

**AUTOMATIC FEATURE
EXTRACTION FROM SATELLITE
IMAGERY FOR REMOTE SENSING**

Submitted in partial fulfillment of the
requirements for the award of
Bachelor of Technology degree in Information Technology

By

**D DIYA MARY (Reg. No - 39120026)
NANDANA. S. NAIR (Reg. No – 39120074)**



**DEPARTMENT OF INFORMATION TECHNOLOGY
SCHOOL OF COMPUTING**

SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

**Accredited with Grade “A” by NAAC | 12B Status by UGC | Approved by AICTE
JEPPIAAR NAGAR, RAJIV GANDHISALAI,
CHENNAI - 600119**

MAY - 2023



SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

Accredited with Grade "A" by NAAC | 12B Status by UGC | Approved by AICTE

www.sathyabama.ac.in

DEPARTMENT OF INFORMATION TECHNOLOGY

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the Bonafide work of **D Diya Mary (39120026)** and **Nandana. S. Nair (39120074)** who carried out the Project Phase-1 entitled **"AUTOMATIC FEATURE EXTRACTION FROM SATTELITE IMAGERY FOR REMOTE SENSING"** under my supervision from January 2023 to May 2023.

Internal Guide

Dr. S. REVATHY M.E., Ph.D.

Head of the Department

Dr. R. SUBHASHINI, M.E., Ph.D.

Submitted for Viva voce Examination held on _____

Internal Examiner

External Examiner

DECLARATION

I, **D Diya Mary**(Reg.No- **39120026**) hereby declare that the Project Phase-1 Report entitled “**AUTOMATIC FEATURE EXTRACTION FROM SATTELITE IMAGERY FOR REMOTE SENSING**” done by me under the guidance of **Dr. S. REVATHY, M.E.,Ph.D** is submitted in partial fulfillment of the requirements for the award of Bachelor of Technology degree in **Information Technology**.



DATE:

D Diya Mary

PLACE: Chennai

SIGNATURE OF THECANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management of SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala M.E., Ph. D, Dean**, School of Computing, **Dr. R. Subhashini M.E., Ph.D.**, Head of the Department of Information Technology for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr. S. Revathy M.E., Ph.D.**, for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my Phase-2 project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Information Technology** who were helpful in many ways for the completion of the project.

ABSTRACT

Characteristic extraction from satellite images has steadily but progressively grown in recent years. Numerous feature extraction techniques have been created as a result of the increase. The loss of certain features is one of the many obstacles that must be overcome while selecting the best path for feature extraction. Satellite images are often composed of many objects crowdedly distributed on the surface, the description of which usually demands more discriminative features as well as local semantics. However, when applied to scene classification, most of the existing convolution neural networks (ConvNets) tend to depict global semantics of images, and the loss of low- and mid-level features can hardly be avoided, especially when the model goes deeper. We suggest utilizing a convolutional neural network approach to tackle these issues and extract attributes from satellite images supplied by an open source dataset called MLRSNet. Images with labels explaining each image's characteristics are displayed in the output. This makes it simpler to recognise and understand the many components in satellite images.

TABLE OF CONTENTS

Chapter No	TITLE	Page No.
	ABSTRACT	v
	LIST OF ABBREVIATIONS	viii
	LIST OF FIGURES	ix
1	INTRODUCTION	10
2	LITERATURE SURVEY	12
2.1	Inferences from Literature Survey	12
2.2	Open problems in Existing System	18
3	REQUIREMENTS ANALYSIS	19
3.1	Feasibility Studies/Risk Analysis of the Project	19
3.2	Software Requirements Specification Document	20
4	DESCRIPTION OF PROPOSED SYSTEM	21
4.1	Selected Methodology or process model	21
4.2	Architecture / Overall Design of Proposed System	24
4.3	Description of Software for Implementation and Testing plan of the Proposed Model/System	25
4.4	Project Management Plan	28
5	IMPLEMENTATION DETAILS	29
5.1	Development and Deployment Setup	29
5.2	Algorithms	29
5.3	Testing	32
6	RESULTS AND DISCUSSION	33
7	CONCLUSION	36
7.1	Future work	36
7.2	Research Issues	37
7.3	Implementation Issues	37

REFERENCES	38
APPENDIX	44
A. SOURCE CODE	44
B. SCREENSHOTS	49
C. RESEARCH PAPER	53

LIST OF ABBREVIATIONS

WORD	ABBREVIATION
CNN	Convolutional Neural Network
GIS	Geographic Information System
PCA	Principal Component Analysis
SVM	Support Vector Machine
ResNet	Residual Network
MLRSNet	Multi-label High Spatial Resolution Remote Sensing Dataset
ANN	Artificial Neural Network

LIST OF FIGURES

FIGURE NO	FIGURE NAME	Page No.
4.1	System Architecture	24
4.2	Images from MLRSNet dataset	25
6.1	Buildings and roads highlighted in the images	33
6.2	Trees and tracks highlighted in the images	34
6.3	Images Before and After Feature Extraction	34
6.4	Graph Plot 1	35
6.5	Area covered by each category	35
A	Feature Extracted Output 1	49
B	Feature Extracted Output 2	49
C	Feature Extracted Output 3	50
D	Feature Extracted Output 4	50
E	Feature Extracted Output 5	51
F	Images Before and After Feature Extraction	51
G	Graph Plot 2	52
H	Heat Map	52
I	Graph Plot 3	52

CHAPTER 1

INTRODUCTION

Over the years, the area of feature extraction has widened significantly. For this, satellite images are frequently employed and significantly improve outcomes. There are several items in images used for feature extraction, all of which must be precisely recognised for effective outcomes. There are various techniques to make it simpler to extract characteristics from an image for processing. Semantic scene classification is one of the activities that helps with picture categorization and retrieval. There have been researches [1] which sheds information on the process of extracting different pictures using CNN from high resolution satellite photos. This work used several datasets and powerful CNNs, which made it easier to see advancements over the current method.

To extract the features, a big initial set of data should be split into smaller, more manageable groups. As a result, processing will be simplified. The notion that these large data sets encompass a wide range of diverse characteristics is their most notable aspect. A huge amount of computing power is needed to manage these variables. By selecting and merging variables into features, feature extraction assists to substantially cut down the amount of data by extracting the critical qualities from all those large data sets. These features properly and clearly portray the actual data set while being simple to use.

Feature extraction is a process, in which, an initial set of the raw data is divided and reduced to more manageable groups. So when you want to process it will be easier. The most important characteristic of these large data sets is that they have a large number of variables. These variables require a lot of computing resources to process. So Feature extraction helps to get the best feature from those big data sets by selecting and combining variables into features, thus, effectively reducing the amount of data. These features are easy to process, but still able to describe the actual data set with accuracy and originality.

The technique of extracting the features is useful when you have a large data set and need to reduce the number of resources without losing any important or relevant information. Feature extraction helps to reduce the amount of redundant data from the data set. In the end, the reduction of the data helps to build the model with less machine effort and also increases the speed of learning and generalization steps in the machine learning process.

Scene classification and Feature Extraction is one of the main tasks in aerial image understanding. In this process, a semantic label is assigned to images collected from remote locations. Scene classification is also known as image classification when it comes to remote sensing datasets, and these two expressions are equally used throughout our paper. Remote sensing classification has found its usage in many areas: battlefields, traffic control, and disaster observation. Remote sensing images are a composition of scene information and contain an enormous feature space describing the texture. Scene composition of remote sensing images are often compound, and because of this, it might be challenging to get the semantic label right from the image data. This is the reason for developing numerous scene classification techniques. In general, they can belong to one of the following categories: methods that utilize low-level image features, methods using mid-level image representation, and methods using high-level image features.

CHAPTER 2

LITERATURE SURVEY

A literature review is the identification and examining of the existing research work in the chosen field to gain valuable information [1]. Literature review was performed to understand the existing learning algorithms and to choose the suitable method for image classification. When conducting research, a literature review is an essential part of the project because it covers all previous research done on the relative problem statements and sets the platform on which the current research is based. As the study was made to compare algorithms, the literature review was performed to identify the most efficient algorithm of each kind. The algorithms identified were further used in experimentation.

2.1 INFERENCES FROM LITREATURE SURVEY

The literature relevant to the current study was studied to gain a better understanding of the current state of the field. The findings of this preliminary study are described in this section.

Remote Sensing [2] is a technique introduced in the early 1960s for data analysis and interpretation. Remote sensing classification has found its usage in many areas: battlefields, traffic control, and disaster observation [3,4]. Remote sensing images are a composition of scene information and contain an enormous feature space describing the texture. Remote sensing collects a huge amount of satellite data. Satellite remote sensing imagery covers a large geographical area with high temporal frequency as compared to other imagery. Interpretation of satellite images helps in a variety of applications such as environmental conservation and management, water bodies research, environmental study after natural disasters, soil quality studies, deriving land use and land cover information, meteorology simulations and preventing natural disasters, studying climatic change evolution.

Different techniques are used for data extraction from remote sensing images. Satellite image classification groups together the pixels of the image into several different defined classes[5]. The classification helps in extracting the information contained in different bands of the satellite sensor and the information is extracted in terms of digital numbers which are then converted to a category.

S. Muhammad et al., [6] proposed a Supervised Satellite Image Classification method using the Decision Tree Technique. This method extracts features from satellite images based on pixel intensity and color. The method classifies satellite images using a decision tree with the support of identified objects.

In the paper, "Very High-Resolution Satellite Image Classification Using Fuzzy Rule-Based Systems", J. Shabnam et al. [7] introduced a Supervised Satellite Image Classification method to classify very high-resolution satellite images into specific classes using fuzzy logic. This method classifies satellite images into five major classes: shadow, vegetation, road, building, and bare land. It uses image segmentation and fuzzy techniques for satellite image classification. It applies 2 levels of segmentation- first level segmentation identifies and classifies shadow, vegetation, and road and the second level regions. Fuzzy techniques are used to improve the classification accuracy at the borders of objects.

A fast and automatic approach to detect objects in images of large size is proposed in "Hybrid approach of video indexing and machine learning for rapid indexing and highly precise object recognition" [8]. Experimentations were performed supported by some calculations to show that the proposed technique provides faster recognition. A similar attempt was made in "Object recognition on satellite images with biologically-inspired computational approaches" [9] as well to improve the accuracy of object detection in complex satellite images. This solution attempted to mimic the human vision system to improve accuracy of detection.

In "Ship detection and recognition in high-resolution satellite images" [10] a GIS-based application is used to detect ships in high resolution satellite images. This is very useful in monitoring maritime activities. In "Aircraft type recognition in satellite images" [11], the task of identifying type of aircraft in satellite imagery was done with

hierarchical recognition methods. By experimentally comparing different approaches, a superior method was recommended at the end. The identification of vehicles in satellite images was performed in "Classification-based vehicle detection in high resolution satellite images" [12]. This was done using segmentation method followed by classification. The usage of support vector machine for classification in satellite images was discussed majorly in "Active learning with support vector machines in remotely sensed image classification" [13]. After several experiments it was shown that this algorithm can be faster and more accurate using active learning.

According to reference "A sea-land segmentation algorithm based on Gray Smoothness Ratio" [14], in place of satellite images, infra-red images were used and a sea-land segmentation algorithm was proposed. Grayness and texture of surface were studied to arrive at a Gray Smoothing Ratio as a feature descriptor, around which an algorithm was developed to classify and segment the images into sea and land. In addition to this, this algorithm also had a function to fill gaps that arise due to natural conditions such as fog or cloudy weather.

In "Circular object recognition from satellite images" [15], the authors implemented a machine learning technique using Adaboost. The primary point of study in this paper, was to separate circular structured things in satellite images. Adaboost, an ensemble of various machine learning classification algorithms to provide a superior classifier, and Haar, a series of statistical feature descriptors were primarily used to implement the tool. Adaboost was trained iteratively and Haar descriptors were directly implemented over the gray-level images so as to avoid segmentation. The final parts of the project used a cascade of Adaboost classifiers so as to avoid re-detection at the various scales of the image. The end result is a segregated set of images.

In the paper "Spatial technique for Image Classification" [16] Selim proposed a classification method using the Bayesian technique. This method uses spatial information for the classification of high-resolution satellite images. It performs classification in two phases. Phase 1: spectral and textural features are extracted for each pixel to train Bayesian classifiers with discrete nonparametric density models. Phase 2: iterative split-and-merge algorithm is used to convert the pixel-level classification maps into contiguous regions.

"Comparison of Four Classification Methods to Extract Land Use and Land Cover from Raw Satellite Images for Some Remote Arid Areas, Kingdom of Saudi Arabia" [17] on ISODATA technique is the most common unsupervised satellite classification method. It creates a predefined number of unlabeled clusters/classes in a satellite image. Later on meaningful labels are assigned to the clusters. ISODATA parameters need several parameters that control the number of clusters and iterations to be run.

