Library Progress International
Vol.43, No.2, Jul-Dec 2023: P.2048-2068

Print version ISSN 0970 1052
Online version ISSN 2320 317X

Original Article

Available online at www.bpasjournals.com

Feature Extraction for Remote Sensing Image Classification using Variants of Deep Learning Pre-trained Models Densenet-169, Densenet-121 and Densenet-201

Nisha Gupta[1], Ajay Mittal[2], and Satvir Singh[3]

[1]Department of Computational Sciences, MRSPTU Bathinda, India.
*nishasbs2019@gmail.com
[2]Department of Applied Sciences, Aryabhatta Group of Institutes, Barnala, India.
*ajay11mittal@gmail.com
[3]Department of Electronics and Communication, IKGPTU Kapurthala, India.
*drsatvir.in@gmail.com
*Address correspondence to: nishasbs2019@gmail.com

How to cite this article: Nisha Gupta, Ajay Mittal, and Satvir Singh (2023). Feature Extraction for Remote Sensing Image Classification using Variants of Deep Learning Pre-trained Models Densenet-169, Densenet-121 and Densenet-201. *Library Progress International,* 43(2), 2048-2068

Abstract: *Accurately recognizing the objects in an image is known as image classification. The effective classification of high-resolution spatial images for extensive remote sensing archives is known as remote sensing image classification. Efficient extraction of features from images is directly related to high classification performance. Prior to deep learning being widely used in the field of remote sensing picture classification, the majority of feature extraction stages relied on manually created low level features, mostly concentrating on fundamental aspects like colour, form, and texture. However, CNNs (Convolution Neural Networks), which effectively extracted abstract information, soon supplanted these conventional handcrafted methods due to their poorer performance. Significant training restrictions affect deep convolution neural networks, including pre-trained models created on the enormous ImageNet dataset for the main objective of image classification and picture retrieval. These limitations must be appropriately considered during the training phase. This paper attempts to simplify all of the crucial elements that need to be considered while training an extremely deep neural network so that the model can produce the best classification results. In order for the model to generate optimal classification results, This paper examines the experiments conducted on four remote sensing datasets—UC-Merced, AID, NWPU-RESISC-45, and Patternet using the pre-trained model DenseNet and its variants, DenseNet169, DenseNet-121, and DenseNet-201. On the UC-Merced, AID, NWPU-RESISC45, and PatterNet datasets, the DenseNet pre-trained model DenseNet-201 achieved highest test accuracy of 97.60% and least Test Loss of 0.6402% respectively. This indicates that the PatterNet dataset is the most effective at classifying remote sensing images.There are several potential applications for the content-based remote sensing information retrieval system, including forestry and agricultural. With merely an aerial view, CBIR might be a huge help in agricultural regions to identify sick crops. By remotely monitoring the impacted region deforestation may be tracked.*

Keywords: Image Classification, Convolution Neural Networks, Pre-Trained Models

1 Introduction

More abstract semantic information could not be efficiently extracted by Convolution Neural Networks since they were not deep enough. Due to the need for huge data samples and high processing power, training a deep convolution neural network is costly and time-consuming (Y. Wang, Ji, and Y. Zhang 2021; Alzu'bi, Amira, and Ramzan 2017). Furthermore, when given extensive training data samples, convolution neural networks are prone to overfitting. Compared to previous Convolution Neural Networks (CNNs) approaches utilised for feature extraction, Pre-Trained models employing Transfer Learning could yield better outcomes in the future era of deep learning (J. Zhang, Jin, et al. 2020; Karimpouli and Tahmasebi 2019). Transfer Learning demonstrated its superiority in extracting features of a specific task by using learnt features in the form of weights for a broader task (Fan, Hongwei Zhao, and Haoyu Zhao 2020). Through Transfer Learning, the knowledge of a previously trained model is applied to a separate but related problem (Risojevi´c and Stojni´c 2021). Investigating training accuracy and training loss on remote sensing images through training of all DenseNet variants—DenseNet-169, DenseNet-121, and DenseNet-201—is the main topic of this paper. Because of its modest layer count, which makes it acceptable for four benchmark datasets, and its significantly higher validation accuracy when compared to other pre-trained models, the DenseNet pre-trained model has been chosen for classification in the remote sensing area. The network's output class is determined by the classification layer. Using a modified network with transfer learning, the test images are categorised. Both the fully connected layer and the classification layer are used in transfer learning. These two layers are in charge of turning networks' extracted characteristics into class probabilities and using those features to forecast labels. In accordance with the benchmark datasets UC-Merced, AID, NWPU-RESISC-45, and PatterNet, these two layers are swapped out in order to preserve a previously trained network for classifying new images with extra layers. Lastly, accuracy that is the proportion of accurately predicted labels of the network is estimated using the four benchmark datasets on the training and testing sets of images indicated above. Convolution neural networks in conjunction with a transfer learning technique have the potential to produce notable outcomes in the field of remote sensing image classification.

