracy and Training Time of fused features (FV1-FV3) with the selection of features using Entropy on UC Merced dataset.

Input dimension = 1470 x 2976	Percent. Selection	Output Dimension		Linear Disc.	En. Sub. Disc	Linear SVM	QSVM	Cubic SVM	MSVM	Wide NN	Med. NN	Cosine KNN	Avg.
	10%	**1470 x 298**	Acc. (%)	**91.1**	90.2	90.2	90.2	90.3	86.7	88.3	87.3	87	**89.0**
			Tr. Time (sec)	3	65	34	45	42	44	59	55	33	42
	20%	**1470 x 595**	Acc. (%)	90.8	90.3	88.9	90.8	**91.1**	86	90.2	87.5	85.4	**89.0**
			Tr. Time (sec)	8.6	77	60	79	76	76	37	32	25	52
	30%	**1470 x 893**	Acc. (%)	89.4	**92.7**	89.8	90.8	91.4	86.2	91.4	88.6	86	**89.6**
			Tr. Time (sec)	23	120	102	109	112	110	50	35	36	77
	40%	**1470 x 1191**	Acc. (%)	84.4	**94.4**	93.3	94	93.8	89.4	94.1	93.8	87.3	**91.6**
			Tr. Time (sec)	33	201	132	150	158	163	85	73	61	117
	50%	**1470 x 1488**	Acc. (%)	48.7	95.9	95.4	95.7	95.7	92.4	**96.2**	95.1	91.1	**89.6**
			Tr. Time (sec)	45	251	166	184	190	182	78	58	60	135
	60%	**1470 x 1786**	Acc. (%)	97.3	96.5	95.7	96	95.9	92.7	95.7	95.7	92.4	**95.3**
			Tr. Time (sec)	52	372	199	227	237	228	107	74	78	175
	70%	**1470 x 2083**	Acc. (%)	93.8	95.9	96	96.5	96.5	93.3	96.3	94.8	93.5	**95.2**
			Tr. Time (sec)	56	481	228	263	276	268	126	109	94	211
	80%	**1470 x 2381**	Acc. (%)	95.4	96.2	96.5	96.5	**96.7**	94.6	96.3	95.4	92.7	**95.6**
			Tr. Time (sec)	50	732	264	327	353	369	283	233	271	320
	90%	**1470 x 2678**	Acc. (%)	95.6	95.7	96.2	**96.8**	96.7	95.4	95.7	95.1	92.4	**95.5**
			Tr. Time (sec)	77	723	303	346	355	346	124	105	112	277

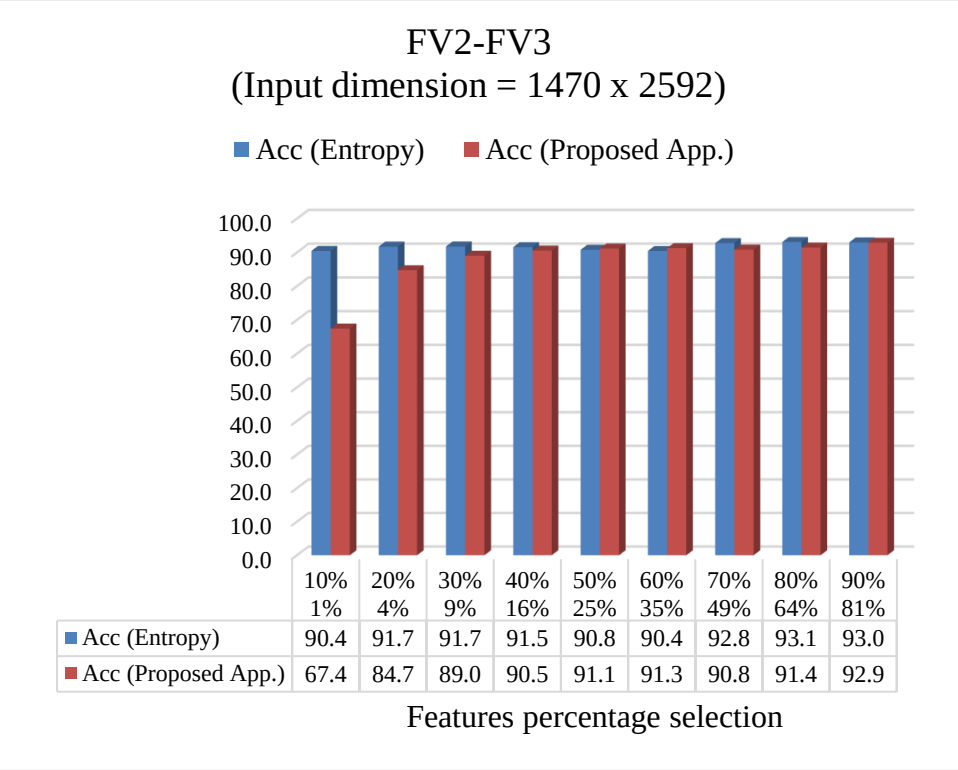
Table 17: Accuracy and Training Time of fused features (FV1-FV3) with the selection of features using the proposed approach on UC Merced dataset