In "Comparing sift descriptors and Gabor texture features for classification of remote sensed imagery" [18], the authors used the IKONOS dataset to compare SIFT and Gabor texture features. In reference "Evaluating the potential of texture and color descriptors for remote sensing image retrieval and classification" [19], authors compare various colors and texture descriptors such as color histograms and local binary pattern (LBP) [20] descriptors. Authors of "Indexing of remote sensing images with different resolutions by multiple features" [21] use six various types of descriptors namely: SIFT, radiometric features, Grey Level Co-Occurrence Matrix (GLCM), Gaussian wavelet features [22], shape features [23], and Gabor filters, and made a combination of them in compound-feature figures with different spatial resolutions for remote sensing.

In "Land-use scene classification using multi-scale completed local binary patterns" [24], the authors proposed completed local binary patterns with multi-scales (MS-CLBP) for remote sensing image classification, and obtained cutting-edge results compared to other low-level methods. Another approach is to apply a sparse coding method as discussed in "High-resolution satellite scene classification using a sparse coding-based multiple feature combination" [25] where structural, spectral, and textural features are extracted and encoded. The state-of-the-art performance can be achieved by performing dimensionality reduction with Principal Component Analysis (PCA) algorithm, before concatenating multi-features, or with methods such as the Improved Fisher Vector (IFV) [26] and Vectors of Locally Aggregated Tensors (VLAT) [27]. The methods mentioned above do not incorporate the spatial distribution of visual words, which, according to some authors, is important.

CIELab color moments [28] are used with Latent Dirichlet Allocation (LDA) [29] and in “Land-use scene classification using multi-scale completed local binary patterns” [30]. GLCM [31] and edge orientation histogram (EOH) [32] are other ways of extracting different information in the LDA model. The probabilistic Latent Semantic Analysis (pLSA) technique [33] is adopted in “Scene classification based on multi-feature probabilistic latent semantic analysis for high spatial resolution remote sensing images” [34].

Support Vector Machine (SVM) [35] is a non-parametric unsupervised statistical classification method. It can be used to extract land maps. But, this method works on the assumption that there is no information on how to distribute the overall data. At the same time, SVM reduces satellite classification cost, increases speed, and improves accuracy. Maximum method is a supervised statistical approach for recognizing the patterns. It allocates pixels to appropriate classes based on the probability values of the pixels [36].

Global features are extracted by Nogueira et al. [37] from the CNN models, and they are fed to a classifier. This method has shown the effectiveness of transfer learning from object classification models. In all of these examples, the global or local extracted feature representations were acquired from CNNs that were pre-trained on datasets with natural images. Those features were utilized for the classification of remote sensing images.

The study in “Delving into deep representations for remote sensing image retrieval” [38] focuses on retrieving various images from high resolution satellite images using CNN technique. Multiple datasets were used in this study with strong CNNs to observe strong improvements over existing approach. “Surface object recognition with CNN and SVM in Landsat 8 images” [39] uses large set of image data to perform surface object recognition. Two methods based on CNN and SVM are proposed and compared. In the performed experiments, it is found that CNN had better performance. In [40], isolated airstrips are identified from satellite images using SVM method after preprocessing the images. The algorithm performed quite well providing an accuracy of about 94%. It was able to identify airstrips among other

common linear objects like roads, canals, etc.

In general, scene classification techniques can belong to one of the following categories: methods that utilize low-level image features, methods using mid-level image representation, and methods using high-level image features. With methods using low-level visual features, aerial scenes are classified from low-level visual descriptors: spectral, textural, structural, and so on. The local fluctuation of structures in remote sensing images is modeled by Scale Invariant Feature Transform (SIFT) [41] as a local structure descriptor.

The mid-level visual representation methods are another group of methods used for scene representation, which attempt to represent scenes with the statistical representation of high-degree locally extracted image features. They first perform local image feature extraction from local patches, using descriptors such as color histograms or SIFT. Then these features are encoded for composing a mid-level representation for remote sensing images. A commonly used mid-level method is bag-of-visual-words (BoVW) [42].

The third group of techniques for image classification rely on high-level vision information. There are a lot of computer vision assignments that can be successfully solved using deep learning methods: image classification, object recognition, and image retrieval. Compared to the other methods, high-level methods can obtain more abstract and discriminative semantic representations, which results in the achievement of better classification performance. Feature extraction with convolutional neural networks (CNNs), pre-trained on massive datasets [43], accomplishes great performance for aerial scene classification. There are many freely available pre-trained deep CNN architectures: ResNet, DenseNet, Inception, Xception, and so on.

Contrary to the deep learning methods described above, an alternative is to optimize CNN from scratch, i.e., from randomly set parameter values. However, as discussed in [40], using the UC-Merced dataset [21] and the WHU-RS19 dataset [41] for full training of CNNs like CaffeNet [42] or GoogLeNet [43] shows poor classification performance. The reason for this poor performance and poor generalization is that

large CNN architectures have a huge number of parameters, and model training from aerial scene datasets with up to thousands of images inevitably gets stuck in the local minima. However, training CNNs from scratch using larger datasets, such as AID [44] and NWPU-RESISC45 [45], has shown good results many a times.

2.2 OPEN PROBLEMS IN EXISTING SYSTEM

When applied to scene classification, most of the existing convolution neural networks tend to depict global semantics of images, and the loss of low- and mid-level features can hardly be avoided, especially when the model goes deeper.

For the feature extraction procedure, several different strategies have been suggested. Automatic object identification is one common method for extracting features, however it still has several flaws. Automatic line identification systems are challenging to build due to the complexity of remotely sensed images. Consequently, a variety of statistical methods have been widely used to classify data on land use or land cover. Researchers' attention has switched to cutting-edge, dependable categorization methods like Artificial Neural Networks as satellite sensors have been created.

However, current CNNs tend to preserve global semantics since objects are more concentrated and have a more stable context relationship in natural image scenes. For aerial scenes, focusing on the global semantics may lead to the incapability of discerning discriminative features in a scene and the weakening of scene representation ability.

To tackle these challenges, we propose use of Res-Net for feature extraction. It regards aerial scene classification as a multiple-instance learning problem so that local semantics can be further investigated.

CHAPTER 3

REQUIREMENT ANALYSIS

3.1 FEASIBILITY STUDIES/RISK ANALYSIS OF THE PROJECT

There are several challenges in image processing, which are concerned with its deep learning for machines. The image processing algorithms may be modeled on multidimensional systems. The noises and inconsistencies are eliminated using a wider range of algorithms. The factors play a key role in refining and preparing some insightful models or patterns, which are tested, validated, and refined to turn into machine learning. But, there are many image processing issues out of which few are as follows:

Image segmentation

It is all about breaking down a digital picture into various subsets. These groups are known as image segments, which simplifies processing. The problem begins to appear when it is segmented. For a human mind, it's not difficult because it can instinctively extract object information. Making a machine behave or act like our brain is a challenge. The machines or devices must be able to see and understand images as we do.

Image classification

It refers to associating one label or more labels to a given image. Classification is not easy. There are several roadblocks that may interfere when you classify them. Assigning the label to an image can have challenges related to variations. These can be like Scale Variation, View-Point Variation, Illumination, Object Detection, Object Localization.

Multiple Aspect Ratios and Spatial Sizes

The object may vary in size and ratio. Therefore, the detection algorithms find it tough to deal with these different scales and views. Issues like Intra-Class Variation, Viewpoint Variation, Occlusion, Deformation, Limited Data, Background Clutter.

Image Enhancement

This process really needs an intense focus on important features of an image. It is all because the image requires more brightness, or clear elements to appear as in X-ray films. In aerial photos, the edges or lines require sharpening for a crystal clear view of buildings or other objects.

Deep learning and computer vision can help in image recognition & then in its enhancement. This phase is extremely difficult to come across. It involves the image processing problem statement in recognition and seeing them through the computer vision.

3.2 SOFTWARE REQUIREMENTS SPECIFICATION DOCUMENT

HARDWARE CONFIGURATION

Processor	:	Processor Intel CORE i3 and above
Internet Connection	:	Existing telephone lines, Data card, Fiber net
RAM	:	minimum 4GB
Hard disk	:	100 GB

SOFTWARE CONFIGURATION

Operating System	:	Windows, Mac
Language	:	Python 3.8.5
GUI	:	Spyder(Anaconda3)
Libraries Used	:	Pandas, numpy, tensorflow, seaborn, csv, matplotlib

CHAPTER 4

DESCRIPTION OF PROPOSED SYSTEM

This system uses Res-Net50 for feature extraction from satellite images. It will use a multiple instance densely-connected convolutional neural network for aerial scene classification. It is an end-to-end solution under the direct supervision of scene labels. Moreover, it has much fewer parameters compared with current CNN models.

We conduct more experiments on benchmarks to demonstrate the effectiveness of our proposed method, and we present a more detailed experiment analysis and in-depth discussion to thoroughly evaluate our proposed method.

With more training samples, the advantages of deep learning based methods become more obvious when compared to the middle and low level feature based methods and the overall accuracy becomes much more promising. Although spatial pyramid pooling is an effective strategy for many ground image task, it mainly utilizes the multiscale information in an image while the effort to focus on relevant local semantics is limited.

4.1 SELECTED METHODOLOGY OR PROCESS MODEL

The process's initial stage is to collect the dataset. In this case, we employed the MLRSNet dataset, a remote sensing dataset with a focus on visualization. The MLRSNet comprises 46 categories and 109,161 samples in total. The pictures size is 256 by 256 and have a spatial resolution of 10–0.1 m. The photographs are then scaled, cropped, and brightness and contrast modifications are applied as part of data augmentation. The photos were then labelled and placed into specified classes.

Following the preparation of the dataset, installation of required packages is done to import certain libraries which will be utilized to execute the process. After this setting up the path for the dataset has to be done in order to access the images for training and testing. There are two types of imagery spectral content provided. The 3-band images are the traditional RGB natural color images. The 16-band images contain

spectral information by capturing wider wavelength channels. Every object class is described in the form of Polygons and MultiPolygons, which are simply a list of polygons. In this dataset there are a set of geo-coordinates that are in the range of $x = [0,1]$ and $y = [-1,0]$. These coordinates are transformed such that we obscure the location of where the satellite images are taken from. To utilize these images, the grid coordinates of each image were provided with which performed scaling and aligned them with the images in pixels. The Xmax and Ymin for each image is required to do the scaling. Astype() command in adjust_contrast function returns a new DataFrame where the data types has been changed to the specified type. It is a Python library to store numpy arrays in Tagged Image File Format files, and read image and metadata from Tagged Image File Format like files. Training is carried out using RESNET50 in batches of 40. Resnet50 has over 23 million parameters and Keras package was used for the model. After that comes the testing step, where 180×180 photos are tested in a single batch. Followed by processing of images, we do the plotting of polygons for images and then compare it with original images. While comparing we increase the figure size and plot multipolygons which plot red lines to show the features. Then the image is displayed along with its id. Following this, the class labels is put in dictionary as key value pair. After this we plot the polygon class wise separately, where we get the polygon coordinates and plot it. Further, we create heatmaps to understand the frequency of classes vs the images. A bar graph is also created to show the areas of the objects in an image.

Scene classification

The loss of middle and low-level features can hardly be avoided when training current CNN models such as VGG-Net. Although shallower features are not that important when compared with semantic features for scene classification, adding these features are beneficial to the extraction of more discriminative features and the boost of scene classification performance. But some recent approaches to preserve shallower features are not end-to-end solutions and are not easy to adjust to different tasks and datasets.

Convolutional neural network

We further propose a more advanced multiple instance densely-connected convolutional neural network for aerial scene classification. The framework is an end-to-end solution under the direct supervision of scene labels. Moreover, it has much fewer parameters compared with current CNN models.

Among these deep learning methods, convolutional neural networks are widely utilized. The effectiveness can be explained by the end-to-end framework and multi-stage feature extraction architecture. Compared with traditional hand-crafted features, features from deep learning methods are more discriminative and are stronger at semantic representation.

Res-Net

A residual neural network (ResNet) is an artificial neural network (ANN). It is a gateless or open-gated variant of the HighwayNet, the first working very deep feedforward neural network with hundreds of layers, much deeper than previous neural networks.

In order to solve the problem of the vanishing/exploding gradient, this architecture introduced the concept called Residual Blocks. In this network, we use a technique called skip connections. The skip connection connects activations of a layer to further layers by skipping some layers in between. This forms a residual block. Resnets are made by stacking these residual blocks together.

The approach behind this network is instead of layers learning the underlying mapping, we allow the network to fit the residual mapping. So, instead of say $H(x)$, initial mapping, let the network fit as shown in equation 4.1.

Equation 4.1: $F(x) := H(x) - x$ which gives $H(x) := F(x) + x$.

Skipping effectively simplifies the network, using fewer layers in the initial training stages. This speeds learning by reducing the impact of vanishing gradients, as there are fewer layers to propagate through. The network then gradually restores the

skipped layers as it learns the feature space. Towards the end of training, when all layers are expanded, it stays closer to the manifold and thus learns faster. A neural network without residual parts explores more of the feature space. This makes it more vulnerable to perturbations that cause it to leave the manifold, and necessitates extra training data to recover.

4.2 ARCHITECTURE OF PROPOSED SYSTEM

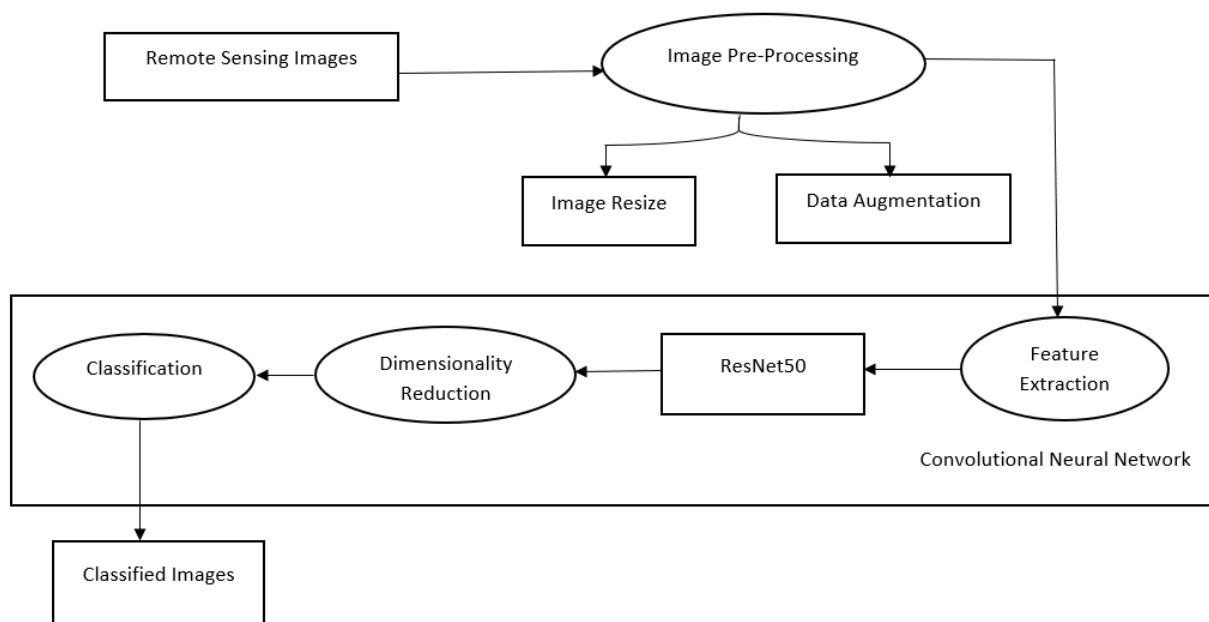


Fig 4.1 System Architecture

Figure 4.1 depicts the flow of the suggested methodology. The procedure starts with satellite pictures that go through image scaling and data augmentation during pre-processing. Data augmentation includes translating certain images to create more data from existing data. The phase of preprocessing improves quality of data and enhances performance which assists in an efficient training and further processes. Convolutional neural network, which contains four stages, is the next step. The first step is to use the Res-Net architecture to extract the image's features. The next step is dimensionality reduction, which lowers complexity. For reducing the dimensionality, we have used the principal component analysis which helps in preserving most of the information. The classification is then completed using the characteristics that were extracted. The system's output are the photos that have been categorized.

4.3 DESCRIPTION OF SOFTWARE FOR IMPLEMENTATION AND TESTING PLAN OF THE PROPOSED MODEL/SYSTEM

To implement this model, execution of program is done through Spyder. Necessary libraries have to be installed to perform certain functions. The dataset path has to be set in order to get the images for resizing and training. Then the functions are executed to alter images which helps in highlighting the features in an image.

DESCRIPTION OF DATASET

MLRSNet Dataset:

MLRSNet is composed of 109,161 labeled RGB images from all around the world annotated into broad categories: residential area, desert, forest, intersection, lake, river, trees, tracks. The number of sample images varies greatly with different broad categories, from 1500 to 3000. Additionally, each image in the dataset is assigned several of 60 predefined class labels, and the number of labels associated with each image varies between 1 and 13. Besides, MLRSNet has multi-resolutions: the pixel resolution changes from about 10 m to 0.1 m, and the size of each multi-label image is fixed to 256×256 pixels to cover a scene with various resolutions. MLRSNet has a more significantly large variability in terms of geographic origins and number of object categories. Different from ImageNet (Deng et al., 2009), which collects the data of outdoor objects from ground-level views, MLRSNet describes the objects on Earth from an overhead perspective through satellite or aerial sensors.



Fig 4.2 Images from MLRSNet dataset

DESCRIPTION OF PROGRAMMING LANGUAGES AND GUI

Python is a high-level, general-purpose and a very popular programming language. Python programming language (latest Python 3) is being used in web development, Machine Learning applications, along with all cutting edge technology in Software Industry.

Anaconda Navigator is a desktop graphical user interface (GUI) included in Anaconda distribution that allows you to launch applications and easily manage conda packages, environments, and channels without using command-line commands. Navigator can search for packages on Anaconda.org or in a local Anaconda Repository.

Spyder is an open-source cross-platform integrated development environment (IDE) for scientific programming in the Python language. Spyder integrates with a number of prominent packages in the scientific Python stack, including NumPy, SciPy, Matplotlib, pandas, as well as other open-source software. Spyder includes support for interactive tools for data inspection and embeds Python-specific code quality assurance and introspection instruments. It is available cross-platform through Anaconda, on Windows, on macOS through MacPorts.

DESCRIPTION OF LIBRARIES USED

NumPy, which stands for Numerical Python, is a library consisting of multidimensional array objects and a collection of routines for processing those arrays. Using NumPy, mathematical and logical operations on arrays can be performed. NumPy provides a convenient and efficient way to handle the vast amount of data. NumPy is also very convenient with Matrix multiplication and data reshaping. NumPy is fast which makes it reasonable to work with a large set of data.

TensorFlow is a Python library for fast numerical computing created and released by Google. It is a foundation library that can be used to create Deep Learning models directly or by using wrapper libraries that simplify the process built on top of TensorFlow. Unlike other numerical libraries intended for use in Deep Learning like

Theano, TensorFlow was designed for use both in research and development and in production systems.

Keras is an open-source high-level Neural Network library, which is written in Python is capable enough to run on Theano, TensorFlow, or CNTK. It is made user-friendly, extensible, and modular for facilitating faster experimentation with deep neural networks. It not only supports Convolutional Networks and Recurrent Networks individually but also their combination.

Pandas is an open-source library that is made mainly for working with relational or labeled data both easily and intuitively. It provides various data structures and operations for manipulating numerical data and time series. Pandas is fast and it has high performance & productivity for users.

Matplotlib is a python library used to create 2D graphs and plots by using python scripts. It has a module named pyplot which makes things easy for plotting by providing feature to control line styles, font properties, formatting axes etc. It supports a very wide variety of graphs and plots namely - histogram, bar charts, power spectra, error charts etc. It is used along with NumPy to provide an environment that is an effective open source alternative for MatLab. It can also be used with graphics toolkits like PyQt and wxPython.

OpenCV is a Python open-source library, which is used for computer vision in Artificial intelligence, Machine Learning, face recognition, etc. In OpenCV, the CV is an abbreviation form of a computer vision, which is defined as a field of study that helps computers to understand the content of the digital images such as photographs and videos. The purpose of computer vision is to understand the content of the images. It extracts the description from the pictures, which may be an object, a text description, and three-dimension model, and so on. For example, cars can be facilitated with computer vision, which will be able to identify and different objects around the road, such as traffic lights, pedestrians, traffic signs, and so on, and acts accordingly.

Seaborn is an amazing visualization library for statistical graphics plotting in Python. It provides beautiful default styles and color palettes to make statistical plots more attractive. It is built on the top of matplotlib library and also closely integrated to the data structures from pandas. Seaborn aims to make visualization the central part of exploring and understanding data. It provides dataset-oriented APIs, so that we can switch between different visual representations for same variables for better understanding of dataset.

4.4 PROJECT MANAGEMENT PLAN

MONTH	PROGRESS
December 2022	Training the data and implementing the algorithm.
January 2023	Performing dimensionality reduction.
February 2023	Classification and getting results.

CHAPTER 5

IMPLEMENTATION DETAILS

5.1 DEVELOPMENT AND DEPLOYMENT SETUP

The implementation of the project included setting up the environment to execute it. Installation of Anaconda3 was carried out to utilize Spyder where the program code will be executed. Necessary packages such as numpy, pandas, seaborn were installed for efficient execution of code. All the images were downloaded and saved in a location on the computer to import them for training. The images were also classified with labels to ensure testing of the system was effective.

5.2 ALGORITHMS

HEX To RGB Conversion

1. Get the 2 left digits of the hex color code and convert to decimal value to get the red color level.
2. Get the 2 middle digits of the hex color code and convert to decimal value to get the green color level.
3. Get the 2 right digits of the hex color code and convert to decimal value to get the blue color level.

MinMaxScaler

Transform features by scaling each feature to a given range.

The transformation is given by:

$$X_std = (X - X.min(axis=0)) / (X.max(axis=0) - X.min(axis=0))$$

$$X_scaled = X_std * (max - min) + min$$

Where min, max = feature_range

RESNET50 Architecture:

Residual Networks, train residual functions with regard to the layer inputs rather than learning unsourced functions. Rather than expecting each layer in a stack to exactly match the desired underlying mapping, residual nets allow these layers to fit a residual mapping. Remaining blocks are stacked one over the other to build networks; a ResNet50 employs these blocks to have fifty layers (48 convolutional layers, one average pool layer and one MaxPool layer). Residual neural networks (RNNs) are artificial neural networks (ANNs) which construct networks using residual blocks.

The ResNet50 architecture abides by two key design principles. First, there are the same number of filters in each layer regardless of size if the output feature map. Second, although though size of the feature map is reduced by half, it contains twice as many filters in order to maintain the temporal complexity of each layer.

Issues about the overall time needed to train the layers concluded to the decision of redesigning the building block in this case as a bottleneck. This used a stack of three layers in place of the previous two levels. Each of the two-layer blocks in Resnet34 was converted into a three-layer bottleneck block to form the Resnet 50 architecture. This is far more accurate than the 34-layer ResNet model..

ResNet trains on F whereas every other algorithm learns from the result " Y " (X). Another way to say it is that ResNet aims to make $F(X)=0$ such that $Y=X$. Some model levels are skipped by a direct link known as a skip connection. The output differs as a result of this skip link. In the absence of a skip connection, input ' X ' is multiplied by the weights of the layer before the bias term has been introduced. It is followed by activation function $n, F()$, and its output can be seen in equation 5.1.

Equation 5.1: $F(w \cdot x + b) (=F(X))$

The output with use of skip connection technique, is shown in equation 5.2.

Equation 5.2: $F(X)+x$

In ResNet-50, there are two different types of blocks: Identity Blocks and Convolutional Blocks. When input size equals output size, the value "x" is added to the output layer. In the event that the case is different, a convolutional block is added to the shortcut path to equalise the input and output sizes.

Convolutional Neural Networks:

In this, Convolutional Neural Networks(CNN) algorithm was used. The Convolutional Neural Network is just a technique under deep learning that has the ability to take in an input image, discriminate between various items and components in the image, and allocate different objects and elements in the image importance. A CNN needs a lot less pre-processing than different categorization techniques. While filters in simple approaches are hand-engineered, CNN has the ability to learn these filters' qualities. It is CNN's responsibility to streamline the graphics without sacrificing any of the information that is necessary for making precise forecasts. This is critical for creating a design that is constructive at learning features while also being scalable to large datasets.

The convolution operation's main goal is to obtain high-level information from the input image, like edges. CNN does not have to employ just one convolutional layer. Typically, the first ConvLayer collects low-level properties such as edges, colour, and gradation direction. Even as number of layers rises, the architecture adjusts to the High-Level characteristics, resulting in a network that encompasses the dataset's images holistically in a way similar to what we do.

Like the Convolutional Layer, the Pooling layer is responsible for decreasing the size of the Convolved Feature. The aggregate of resources required to process the data will be subsided by dimensionality reduction. Apart from this, it facilitates appropriate model training by obtaining dominant traits which are rotational and positional invariant.