Input dimension = 1470 x 2976	**Percent. Selection**	**Output Dimension**		**Linear Disc.**	**En. Sub. Disc**	**Linear SVM**	**QSVM**	**Cubic SVM**	**MSVM**	**Wide NN**	**Med. NN**	**Cosine KNN**	**Avg.**
	1%	**1470 x 30**	Acc. (%)	81.6	80.3	84.3	87.5	86.3	86.8	84.6	81.7	81	**83.8**
			Tr. Time (sec)	0.8	23	20	20	21	22	24	21	3.6	17.3
	4%	**1470 x 119**	Acc. (%)	91.4	90.2	91.3	92.2	**93**	92.7	92.2	89.5	84.9	**90.8**
			Tr. Time (sec)	1.7	31	25	28	28	29	30	28	13	23.7
	9%	**1470 x 268**	Acc. (%)	93.5	**93.2**	93.5	**94.6**	94.3	94.4	93.3	90.2	87	**92.7**
			Tr. Time (sec)	1.9	39	32	37	40	38	38	36	20	31.3
	16%	**1470 x 476**	Acc. (%)	94.1	94.6	93	**94.9**	94.6	94.1	94.4	92.1	87.8	**93.3**
			Tr. Time (sec)	5	75	48	61	65	67	54	50	26	50.1
	26%	**1470 x 744**	Acc. (%)	95.6	95.7	94.6	95.2	95.2	94.9	**95.9**	94	88.6	**94.4**
			Tr. Time (sec)	13	114	68	85	92	95	77	69	45	73.1
	36%	**1470 x 1072**	Acc. (%)	93.8	**96.7**	94.8	95.2	95.2	95.6	94.4	94	90.5	**94.5**
			Tr. Time (sec)	33	197	110	134	138	139	151	145	56	122.6
	49%	**1470 x 1458**	Acc. (%)	26.5	**97.3**	96	96.2	96	95.9	95.7	94.6	91.6	**87.8**
			Tr. Time (sec)	31	437	148	185	210	213	268	247	205	216.0
	64%	**1470 x 1905**	Acc. (%)	91.9	**96.3**	95.6	95.7	96.2	95.1	96.3	95.7	92.4	**95.0**
			Tr. Time (sec)	46	433	208	236	248	260	132	95	109	196.3
	81%	**1470 x 2410**	Acc. (%)	94.6	96.3	95.6	96	95.9	95.2	**97**	95.1	92.7	**95.4**
			Tr. Time (sec)	74	614	275	311	317	311	142	98	106	249.8

FV2-FV3:

When FV2 and FV3 are fused, classification accuracies of features selected with Entropy show better results as is clear from Figure 17. The highest average accuracy of 93.1% is achieved with 80% features selection through Entropy.

Table 18 and Table 19 show the classification accuracies of all the classifiers. En. Sub. Disc. Classifier achieved 96.5% accuracy with 49% features selection through the proposed features section method.



	10% 1%	20% 4%	30% 9%	40% 16%	50% 25%	60% 35%	70% 49%	80% 64%	90% 81%
Acc (Entropy)	90.4	91.7	91.7	91.5	90.8	90.4	92.8	93.1	93.0
Acc (Proposed App.)	67.4	84.7	89.0	90.5	91.1	91.3	90.8	91.4	92.9

Figure 17: Average classification accuracies of FV1-FV3 after features selection with Entropy and the proposed approach (UC Merced dataset)

Table 18: Accuracy and Training Time of fused features (FV2-FV3) with the selection of features using Entropy on UC Merced dataset

Input dimension = 1470 x 2976	Percent. Selection	Output Dimension		Linear Disc.	En. Sub. Disc	Linear SVM	QSVM	Cubic SVM	MSVM	Wide NN	Med. NN	Cosine KNN	Avg.
	10%	**1470 x 259**	Acc. (%)	93.2	91.3	91	91.4	91.3	90	91	89.8	84.9	**90.4**
			Tr. Time (sec)	2	54	33	37	41	46	55	51	22	38
	20%	**1470 x 518**	Acc. (%)	93.7	**94.3**	91.4	92.5	92.1	91.3	93.8	90	86.2	**91.7**
			Tr. Time (sec)	7	104	51	63	65	67	82	77	31	61
	30%	**1470 x 778**	Acc. (%)	92.7	**94.8**	92.2	93	92.4	91	92.9	90	86.7	**91.7**
			Tr. Time (sec)	13	112	78	94	101	100	51	41	32	69
	40%	**1470 x 1037**	Acc. (%)	88.7	**96**	92.2	92.5	92.1	90.5	93	90.2	88.7	**91.5**
			Tr. Time (sec)	25	172	122	142	152	151	71	58	52	105
	50%	**1470 x 1296**	Acc. (%)	76.7	**96**	92.5	93	93	91.4	94.8	91.9	87.6	**90.8**
			Tr. Time (sec)	44	216	139	159	168	164	72	57	66	121
	60%	**1470 x 1555**	Acc. (%)	74.6	**95.7**	92.5	93.3	93.5	90.8	93.5	92.2	87.6	**90.4**
			Tr. Time (sec)	60	358	180	208	214	206	85	48	67	158
	70%	**1470 x 1814**	Acc. (%)	87.6	**96**	92.7	92.7	93.8	96.1	94.6	92.1	89.2	**92.8**
			Tr. Time (sec)	49	381	193	236	242	247	137	123	104	190
	80%	**1470 x 2074**	Acc. (%)	93.5	**95.9**	92.5	93.5	93.8	91.4	94.9	93.3	88.9	**93.1**
			Tr. Time (sec)	66	448	212	245	249	240	94	61	80	188
	90%	**1470 x 2333**	Acc. (%)	93	94.4	92.7	93.7	93.8	90.5	**95.2**	93.5	89.8	**93.0**
			Tr. Time (sec)	62	585	265	302	308	300	200	152	93	252

Table 19: Accuracy and Training Time of fused features (FV2-FV3) with the selection of features using Entropy and ReliefF on UC Merced dataset

Input dimension = 1470 x 2976	Percent. Selection	Output Dimension		Linear Disc.	En. Sub. Disc	Linear SVM	QSVM	Cubic SVM	MSVM	Wide NN	Med. NN	Cosine KNN	Avg.
	1%	**1470 x 26**	Acc. (%)	64	60.6	67.6	**73.7**	72.4	72.9	67.8	61.9	65.9	**67.4**
			Tr. Time (sec)	0.7	17	17	21	21	22	24	23	3	17
	4%	**1470 x 104**	Acc. (%)	86.5	84.1	84.9	87.5	**87.6**	86.5	84.4	83.2	77.8	**84.7**
			Tr. Time (sec)	1	27	23	27	25	28	30	29	11	22
	9%	**1470 x 234**	Acc. (%)	**91.1**	90.6	89.2	90	90.2	88.3	89.2	87.3	85.1	**89.0**
			Tr. Time (sec)	2	38	33	35	41	41	42	38	19	32
	16%	**1470 x 415**	Acc. (%)	**93**	92.5	91.4	91.4	91.3	87.8	92.2	89.7	85.4	**90.5**
			Tr. Time (sec)	5	58	46	53	58	60	42	34	27	43
	26%	**1470 x 648**	Acc. (%)	94.1	**95.1**	91.9	91.7	91..3	88.9	93.2	88.6	85.6	**91.1**
			Tr. Time (sec)	11	82	65	75	79	80	40	35	26	55
	36%	**1470 x 933**	Acc. (%)	91.9	**95.7**	91.7	91.4	91.4	89.5	93	90.6	86.2	**91.3**
			Tr. Time (sec)	24	133	99	111	116	113	53	47	37	81
	49%	**1470 x 1270**	Acc. (%)	78.6	**96.5**	92.5	92.9	92.5	90	**93.5**	92.9	87.9	**90.8**
			Tr. Time (sec)	50	198	134	151	155	151	64	50	56	112
	64%	**1470 x 1659**	Acc. (%)	80	**96**	92.7	93.7	93.7	90.3	94.4	94.4	87.5	**91.4**
			Tr. Time (sec)	31	282	156	182	186	178	30	19	39	123
	81%	**1470 x 2100**	Acc. (%)	92.1	96	93	93.5	**94.3**	90.5	95.2	93.3	87.9	**92.9**
			Tr. Time (sec)	60	479	232	261	266	258	113	78	86	204