The i-th layer of a CNN is made up of the convolutional layer as well as the pooling layer. Depending on how complex the images are, the layer count may be increased

to concentrate on an even finer details, but this will require more processing resources. Max Pooling and Average Pooling are two types of pooling. Max Pooling returns the greatest value from region of the image covered by the Kernel. Average pooling, on the other hand, delivers the mean of all of the values from portion of the image encompassed by the Kernel. Max Pooling additionally acts as a noise reduction. It accomplishes de-noising along with dimensionality reduction in addition to thoroughly removing noisy activations. After working through method mentioned above, we were capable of assisting the model in understanding the properties. After that, the output will be flattened for classification before even being loaded into a conventional neural network.

5.3 TESTING

The testing phase included images being processed to get the features highlighted and respective labels given as output. Certain images which included trees and roads were given as input and after the execution the output was derived. The images could be seen with highlighted parts which had trees or roads. Along with the images the respective labels were also mentioned.

CHAPTER 6

RESULTS AND DISCUSSION

The results are displayed in the photos below, which highlight the characteristics that were taken out of each image. Along with the highlighted elements, the label that explains the scenario in the photograph is also displayed. The accuracy achieved by this methodology is 92%. Additionally, performance was evaluated for the CNN algorithm used which resulted in an accuracy of 87%. In Figure 6.1 buildings and roads are highlighted. The buildings demonstrate the region's dense population. Additionally, figure 6.2 illustrates the roadways that link the area's many locations which demonstrates how well connected it is and the sparse distribution of trees in the area. In figure 6.3, we can see all features highlighted in the image, it shows the tracks, trees and buildings are extracted. The frequency of each class that is any feature is depicted in figure 6.4 through bar graph. The area that each category such as trees or roads takes up in a picture is depicted in figure 6.5. Overall, this model facilitates the visual analysis of images and the comprehension of their aspects.

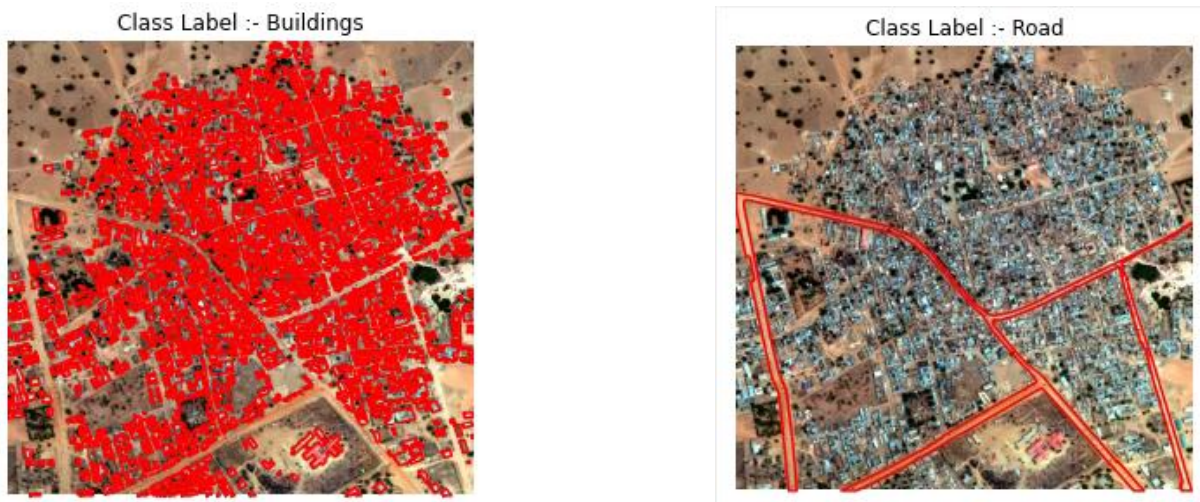


Fig 6.1 Buildings and roads highlighted in the images.



Fig 6.2 Trees and tracks highlighted in the images.

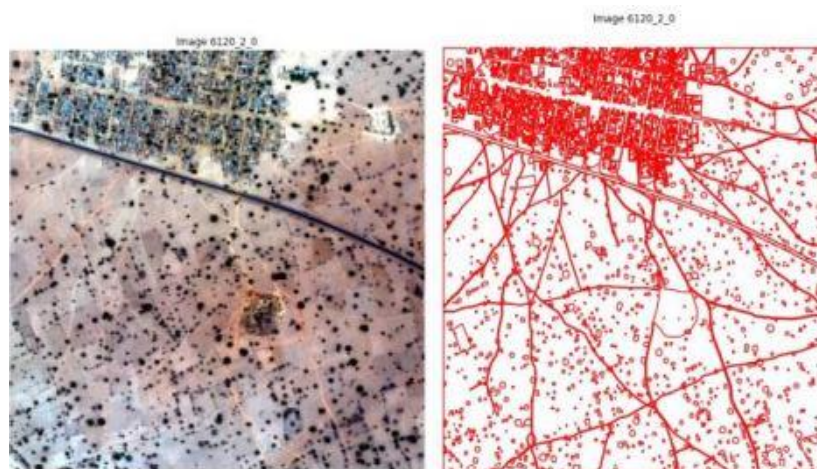


Fig 6.3 Image Before and After Feature Extraction

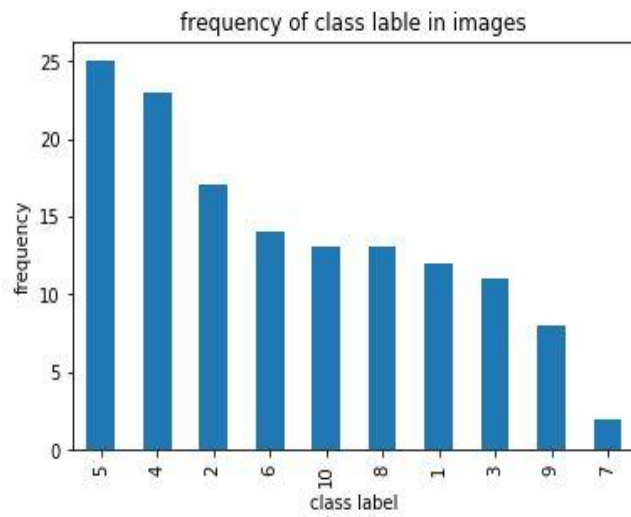


Fig 6.4 Graph Plot 1

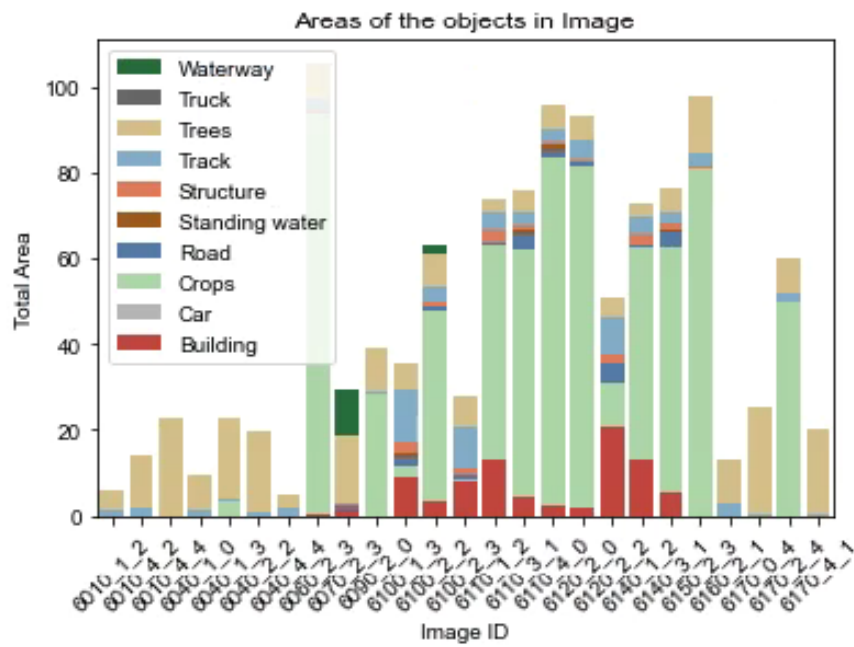


Fig 6.5 Area covered by each category.

CHAPTER 7

CONCLUSION

The MLRSNet is a dataset with high resolution images which enables deep learning to understand semantic scene interpretation, at least from an overhead perspective. Hierarchy, scalability, diversity, and multi-label are the characteristics that distinguish MLRSNet. Images with labels identifying the category to which they belong are extracted. Each image's particular traits are accentuated to make them stand out. These results assist in examining the visuals from a fresh angle, which may further aid in effective comprehension. The pictures make good use of the characteristics that were retrieved, making it easier to analyze a satellite image.

7.1 FUTURE WORK

This model can be further developed in future to gain more accuracy and also expand its working on larger satellite images. Larger images will require more intensive training and the algorithm will have to be modified to work with it. Training the system to extract multiple features from an image can be obtained.

The proposed technique for remote sensing image classification can be further explored with extracting features from lower layers of pre-trained deep CNN. In addition to this, stratification can also have an impact on the classification accuracy of the researched technique. So, to boost the accuracy and to come closer to the best classification practices, the split of the training/testing dataset should be stratified. All gathered knowledge and experience could be checked on other remote sensing datasets, not necessarily publicly available, preferably small scale, because the proposed classification method gives good results with a small portion of the training set.

7.2 RESEARCH ISSUES

Feature extraction process is not an easy task to implement and hence there were few challenges faced while researching on the project. Obtaining the right images which could be translated to a particular size for processing was difficult. The images had to be of good clarity for the machine to be able to interpret the feature in an image and train accordingly.

7.3 IMPLEMENTATION ISSUES

The implementation process gave rise to certain issues which had to be resolved to get an efficient result with more accuracy. The images had to be cropped and had to be adjusted to improve the clarity. While implementing the algorithm certain functions could not be executed without the right library and hence the libraries used had to be investigated in detail.

REFERENCES

- [1] H. M. Cooper, Synthesizing research: A guide for literature reviews, vol. 2. Sage, 1998.
- [2] E. C. Barret and L. F. Curtis, Introduction to Environmental Remote Sensing. Cheltenham, U.K.: Cheltenham Stanley Thornes Publishers.
- [3] Yang, W.; Yin, X.; Xia, G.S. Learning high-level features for satellite image classification with limited labeled samples. IEEE Trans. Geosci. Remote Sens. 2015, 53, 4472–4482. [CrossRef]
- [4] Huang, F.; Yan, L. Hull vector-based incremental learning of hyperspectral remote sensing images. J. Appl. Remote Sens. 2015, 9, 096022. [CrossRef]
- [5] W.G. Rees, Physical Principles of Remote Sensing, 2nd ed. Cambridge, U.K.: Cambridge Univ. Press, 2001.
- [6] Muhammad, S., Aziz, G., Aneela, N. and Muhammad, S. 2012. "Classification by Object Recognition in Satellite Images by using Data Mining". In Proc. Proceedings of the Congress on Engineering (WCE 2012), Vol I, July 4 - 6, London, U.K.
- [7] Shabnam Jabari and Yun Zhang, 2013. "Very High-Resolution Satellite Image Classification Using Fuzzy Rule-Based Systems", Algorithms, vol.6, no.4, pp. 762-781.
- [8] F. Tsutsumi and C. Nakajima, "Hybrid approach of video indexing and machine learning for rapid indexing and highly precise object recognition," in Image Processing, 2001. Proceedings. 2001 International Conference on, 2001, vol. 2, pp. 645–648.

- [9] M. I. Sina, P. Payeur, and A.-M. Cretu, "Object recognition on satellite images with biologically-inspired computational approaches," in *Applied Computational Intelligence and Informatics (SACI)*, 2012 7th IEEE International Symposium on, 2012, pp. 81–86.
- [10] J. Antelo, G. Ambrosio, J. González, and C. Galindo, "Ship detection and recognition in high-resolution satellite images," in *Geoscience and Remote Sensing Symposium*, 2009 IEEE International, IGARSS 2009, 2009, vol. 4, pp. IV–514.
- [11] J.-W. Hsieh, J.-M. Chen, C.-H. Chuang, and K.-C. Fan, "Aircraft type recognition in satellite images," in *Vision, Image and Signal Processing*, IEE Proceedings-, 2005, vol. 152, pp. 307–315.
- [12] L. Eikvil, L. Aurdal, and H. Koren, "Classification-based vehicle detection in high resolution satellite images," *ISPRS J. Photogramm. Remote Sens.*, vol. 64, no. 1, pp. 65– 72, Jan. 2009.
- [13] Z. Sun, Z. Liu, S. Liu, Y. Zhang, and B. Yang, "Active learning with support vector machines in remotely sensed image classification," in *Image and Signal Processing*, 2009. CISP'09. 2nd International Congress on, 2009, pp. 1–6.
- [14] X. Huihui, X. Qizhi, and H. Lei, "A sea-land segmentation algorithm based on Gray Smoothness Ratio," in *Earth Observation and Remote Sensing Applications (EORSA)*, 2016 4th International Workshop on, 2016, pp. 117–121.
- [15] H. Taud, J. C. Herrera-Lozada, M. Marciano-Melchor, R. Silva-Ortigoza, M. Olguín Carbajal, and others, "Circular object recognition from satellite images," in *Geoscience 24 and Remote Sensing Symposium (IGARSS)*, 2012 IEEE International, 2012, pp. 2324– 2327.
- [16] Selim Aksoy. 2006. "Spatial Techniques for Image Classification," in C. H. Chen, ed., *Signal and Image Processing for Remote Sensing*, CRC Press, pp.491-513.