Chapter 5
Conclusion

5.1 Conclusion

In this thesis, a methodology is proposed for the classification of satellite images in which multiple features extracted from different pre-trained CNN models are fused prior to features selection. As compared to classification accuracies of single layers, this features fusion approach resulted in improving classification accuracy. The features fusion approach is followed by a two-level feature selection approach. In the first level of features selection, Entropy is used for the selection of high-ranked features for classification. In the second level of features selection, the ReliefF algorithm is used to select high-ranked features from already selected features. This approach significantly reduced the features and classification accuracy is almost close to the fused features classification accuracy.

References

[1] A. Alem and S. Kumar, "Deep Learning Methods for Land Cover and Land Use Classification in Remote Sensing: A Review," in 2020 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions)(ICRITO), 2020: IEEE, pp. 903-908.

[2] A. Vali, S. Comai, and M. Matteucci, "Deep learning for land use and land cover classification based on hyperspectral and multispectral earth observation data: A review," Remote Sensing, vol. 12, no. 15, p. 2495, 2020.

[3] P. Helber, B. Bischke, A. Dengel, and D. Borth, "Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification," IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 12, no. 7, pp. 2217-2226, 2019.

[4] J. Qin, K. Chao, M. S. Kim, R. Lu, and T. F. Burks, "Hyperspectral and multispectral imaging for evaluating food safety and quality," Journal of Food Engineering, vol. 118, no. 2, pp. 157-171, 2013.

[5] F. Marschner, "Major land uses in the United States (map scale 1: 5,000,000)," USDA Agricultural Research Service, Washington, DC, vol. 252, 1950.

[6] J. R. Anderson, A land use and land cover classification system for use with remote sensor data. US Government Printing Office, 1976.

[7] J. Ma et al., "Evaluation of Different Approaches of Convolutional Neural Networks for Land Use and Land Cover Classification Based on High Resolution Remote Sensing Images," in 2019 IEEE International Conference on Signal, Information and Data Processing (ICSIDP), 2019: IEEE, pp. 1-4.

[8] P. Scheunders, D. Tuia, and G. Moser, "Contributions of machine learning to remote sensing data analysis," in Data processing and analysis methodology: Elsevier, 2018, pp. 199-243.

[9] Y. Shao and R. S. Lunetta, "Comparison of support vector machine, neural network, and CART algorithms for the land-cover classification using limited training data points," ISPRS Journal of Photogrammetry and Remote Sensing, vol. 70, pp. 78-87, 2012.

[10] H. Ghanbari, M. Mahdianpari, S. Homayouni, and F. Mohammadimanesh, "A meta-analysis of convolutional neural networks for remote sensing applications,"

IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 14, pp. 3602-3613, 2021.

[11] S. Talukdar, P. Singha, S. Mahato, S. Pal, Y.-A. Liou, and A. Rahman, "Land-use land-cover classification by machine learning classifiers for satellite observations—a review," Remote Sensing, vol. 12, no. 7, p. 1135, 2020.

[12] S. Dara and P. Tumma, "Feature extraction by using deep learning: A survey," in 2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA), 2018: IEEE, pp. 1795-1801.

[13] J. E. Ball, D. T. Anderson, and C. S. Chan Sr, "Comprehensive survey of deep learning in remote sensing: theories, tools, and challenges for the community," Journal of Applied Remote Sensing, vol. 11, no. 4, p. 042609, 2017.

[14] L. Ma, Y. Liu, X. Zhang, Y. Ye, G. Yin, and B. A. Johnson, "Deep learning in remote sensing applications: A meta-analysis and review," ISPRS journal of photogrammetry and remote sensing, vol. 152, pp. 166-177, 2019.

[15] R. Naushad, T. Kaur, and E. Ghaderpour, "Deep Transfer Learning for Land Use and Land Cover Classification: A Comparative Study," Sensors, vol. 21, no. 23, p. 8083, 2021.

[16] D. H. Wolpert and W. G. Macready, "Coevolutionary free lunches," IEEE Transactions on evolutionary computation, vol. 9, no. 6, pp. 721-735, 2005.

[17] T. Akram, B. Laurent, S. R. Naqvi, M. M. Alex, and N. Muhammad, "A deep heterogeneous feature fusion approach for automatic land-use classification," Information Sciences, vol. 467, pp. 199-218, 2018.

[18] J. Jagannathan and C. Divya, "Deep learning for the prediction and classification of land use and land cover changes using deep convolutional neural network," Ecological Informatics, vol. 65, p. 101412, 2021.

[19] X. Chen, S. Xiang, C.-L. Liu, and C.-H. Pan, "Vehicle detection in satellite images by hybrid deep convolutional neural networks," IEEE Geoscience and remote sensing letters, vol. 11, no. 10, pp. 1797-1801, 2014.

[20] R. Nijhawan, D. Joshi, N. Narang, A. Mittal, and A. Mittal, "A futuristic deep learning framework approach for land use-land cover classification using remote

sensing imagery," in Advanced Computing and Communication Technologies: Springer, 2019, pp. 87-96.

[21] A. M. Abdi, "Land cover and land use classification performance of machine learning algorithms in a boreal landscape using Sentinel-2 data," GIScience & Remote Sensing, vol. 57, no. 1, pp. 1-20, 2020.

[22] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 2818-2826.

[23] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 4700-4708.

[24] G.-S. Xia et al., "AID: A benchmark data set for performance evaluation of aerial scene classification," IEEE Transactions on Geoscience and Remote Sensing, vol. 55, no. 7, pp. 3965-3981, 2017.

[25] Y. Chen, H. Jiang, C. Li, X. Jia, and P. Ghamisi, "Deep feature extraction and classification of hyperspectral images based on convolutional neural networks," IEEE Transactions on Geoscience and Remote Sensing, vol. 54, no. 10, pp. 6232-6251, 2016.

[26] B. Hedayatnia, M. Yazdani, M. Nguyen, J. Block, and I. Altintas, "Determining feature extractors for unsupervised learning on satellite images," in 2016 IEEE International Conference on Big Data (Big Data), 2016: IEEE, pp. 2655-2663.

[27] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?," Advances in neural information processing systems, vol. 27, 2014.

[28] L. Ladha and T. Deepa, "Feature selection methods and algorithms," International journal on computer science and engineering, vol. 3, no. 5, pp. 1787-1797, 2011.

[29] S. Khalid, T. Khalil, and S. Nasreen, "A survey of feature selection and feature extraction techniques in machine learning," in 2014 science and information conference, 2014: IEEE, pp. 372-378.

[30] A. Jović, K. Brkić, and N. Bogunović, "A review of feature selection methods with applications," in 2015 38th international convention on information and

communication technology, electronics and microelectronics (MIPRO), 2015: Ieee, pp. 1200-1205.

[31] Q. Zou, L. Ni, T. Zhang, and Q. Wang, "Deep learning based feature selection for remote sensing scene classification," IEEE Geoscience and Remote Sensing Letters, vol. 12, no. 11, pp. 2321-2325, 2015.

[32] R. J. Urbanowicz, M. Meeker, W. La Cava, R. S. Olson, and J. H. Moore, "Relief-based feature selection: Introduction and review," Journal of biomedical informatics, vol. 85, pp. 189-203, 2018.

[33] G. Roffo, "Feature selection library (MATLAB toolbox)," arXiv preprint arXiv:1607.01327, 2016.

[34] S. F. Rosario and K. Thangadurai, "RELIEF: feature selection approach," International journal of innovative research and development, vol. 4, no. 11, 2015.

[35] X. Yao, J. Han, L. Guo, S. Bu, and Z. Liu, "A coarse-to-fine model for airport detection from remote sensing images using target-oriented visual saliency and CRF," Neurocomputing, vol. 164, pp. 162-172, 2015.

[36] O. Russakovsky et al., "Imagenet large scale visual recognition challenge," International journal of computer vision, vol. 115, no. 3, pp. 211-252, 2015.

[37] A. S. Sankar, S. S. Nair, V. S. Dharan, and P. Sankaran, "Wavelet sub band entropy based feature extraction method for BCI," Procedia Computer Science, vol. 46, pp. 1476-1482, 2015.

[38] Y. Yang and S. Newsam, "Bag-of-visual-words and spatial extensions for land-use classification," in Proceedings of the 18th SIGSPATIAL international conference on advances in geographic information systems, 2010, pp. 270-279.

[39] G. Cheng, J. Han, and X. Lu, "Remote sensing image scene classification: Benchmark and state of the art," Proceedings of the IEEE, vol. 105, no. 10, pp. 1865-1883, 2017.