- [17] Al-Ahmadi, F, S. and Hames, A, S. 2009. "Comparison of Four Classification Methods to Extract Land Use and Land Cover from Raw Satellite Images for Some Remote Arid Areas, Kingdom of Saudi Arabia", *Journal of King Abdulaziz University-Earth Sciences*, Vol. 20, No.1, pp: 167-191.
- [18] Yang, Y.; Newsam, S. Comparing sift descriptors and Gabor texture features for classification of remote sensed imagery. In *Proceedings of the IEEE International Conference on Image Processing*, San Diego, CA, USA, 12–15 October 2008; pp. 1852–1855.
- [19] Santos, J.A.d.; Penatti, O.A.B.; da Silva Torres, R. Evaluating the potential of texture and color descriptors for remote sensing image retrieval and classification. In *Proceedings of the VISAPP (2)*, Angers, France, 17–21 May 2010; pp. 203–208.
- [20] Ojala, T.; Pietikainen, M.; Maenpaa, T. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Trans. Pattern Anal. Mach. Intell.* 2002, 24, 971–987. [CrossRef]
- [21] Luo, B.; Jiang, S.; Zhang, L. Indexing of remote sensing images with different resolutions by multiple features. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* 2013, 6, 1899–1912. [CrossRef]
- [22] Luo, B.; Aujol, J.-F.; Gousseau, Y.; Ladjal, S. Indexing of satellite images with different resolutions by wavelet features. *IEEE Trans. Image Process.* 2008, 17, 1465–1472. [PubMed]
- [23] Luo, B.; Aujol, J.-F.; Gousseau, Y. Local scale measure from the topographic map and application to remote sensing images. *Multiscale Model. Simul.* 2009, 8, 1–29. [CrossRef]
- [24] Chen, C.; Zhang, B.; Su, H.; Li, W.; Wang, L. Land-use scene classification using multi-scale completed local binary patterns. *Signal image and Video Process.* 2015, 10, 745–752. [CrossRef]

- [25] Sheng, G.; Yang, W.; Xu, T.; Sun, H. High-resolution satellite scene classification using a sparse coding-based multiple feature combination. *Int. J. Remote Sens.* 2012, 33, 2395–2412. [CrossRef]
- [26] Perronnin, F.; Sanchez, J.; Mensink, T. Improving the fisher kernel for large-scale image classification. In *Proceedings of the European Conference on Computer Vision, Heraklion, Greece, 5–11 September 2010*; pp. 143–156.
- [27] Negrel, R.; Picard, D.; Gosselin, P.-H. Evaluation of second-order visual features for land-use classification. In *Proceedings of the International Workshop on Content-Based Multimedia Indexing (CBMI), Klagenfurt, Austria, 18–20 June 2014*; pp. 1–5.
- [28] Stricker, M.A.; Orengo, M. Similarity of color images, in *IS&T/SPIE's Symposium on Electronic Imaging: Science & Technology*; International Society for Optics and Photonics: Bellingham, WA, USA, 1995; pp. 381–392.
- [29] Blei, M.D.; Ng, A.Y.; Jordan, M.I. Latent dirichlet allocation. *J. Mach. Learn. Res.* 2003, 3, 993–1022.
- [30] Chen, C.; Zhang, B.; Su, H.; Li, W.; Wang, L. Land-use scene classification using multi-scale completed local binary patterns. *Signalimageand Video Process.* 2015, 10, 745–752. [CrossRef]
- [31] Haralick, R.M.; Shanmugam, K.; Dinstein, I.H. Textural features for image classification. *IEEE Trans. Syst. Man Cybern.* 1973, 610–621. [CrossRef]
- [32] Kusumaningrum, R.; Wei, H.; Manurung, R.; Murni, A. Integrated visual vocabulary in latent dirichlet allocation-based scene classification for ikonos image. *J. Appl. Remote Sens.* 2014, 8, 083690. [CrossRef]
- [33] Bosch, A.; Zisserman, A.; Muñoz, X. Scene classification via plsa. In *Proceedings of the European Conference on Computer Vision, Graz, Austria, 7–13 May 2006*; pp. 517–530.

- [34] Zhong, Y.; Cui, M.; Zhu, Q.; Zhang, L. Scene classification based on multi-feature probabilistic latent semantic analysis for high spatial resolution remote sensing images. *J. Appl. Remote Sens.* 2015, 9, 095064.
- [35] N. L. Tun, A. Gavrilov, and N. M. Tun, "Multi-classification of satellite imagery using fully convolutional neural network," *Proc.- 2020 Int.Conf. Ind. Eng. Appl. Manuf. ICIEAM 2020*, pp.7–11, 2020, doi:10.1109/ICIEAM48468.2020.9111928
- [36] A. Van Etten, "Satellite imagery multiscale rapid detection with windowed networks," *Proc. - 2019 IEEE Winter Conf. Appl. Comput. Vision, WACV 2019*, pp. 735–743, 2019.
- [37] Nogueira, K.; Penatti, O.A.B.; dos Santos, J.A. Towards better exploiting convolutional neural networks for remote sensing scene classification. *Pattern Recognit.* 2017, 61, 539–556.
- [38] F. Hu, X. Tong, G.-S. Xia, and L. Zhang, "Delving into deep representations for remote sensing image retrieval," in *Signal Processing (ICSP), 2016 IEEE 13th International Conference on*, 2016, pp. 198–203.
- [39] T. Ishii, R. Nakamura, H. Nakada, Y. Mochizuki, and H. Ishikawa, "Surface object recognition with CNN and SVM in Landsat 8 images," in *Machine Vision Applications (MVA), 2015 14th IAPR International Conference on*, 2015, pp. 341–344.
- [40] A. Kausar, R. Javaid, N. I. Rao, and M. J. Khan, "Automatic recognition of isolated airstrips in multiscale satellite images using radon transformation and support vector machine," in *Software Engineering Conference (NSEC), 2015 National*, 2015, pp. 55– 60.
- [41] Lowe, D.G. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.* 2004, 60, 91–110.
- [42] Yang, Y.; Newsam, S. 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*, San Jose, CA, USA, 4–6 November

2010; pp.270–279.

[43] Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; et al. ImageNet Large Scale Visual Recognition Challenge. *Int. J. Comput. Vis.* 2015, 115, 211–252.

[44] Nogueira, K.; Penatti, O.A.B.; dos Santos, J.A. Towards better exploiting convolutional neural networks for remote sensing scene classification. *Pattern Recognit.* 2017, 61, 539–556.

[45] Yang, Y.; Newsam, S. 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*, San Jose, CA, USA, 4–6 November 2010; pp. 270–279.

[46] Xia, G.-S.; Yang, W.; Delon, J.; Gousseau, Y.; Sun, H.; Mantre, H. Structural high-resolution satellite image indexing. In *Proceedings of the ISPRS TC VII Symposium-100 Years ISPRS 38*, Vienna, Austria, 5–7 July 2010; pp. 298–303.

[47] Jia, Y.; Shelhamer, E.; Donahue, J.; Karayev, S.; Long, J.; Girshick, R.; Guadarrama, S.; Darrell, T. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the ACM International Conference on Multimedia*, Orlando, FL, USA, 3–7 November 2014; pp. 675–678.

[48] Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. *arXiv* 2014, arXiv:1409.4842,.

[49] Xia, G.S.; Hu, J.; Hu, F.; Shi, B. AID: A Benchmark Data Set for Performance Evaluation of Aerial Scene Classification. *IEEE Trans. Geosci. Remote Sens.* 2017, 55, 3965–3981. [CrossRef]

[50] Cheng, G.; Han, J.; Lu, X. Remote Sensing Image Classification: Benchmark and State of the Art. *Proc. IEEE* 2017, 105, 1865–1883.

APPENDIX

A. CODE

Packages Installations:

```
!pip install opencv-python
!pip3 install tensorflow==2.2
!pip3 install keras==2.3.1
```

Importing Packages and Dataset:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import cv2
import datetime
import seaborn as sns
import os
import tifffile as tiff
wkt_df = pd.read_csv('train_wkt_v4/train_wkt_v4.csv')
len(wkt_df['ImageId'].unique()) #Unique Images
wkt_df[wkt_df['MultipolygonWKT'] != 'MULTIPOLYGON
EMPTY'][['ClassType']].value_counts().plot.bar(title = 'frequency of class lable in
images')
plt.xlabel('class label')
plt.ylabel('frequency')
plt.show()

from shapely import wkt, affinity
from shapely.wkt import loads

import warnings
warnings.filterwarnings('ignore')
```

Resizing and processing of Images:

```
# 3-Band images
```

```
def stretch2(band, lower_percent=2, higher_percent=98):
    a = 0 #np.min(band)
    b = 255 #np.max(band)
    c = np.percentile(band, lower_percent)
    d = np.percentile(band, higher_percent)
    out = a + (band - c) * (b - a) / (d - c)
    out[out<a] = a
    out[out>b] = b
    return out

def adjust_contrast(x):
```

```

for i in range(3):
    x[:, :, i] = stretch2(x[:, :, i])
return x.astype(np.uint8)

def display_img(ImageID):
    #Read 3-band image
    rgbfile=os.path.join( 'three_band', '{}.tif'.format(ImageID))
    rgb = tiff.imread(rgbfile)
    rgb = np.rollaxis(rgb, 0, 3)

    x = adjust_contrast(rgb).copy()
    ax[1].imshow(x,extent=[0, 0.0092, -0.009, 0])
    pol = loads(wkt_df['MultipolygonWKT'][11]) #storing one multipolygon in pol
    wkt_df['Multipolygons'] = wkt_df.apply(lambda a: loads(a.MultipolygonWKT),
axis=1)
wkt_df['Num_Multipolygons']=wkt_df.apply(lambda a: len(a['Multipolygons'].geoms),
axis=1) #https://stackoverflow.com/questions/38930192/how-to-extract-polygons-
from-multipolygons-in-shapely
objects_per_image = wkt_df.pivot(index='ClassType', columns='ImageId',
values='Num_Multipolygons')

```

Plotting Polygons for images and Comparing with original images:

```

c=0
for img_id in wkt_df['ImageId'].unique():
    df = wkt_df[wkt_df['ImageId']==img_id]
    fig, ax = plt.subplots(1, 2,figsize=(20, 10)) #increase figsize and run again
    for ele1 in df['Multipolygons']:
        for ele2 in ele1:
            x,y = ele2.exterior.xy
            ax[0].plot(x,y,'red')
            ax[0].axis('off')
    display_img(img_id)
    #ax[0].title(f"Image {wkt_df['ImageId'][i]}")
    ax[0].set_title(f"Image {img_id}")
    ax[1].set_title(f"Image {img_id}")
    ax[1].axis('off')
    plt.show()
    c += 1
    if c==3:
        break

```

Class labels in Dictionary as key value pair:

```

class_label = {
    1:'Buildings',
    2:'Misc. Manmade structures',
    3:'Road',
    4:'Track',
    5:'Trees',
    6:'Crops',

```

```

7:'Waterway',
8:'Standing water',
9:'Vehicle Large',
10:'Vehicle Small'
}

```

Adjust contrast of Images and display Images:

```

def stretch2(band, lower_percent=2, higher_percent=98):
    """
    Adjusting the contrast
    """
    a = 0 #np.min(band)
    b = 255 #np.max(band)
    c = np.percentile(band, lower_percent)
    d = np.percentile(band, higher_percent)
    out = a + (band - c) * (b - a) / (d - c)
    out[out<a] = a
    out[out>b] = b
    return out

def adjust_contrast(x):
    for i in range(3):
        x[:, :, i] = stretch2(x[:, :, i])
    return x.astype(np.uint8)

def display_img(ImageID, ax):
    """
    Displaying images
    """
    rgbfile=os.path.join( 'three_band', '{}.tif'.format(ImageID))
    rgb = tiff.imread(rgbfile)
    rgb = np.rollaxis(rgb, 0, 3)
    x = adjust_contrast(rgb).copy()
    ax.imshow(x, extent=[0, 0.0092, -0.009, 0])

```

Plotting polygon class wise separately:

```

def mask_classes(imgID):
    """
    getting polygon coordinate and plotting the polygon class wise separately
    """
    df = wkt_df[wkt_df['ImageID']==imgID]
    for class_l in df['ClassType']:
        fig, ax = plt.subplots(figsize=(5, 5))
        df1 = df[df['ClassType']==class_l]
        display_img(imgID, ax)
        for ele in df1['Multipolygons']:
            for ele2 in ele:
                x,y = ele2.exterior.xy
                ax.plot(x,y,'red')

```

```

plt.title(f"Class Label :- {class_label[class_l]}")
plt.axis('off')
plt.show()
mask_classes('6120_2_2')

```

Plotting Heatmap of Class Frequency Vs Images:

```

figure, axis = plt.subplots(figsize=(10, 8))
axis.set_aspect('equal') #keep x-unit and y-unit ratio equal
plt.imshow(objects_per_image, cmap='BuPu', extent=[0, 25, 10, 0])

"""for i in range(objects_per_image.shape[0]):
    for j in range(objects_per_image.shape[1]):
        plt.text(j+0.5, i+0.5, "{:.2f}".format(objects_per_image.iloc[i,j]), ha="center",
va="center",fontsize=16)"""

plt.xticks(np.arange(0, 25, 1.0),list(objects_per_image.columns),
rotation='vertical',fontsize=16)
plt.yticks(np.arange(1, 11, 1.0),fontsize=18)
plt.title('Number of objects per image',fontsize=18)
plt.xlabel('Images',fontsize=18)
plt.ylabel('Classes',fontsize=18)
plt.colorbar()
plt.show()

Classes = {1: 'Building', 2: 'Structure', 3: 'Road', 4: 'Track', 5: 'Trees', 6: 'Crops',
7: 'Waterway', 8: 'Standing water', 9: 'Truck', 10: 'Car'}
Colors = {1:'#d73027' , 2: '#f46d43', 3: '#4575b4', 4:'#74add1' , 5: '#dfc27d', 6:
'#a6dba0', 7: '#1b7837', 8: '#b35806', 9: '0.4', 10: '0.7'}

```

Creating dataframe containing classes, number of polygon as counts and total area of image

```

def image_stats(image_id):
    """
    creating dataframe which contains
    classes, number of polygon as counts and total area of image with id
    image_id
    """
    counts, total_area = {}, {}
    xmax = GS[GS.ImageId == image_id].Xmax.values[0]
    ymin = GS[GS.ImageId == image_id].Ymin.values[0]
    image_area = abs(xmax * ymin) #area of image
    for cl in Classes:
        all_poly = wkt_df[wkt_df.ImageId == image_id]
        poly = all_poly[all_poly.ClassType == cl].MultipolygonWKT
        poly_list = wkt.loads(poly.values[0])
        counts[cl] = len(poly_list) #number of polygons of classtype cl

```

```

        if len(poly_list) > 0:
            #calculating area of all polygon for class cl having number of polygons
            more than one
            total_area[cl] = np.sum([poly.area for poly in poly_list])\
                / image_area * 100

    df = pd.DataFrame({'Class': Classes, 'Counts': counts,
                      'TotalArea': total_area})
    return df

```

```

GS = pd.read_csv('grid_sizes/grid_sizes.csv')
GS = GS.rename(columns={'Unnamed: 0': 'ImageId'})

```

```

stats_list = []
Images = sorted(set(wkt_df['ImageId']))
for image_no, image_id in enumerate(Images):
    stat = image_stats(image_id)
    stat['ImageId'] = image_id
    stats_list.append(stat)

```

```

stats = pd.concat(stats_list)
pvt_stats = stats.pivot(index = 'Class', columns = 'ImageId', values =
'TotalArea')
percent_area = np.cumsum(pvt_stats, axis = 0)

```

```

class_r = {}
for cl in Classes:
    class_r[Classes[cl]] = cl

```

Plot Bar plot iterating through all classes:

```

for cl in np.arange(1, 11):
    class_name = percent_area.index[-cl]
    class_id = class_r[class_name]
    ax = sns.barplot(x = percent_area.columns, y =
percent_area.loc[class_name],
                    color = Colors[class_id], label = class_name)

```

```

sns.set_context({'figure.figsize': (10, 5)})
ax.legend(loc = 2)
sns.set_style("dark")
ax.set_xticklabels(percent_area.columns, rotation = 45)

```

```

ax.set_xlabel('Image ID')
ax.set_ylabel('Total Area')
plt.title('Areas of the objects in Image')
plt.show()

```

B. OUTPUT SCREENSHOTS



Fig A. Feature Extracted Output 1



Fig B. Feature Extracted Output 2



Fig C. Feature Extracted Output 3

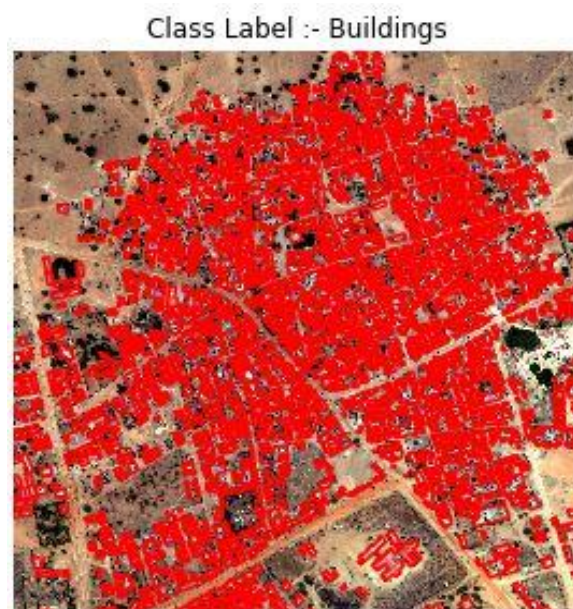


Fig D. Feature Extracted Output 4

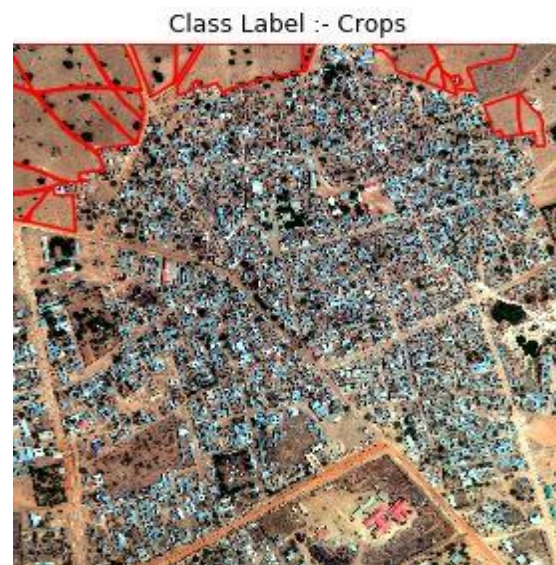


Fig E. Feature Extracted Output 5

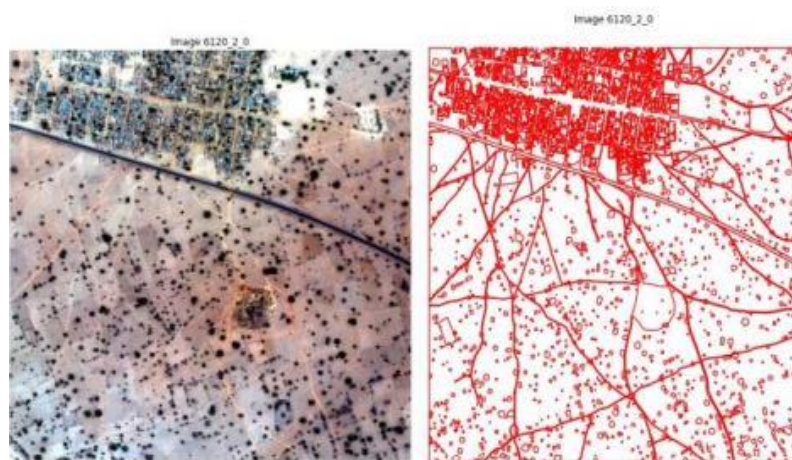


Fig F. Image Before and After Feature Extraction

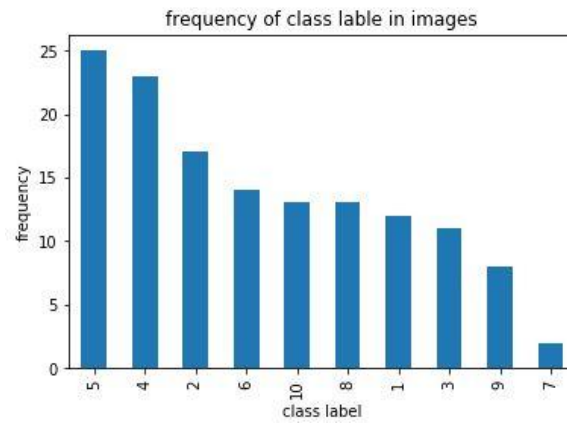


Fig G. Graph Plot 2

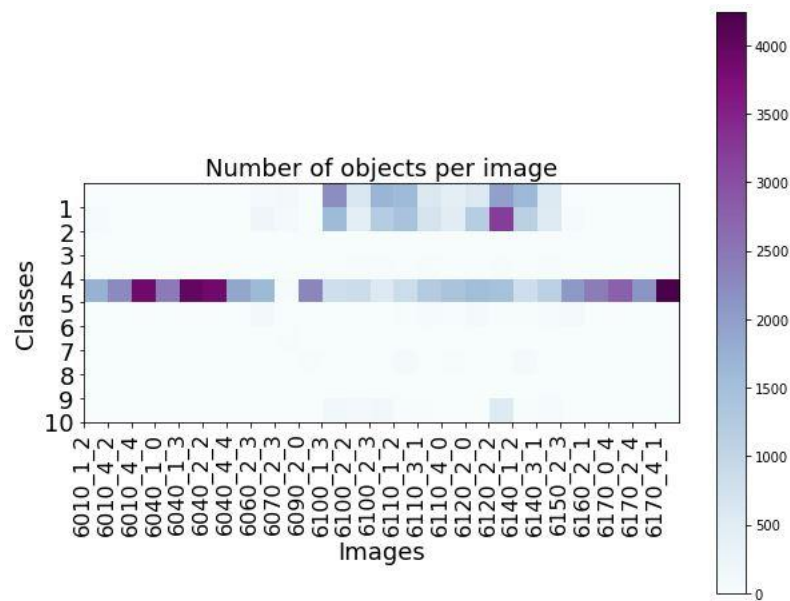


Fig H. Heat Map

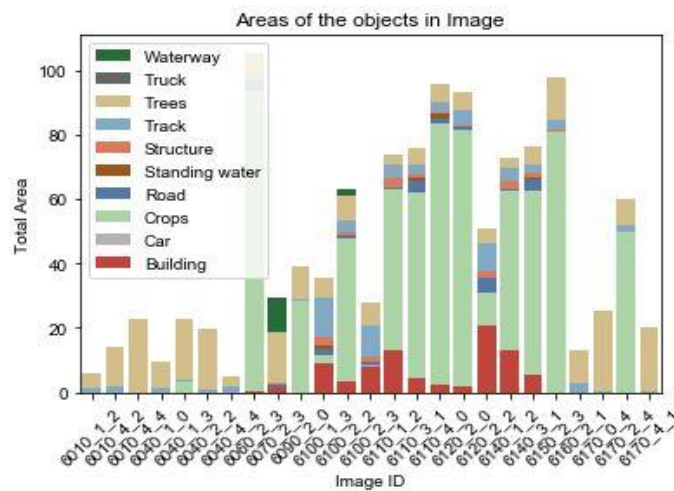


Fig I. Graph Plot 3

Automatic Feature Extraction from Satellite Imagery for Remote Sensing

D Diya Mary
U.G Scholar

*Department of Information Technology
Sathyabama Institute of Science and
Technology
Chennai, India
diyamary1576@gmail.com*

Nandana S. Nair
U.G Scholar

*Department of Information Technology
Sathyabama Institute of Science and
Technology
Chennai, India
devus7875@gmail.com*

Dr. Mohana

*Research Scientist
Centre for Remote Sensing and
Geoinformatics
Sathyabama Institute of Science and
Technology
Chennai, India
mohanaperumalvv@gmail.com*

Dr. S. Revathy

*Associate Professor
Department of Information Technology
Sathyabama Institute of Science and
Technology
Chennai, India
revathy.it@sathyabama.ac.in*

Dr. L. Mary Gladence

*Associate Professor
School of Computing
Sathyabama Institute of Science and
Technology
Chennai, India
marygladence.it@sathyabama.ac.in*

Bernatin T

*Assistant Professor
Department of ECE
Sathyabama Institute of Science and
Technology
Chennai, India
bernatin12@gmail.com*

Abstract— Feature extraction from satellite images has steadily but progressively grown in recent years. Numerous feature extraction techniques have been created as a result of the increase. The loss of certain features is one of the various obstacles that must be overcome while selecting the best path for feature extraction. This study suggests utilizing a Convolutional Neural Network (CNN) approach to tackle these issues and extract attributes from satellite images supplied by an open source dataset called MLRSNet. Images with labels explaining each image's characteristics are displayed in the output. This makes it simpler to recognize and understand different components in satellite images.

Keywords— *Feature extraction, Satellite Imagery, Neural Network, Image Processing*

variables into features, feature extraction assists to substantially cut down the amount of data by extracting the critical qualities from all those large data sets. These features properly and clearly portray the actual data set while being simple to use.

The feature extraction strategy may be useful when you possess a large amount of data and wish to decrease resources without compromising any critical or relevant information. Using feature extraction, the amount of redundant data in the data harvest is reduced. Finally, data reduction accelerates the training and generalization stages of the machine-learning procedure and allows the model to be developed with little effort from the machine

I. INTRODUCTION

Over the years, the area of feature extraction has widened significantly. For this, satellite photos are frequently employed and significantly improve outcomes. There are several items in images used for feature extraction, all of which must be precisely recognised for effective outcomes. There are various techniques to make it simpler to extract characteristics from an image for processing. Semantic scene classification is one of the activities that helps with picture categorization and retrieval. There have been researches [1] which sheds information on the process of extracting different pictures using CNN from high resolution satellite images. This work used several datasets and powerful CNNs, which made it easier to see advancements over the current method.

To extract the features, a big initial set of data should be split into smaller, more manageable groups. As a result, processing will be simplified. The notion that these large data sets encompass a wide range of diverse characteristics is their most notable aspect. A huge amount of computing power is needed to manage these variables. By selecting and merging

II. LITERATURE SURVEY

S. Muhammad et al. published a Supervised Satellite Data Classification method in [2] that makes use of the Decision Tree Approach. This method obtains features from satellite images which relies on pixel color and intensity. With the use of known items, the approach employs a decision tree to categorize satellite pictures.

Classification Methods to Extract Areas from Satellite Images [3] describes the most well-liked unsupervised satellite classification algorithm. It defines a collection of unlabeled classes within a picture before naming the clusters.

Complete local binary patterns alongside multi-scales were introduced by the authors of "Land-use scene categorization using multi-scale completed local binary patterns" [4] and demonstrated to be more successful than earlier low-level techniques.

The use of the sparse coding method outlined in High-resolution Satellite Scene Categorization Using a Sparse Coding-Based Multiple Feature Combination [5], where structural, spectral, and textural characteristics were extracted and encoded.

In his article Spatial methodology for Image Classification [6], Selim proposed a classification method utilizing the Bayesian approach. This method takes use of spatial information to categorize very high-resolution satellite images. The categorizing procedure involves two steps. Bayesian classifiers are trained using discontinuous nonparametric density models in phase 1. The spectral and textural characteristics of each pixel are extracted. Phase 2: Using an iterative split-and-merge procedure, the pixel-level classification maps are converted into continuous areas.

The Support Vector Machine is an unsupervised classification technique [7]. It may be used to extract land maps. This strategy, however, presupposes that no knowledge on how to disseminate the complete quantity of data exists. SVM increases accuracy, speed, and efficiency while lowering the cost of satellite categorization. The best strategy for identifying patterns is a supervised statistical approach. It places pixels in the appropriate classes by using their probability values [8].

Global characteristics were extracted by Nogueira [9] from CNN models and supplied to a classifier. This method has been used to show the effectiveness of object classification models in transfer learning. These local or global extracted feature representations were derived using CNNs that had been trained beforehand on datasets of real-world images in each of these examples. Remote sensing-derived pictures were categorized using these characteristics.

Surface object recognition using CNN and SVM in Landsat 8 pictures [10] performs surface item detection using large collections of image data. We provide two methods which rely on CNN and SVM and compare them. In the trials, CNN outperformed other options.

Automatic recognition of isolated airstrips using radon transformation and support vector machine [11] employs the SVM technique after preprocessing the satellite images to find solitary airstrips. The algorithm had an efficient result and obtained accuracy which estimated 94%. It recognized airfields as well as other common linear features like roads, canals, and so on.

However, as mentioned in [11], rigorous training of CNNs on the UC-Merced dataset [12] and the WHU-RS19 dataset [13] leads to mediocre classification performance. The cause for inefficient performance and generalization is because huge CNN architectures, which consist of large number of factors, always cause model training utilizing aerial scene datasets which have thousands of pictures to get stuck in the local minima.

III. EXISTING WORK

In order to monitor Earth resources and the effects of human activities on the planet with finer spatial and spectral resolution and give more accurate and precisely detailed information, it is currently not very effective to use satellite and airborne remote sensing technologies. Urban regions are undergoing changes. The need for current large-scale data is therefore urgent for a number of reasons. Feature extraction is a significant step in the construction of maps for many countries. Additionally, it is usually impossible to find obsolete maps, and automated methods have been suggested to hasten and improve the efficiency of the map generating process. However, the process for updating this data is now in a slow state.

For the feature extraction procedure, several different strategies have been suggested. Automatic object identification is one common method for extracting features, however it still has several flaws. Automatic line identification systems are challenging to build due of the complexity of remotely sensed images. Consequently, a variety of statistical methods have been widely used to classify data on land use or land cover. Researchers' attention has switched to cutting-edge, dependable categorization methods like Artificial Neural Networks as satellite sensors have been created.

Artificial neural networks, or ANNs, were first developed as tools for pattern recognition and data analysis that mimicked the neural processing and storage capabilities of the human brain. High spatial resolution remote sensing data usually contain several diverse terrain patches and complex non-linear patterns. Outstanding benefits of ANNs include their capability for rapid generalisation, parallel processing, and the ability to predict the non-linear connection between input data and desired outputs. Numerous past studies have shown that ANNs outperform traditional classification methods like maximum likelihood classifiers on multispectral data. Avoiding the use of distinct operators such as simplification, smoothing, and enhancement.

It can be challenging to avoid losing low- and mid-level information as a convolution neural network grows increasingly complicated since the majority of convolution neural networks currently in use when used for scene categorization have a propensity to portray the global semantics of pictures. To solve these problems, we suggest using Res-Net for feature extraction. In order to further investigate local semantics, it sees the categorization of aerial images as a multiple-instance learning problem. [17]

Following that, the acquired convolution features are modified to create instance feature vectors. Following that, we provide trainable attention-based multiple instance pooling. It computes the bag-level probability rapidly and shows the local semantics relevant to the scene label. Finally, the multiple instance learning architecture is supervised by bag labels due to our bag-level classification layer.

Modern CNNs, on the other hand, prefer to preserve global semantics as objects in actual image contexts are much more dense and possess more robust context linkages. Concentrating on generic semantics may render it more difficult to notice distinguishing elements in an aerial viewpoint and may make representing the scene more difficult.

IV. PROPOSED WORK

A. Methodology

The process's initial stage is to collect the dataset. In this case, we employed the MLRSNet dataset[16], a remote sensing dataset with a focus on visualization. The MLRSNet comprises 46 categories and 109,161 samples in total. Fig. 1 displays the number of samples in each group. Table 1 displays all of the categories along with the number of photos for each category. The pictures size is 256 by 256 and have a spatial resolution of 10–0.1 m. The photographs are then scaled, cropped, and brightness and contrast modifications are applied as part of data augmentation. The photos were then labelled and placed into specified classes. For instance, as shown in Fig. 2, photographs of airport scenes are sorted into one class while those of residential regions are placed into another.

Following the preparation of the dataset, training is carried out using RESNET50 in batches of 40. Resnet50 has over 23 million parameters and Keras package was used for the model. After that comes the testing step, where 180×180 photos are tested in a single batch. 50 layers make up the convolutional neural network that it is. Finally, Convolutional Neural Networks are used to segment the images (CNN). [18]

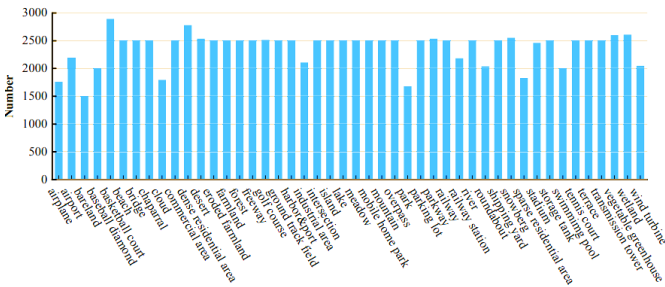


Fig 1. Representation of number of image samples in every category.

TABLE 1: Images in the dataset for every class label.

Class Label	Number	Class Label	Number	Class Label	Number
Airplane	2306	Freeway	2500	Roundabout	2039
Airport	2480	Golf course	2515	Runway	2259
Bare soil	39,345	Grass	49,391	Sand	11,014
Baseball diamond	1996	Greenhouse	2601	Sea	4980
Basketball court	3726	Gully	2413	Ships	4092
Beach	2485	Habor	2492	Snow	3565
Bridge	2772	Intersection	2497	Snowberg	2555
Buildings	51,305	Island	2493	Sparse residential area	1829
Cars	34,013	Lake	2499	Stadium	2462
Chaparral	5903	Mobile home	2499	Swimming pool	5078
Cloud	1798	Mountain	5468	Tanks	2500
Containers	2500	Overpass	2652	Tennis court	2499
Crosswalk	2673	Park	1682	Terrace	2345
Dense residential	2774	Parking lot	7061	Track	3693
Desert	2537	Parkway	2537	Trail	12,376
Dock	2492	Pavement	56,383	Transmission tower	2500
Factory	2667	Railway	4399	Trees	70,728
Field	15,142	Railway station	2187	Water	27,834
Football field	1057	River	2493	Wetland	3417
Forest	3562	Road	37,783	Wind turbine	2049



Fig 2. The images show residential areas and airport scenes.

Figure 3 depicts the flow of the suggested methodology. The procedure starts with satellite pictures that go through image scaling and data augmentation during pre-processing. Data augmentation includes translating certain images to create more data from existing data. The phase of preprocessing improves quality of data and enhances performance which assists in an efficient training and further processes. Convolutional neural network, which contains four stages, is the next step. The first step is to use the Res-Net architecture to extract the image's features. The next step is dimensionality reduction, which lowers complexity. For reducing the dimensionality, we have used the principal component

analysis which helps in preserving most of the information. The classification is then completed using the characteristics that were extracted. The system's output are the photos that have been categorised.

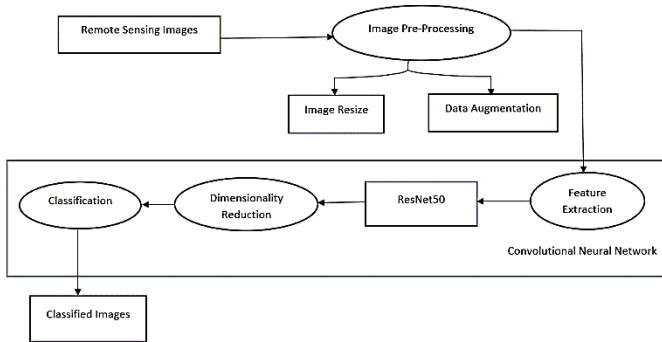


Fig 3. System Architecture.

The RESNet50 architecture-based CNN algorithm is trained on the pictures. As seen in Fig. 4, the pictures pass through a completely linked layer. In a fully linked layer, every neuron within the input layer is coupled to every neuron in the output layer. Before the output is subjected to a nonlinear activation function at this layer, the input neurons undergo a weighted linear transformation. The hash layer, which comes before processing and hashes and organizes category characteristics, comes after the fully connected layer. This layer converts categorical inputs into hashed outputs. It converts a predetermined range of ints or strings to ints element-by-element.

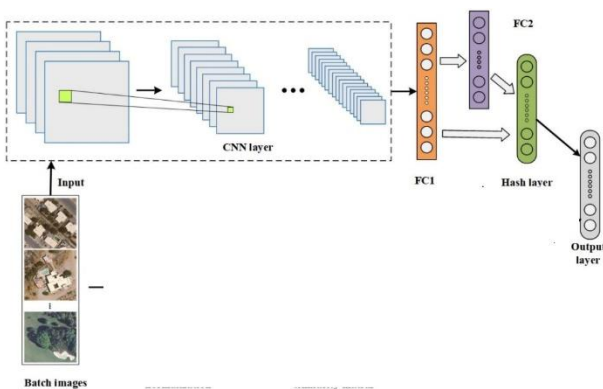


Fig 4. Working of the model.

B. RESNet50

Residual Networks, train residual functions with regard to the layer inputs rather than learning unsourced functions. Rather than expecting each layer in a stack to exactly match the desired underlying mapping, residual nets allow these layers to fit a residual mapping. Remaining blocks are stacked one over the other to build networks; a ResNet50 employs these blocks to have fifty layers (48 convolutional layers, one average pool layer and one MaxPool layer). Residual neural

networks (RNNs) are artificial neural networks (ANNs) which construct networks using residual blocks.

The ResNet50 architecture abides by two key design principles. First, there are the same number of filters in each layer regardless of size if the output feature map. Second, although though size of the feature map is reduced by half, it contains twice as many filters in order to maintain the temporal complexity of each layer.

Issues about the overall time needed to train the layers concluded to the decision of redesigning the building block in this case as a bottleneck. This used a stack of three layers in place of the previous two levels. Each of the two-layer blocks in Resnet34 was converted into a three-layer bottleneck block to form the Resnet 50 architecture. This is far more accurate than the 34-layer ResNet model..

ResNet trains on F whereas every other algorithm learns from the result "Y" (X). Another way to say it is that ResNet aims to make $F(X)=0$ such that $Y=X$. Some model levels are skipped by a direct link known as a skip connection. The output differs as a result of this skip link. In the absence of a skip connection, input 'X' is multiplied by the weights of the layer before the bias term has been introduced. It is followed by activation function, $F()$, and its output can be given as follows:

$$F(w * x + b) (=F(X))$$

The output with use of skip connection technique, is :

$$F(X)+x$$

In ResNet-50, there are two different types of blocks: Identity Blocks and Convolutional Blocks. When input size equals output size, the value "x" is added to the output layer. In the event that the case is different, a convolutional block is added to the shortcut path to equalise the input and output sizes.

C. Convolutional Neural Networks

The Convolutional Neural Network is just a technique under deep learning that has the ability to take in an input image, discriminate between various items and components in the image, and allocate different objects and elements in the image importance. A CNN needs a lot less pre-processing than different categorization techniques. While filters in simple approaches are hand-engineered, CNN has the ability to learn these filters' qualities. It is CNN's responsibility to streamline the graphics without sacrificing any of the information that is necessary for making precise forecasts. This is critical for creating a design that is constructive at learning features while also being scalable to large datasets.

The convolution operation's main goal is to obtain high-level information from the input image, like edges. CNN does not have to employ just one convolutional layer. Typically, the first ConvLayer collects low-level properties such as edges, colour, and gradation direction. Even as number of layers

risers, the architecture adjusts to the High-Level characteristics, resulting in a network that encompasses the dataset's images holistically in a way similar to what we do.

Two alternative sorts of outcomes may be obtained from the procedure: one where the dimensionality of the convolved feature is narrowed with respect to the input, and another where it is increased or remains the same. This is achieved by utilizing Valid Padding or Same Padding.

Like the Convolutional Layer, the Pooling layer is responsible for decreasing the size of the Convolved Feature. The aggregate of resources required to process the data will be subsided by dimensionality reduction. Apart from this, it facilitates appropriate model training by obtaining dominant traits which are rotational and positional invariant.

The i -th layer of a CNN is made up of the convolutional layer as well as the pooling layer. Depending on how complex the images are, the layer count may be increased to concentrate on an even finer details, but this will require more processing resources. Max Pooling and Average Pooling are two types of pooling. Max Pooling returns the greatest value from region of the image covered by the Kernel. Average pooling, on the other hand, delivers the mean of all of the values from portion of the image encompassed by the Kernel. Max Pooling additionally acts as a noise reduction. It accomplishes de-noising along with dimensionality reduction in addition to thoroughly removing noisy activations. After working through method mentioned above, we were capable of assisting the model in understanding the properties. After that, the output will be flattened for classification before even being loaded into a conventional neural network.

V. RESULTS AND DISCUSSIONS

The results are displayed in the photos below, which highlight the characteristics that were taken out of each image. Along with the highlighted elements, the label that explains the scenario in the photograph is also displayed. The accuracy achieved by this methodology is 92%. Additionally, performance was evaluated for the CNN algorithm used which resulted in an accuracy of 87%. Figure 5's highlighted buildings demonstrate the region's dense population. The principal thoroughfares in the region that link the city's ends are shown in Fig. 6. In fig. 7, the trees have been removed to highlight the region's lack of vegetation. Additionally, fig. 8 illustrates the roadways that link the area's many locations and demonstrates how well connected it is. Figure 9 depicts the picture before and after the extraction; the characteristics are emphasised in the latter image. The area that each of them takes up in a picture is depicted in the graph in fig. 10. Overall, this model facilitates the visual analysis of images and the comprehension of their aspects.

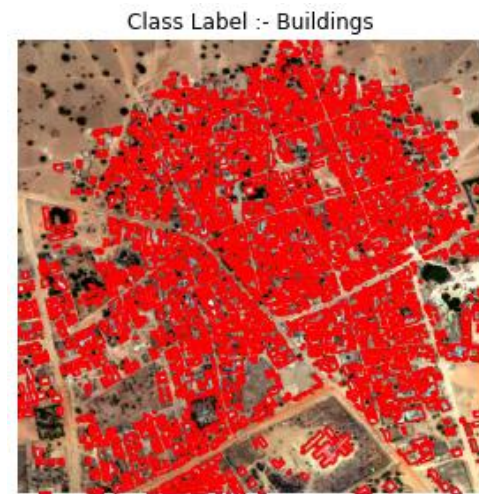


Fig 5. Image highlighting the residential area.



Fig 6. Image with roads.



Fig 7. Image with trees in the area.

Class Label :- Track



Fig 8. Image with streets in an area.

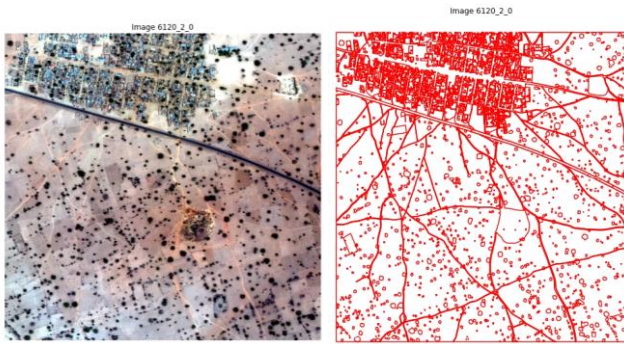


Fig 9. The image before and after highlighting of features.

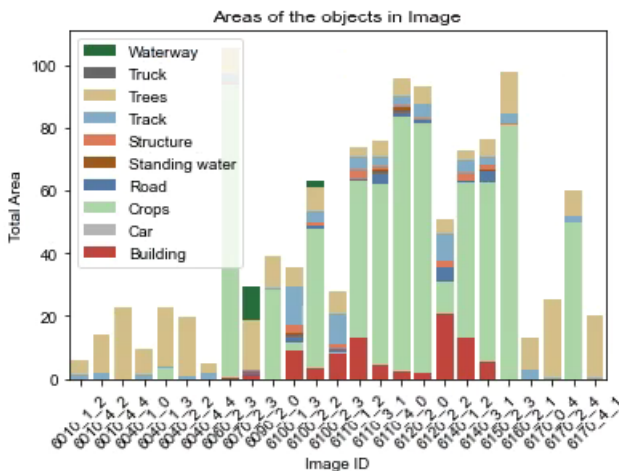


Fig 10. Graph of areas covered in an image.

VI. CONCLUSION

The MLRSNet is a dataset with high resolution images which enables deep learning to understand semantic scene interpretation, at least from an overhead perspective. Hierarchy, scalability, diversity, and multi-label are the

characteristics that distinguish MLRSNet. Images with labels identifying the category to which they belong are extracted. Each image's particular traits are accentuated to make them stand out. These results assist in examining the visuals from a fresh angle, which may further aid in effective comprehension. This model can be further developed in future to gain more accuracy and also expand its working on larger satellite images. Larger images will require more intensive training and the algorithm will have to be modified to work with it. The pictures make good use of the characteristics that were retrieved, making it easier to analyze a satellite image.

REFERENCES

- [1] F. Hu, X. Tong, G.-S. Xia, and L. Zhang, "Delving into deep representations for remote sensing image retrieval," in *Signal Processing (ICSP)*, 2016 IEEE 13th International Conference on, 2016, pp. 198–203.
- [2] Muhammad, S., Aziz, G., Aneela, N. and Muhammad, S. 2012. "Classification by Object Recognition in Satellite Images by using Data Mining". In *Proc. Proceedings of the Congress on Engineering (WCE 2012)*, Vol I, July 4 - 6, London, U.K.
- [3] Al-Ahmadi, F. S. and Hames, A. S. 2009. "Comparison of Four Classification Methods to Extract Land Use and Land Cover from Raw Satellite Images for Some Remote Arid Areas, Kingdom of Saudi Arabia", *Journal of King Abdulaziz University-Earth Sciences*, Vol. 20, No.1, pp: 167-191.
- [4] Chen, C.; Zhang, B.; Su, H.; Li, W.; Wang, L. Land-use scene classification using multi-scale completed local binary patterns. *Signal image and Video Process.* 2015, 10, 745–752.
- [5] Sheng, G.; Yang, W.; Xu, T.; Sun, H. High-resolution satellite scene classification using a sparse coding-based multiple feature combination. *Int. J. Remote Sens.* 2012, 33, 2395–2412.
- [6] Selim Aksoy. 2006. "Spatial Techniques for Image Classification," in C. H. Chen, ed., *Signal and Image Processing for Remote Sensing*, CRC Press, pp.491- 513.
- [7] N. L. Tun, A. Gavrilov, and N. M. Tun, "Multi-classification of satellite imagery using fully convolutional neural network," *Proc.-2020 Int.Conf. Ind. Eng. Appl. Manuf. ICIEAM 2020*, pp.7–11, 2020, doi:10.1109/ICIEAM48468.2020.9111928
- [8] A. Van Etten, "Satellite imagery multiscale rapid detection with windowed networks," *Proc. - 2019 IEEE Winter Conf. Appl. Comput. Vision, WACV 2019*, pp. 735–743, 2019.
- [9] Nogueira, K.; Penatti, O.A.B.; dos Santos, J.A. Towards better exploiting convolutional neural networks for remote sensing scene classification. *Pattern Recognit.* 2017, 61, 539–556.
- [10] T. Ishii, R. Nakamura, H. Nakada, Y. Mochizuki, and H. Ishikawa, "Surface object recognition with CNN and SVM in Landsat 8 images," in *Machine Vision Applications (MVA)*, 2015 14th IAPR International Conference on, 2015, pp. 341–344.
- [11] A. Kausar, R. Javaid, N. I. Rao, and M. J. Khan, "Automatic recognition of isolated airstrips in multiscale satellite images using radon transformation and support vector machine," in *Software Engineering Conference (NSEC), 2015 National*, 2015, pp. 55– 60.
- [12] Luo, B.; Jiang, S.; Zhang, L. Indexing of remote sensing images with different resolutions by multiple features. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* 2013, 6, 1899–1912.
- [13] Lowe, D.G. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.* 2004, 60, 91–110.
- [14] Yang, Y.; Newsam, S. 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*, San Jose, CA, USA, 4–6 November 2010; pp.270–279.
- [15] Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; et al. ImageNet Large Scale Visual Recognition Challenge. *Int. J. Comput. Vis.* 2015, 115, 211–252.
- [16] X. Qi *et al.*, "MLRSNet: A multi-label high spatial resolution remote sensing dataset for semantic scene understanding," *ISPRS Journal of*

Photogrammetry and Remote Sensing, vol. 169, pp. 337–350, Nov. 2020, doi: 10.1016/j.isprsjprs.2020.09.020.

- [17] Q. Bi, K. Qin, Z. Li, H. Zhang, K. Xu, and G. S. Xia, “A Multiple-Instance Densely-Connected ConvNet for Aerial Scene Classification,” *IEEE Transactions on Image Processing*, vol. 29, pp. 4911–4926, 2020, doi: 10.1109/TIP.2020.2975718.
- [18] B. Petrovska, E. Zdravevski, P. Lameski, R. Corizzo, I. Štajduhar, and J. Lerga, “Deep learning for feature extraction in remote sensing: A case-study of aerial scene classification,” *Sensors (Switzerland)*, vol. 20, no. 14, pp. 1–22, Jul. 2020, doi: 10.3390/s20143906.
- [19] Han, W., Feng, R., Wang, L., Cheng, Y., 2018. A semi-supervised generative framework with deep learning features for high-resolution remote sensing image scene classification.
- [20] Cheng, G., Han, J., Zhou, P., Guo, L., 2014. Multi-class geospatial object detection and geographic image classification based on collection of part detectors. *ISPRS J. Photogramm. Remote Sens.* 98, 119–132.