2 Related Literature Survey

To extract features and classify images, researchers have employed a variety of machine learning and deep learning techniques. Researchers addressed the problem of CNN training requiring large labelled samples (Yishu Liu, Ding, et al. 2020). By applying transfer learning to CNN training, the author's proposed SBS-CNN (Similarity Based Supervised Learning using Convolution Neural Network) method turns similarity learning into deep ordinal classification using CNN experts who have already been trained on large scale labelled everyday image sets. Together, it establishes visual similarity and offers fictitious classification labels. Compact feature vectors, a small network size, and a shorter retrieval time are all advantages of SBS-CNN. To reduce error, gradient descent is computed at every step. The gradient descent is computed at each step to reduce error, which is a major research concern. The ReLU (Rectified Linear Unit) activation function receives negative values and converts them to zero, which reduces the model's capacity to appropriately fit or train from data.

The author experimented using the Everyday ImageSet, UC-Merced, and PatterNet datasets; the benchmark dataset yielded the best ANMR of 0.2185.In 2020, Shen, Qian, Huang, Jiru, Zheng, Yalan, and Wang, Min introduced OSA-HSR + CNN (Object Scale Adaptive High Spatial Resolution Remote Sensing Image Classification in conjunction with Convolution Neural Networks) (J. Wang et al. 2020). Segmentation and classification are two processes that are taken. Segmentation yields heterogeneous segments, from which features are subsequently extracted. Classification is done using the features that have been extracted. Increasing segmentation scale neighbour objects and decreasing classification quality are the main research concerns. Misclassification may result from both excessive and insufficient segmentation. Run time is increased by OSA-HSR+CNN. In order to reduce error, the stochastic gradient is assessed at each stage, which increases computing time. The Ohio State-wide Imagery Program collects the aerial images used in the experiments. A computation time of 188 seconds was attained. Some researcher suggested ANNs, EfficientNet-B3-Attn-2, which use the pre-trained EfficientNet-B3 CNN with an attention mechanism added (Paoletti et al. 2018). Using remote sensed images, experiments are conducted with CNN feature extractors by building them from the ground up, using transfer learning, and optimizing them for the LCLU (Land Cover and Landuse) classification system. In the UCM dataset, the refined deep learning model produced remarkably accurate performance outcomes.Experimental results on six well-known remote sensing datasets—KSA, UC Merced, WHU-RS19, RSSCN7, OPTIMAL-31, and AID datasets—show that EfficientNet-B3-Attn-2 performs accurately and efficiently, cutting down on computation time and increasing accuracy in the classification of hyperspectral images when compared to other traditional ANN techniques.

DenseNet (Dense Convolutional Network) and connects all layers in a feed-forward manner (Huang et al. 2017). In conventional convolutional networks, each of the L layers has a connection with its subsequent layers. There are direct links in a network (L(L+1))/2. Each layer's feature map was used as an input for the layers that followed, while the current layer uses feature maps from earlier layers as an input. The vanishing-gradient problem is resolved by DenseNets. Four benchmark datasets—SVHN, CIFAR-10, Imagenet, and CIFAR-100 (Canadian Institute For Advanced Research)—are used for the experiments. The author's accuracy on the AID dataset was 97.44%, on the UC-Merced dataset it was 99.50%, on the Optimal it was 95.89%, and on the NWPU-RESIS45 dataset it was 94.98%. Convolutional Neural Networks (CNNs) have two problems (J. Zhang, Lu, et al. 2019).

Firstly, these models have multiple parameters, which leads to over-fitting, and secondly, they are not deep enough to extract abstract information. The author suggested the DenseNet pretrained model for remote sensing picture categorisation as a solution to these two issues. DenseNet generates a number of reusable features with fewer convolutional kernels. The network is deeper than 100 layers thanks to dense connections. They employ data augmentation. The AID (Aerial Image Dataset), UCM, NWPU-RESISC45, and Optimal-31 datasets are used for the experiments. The accuracy that the author has attained on the UCM dataset is 98.67% (50% training ratio), on the AID dataset it is 97.19% (50% training ratio), on the optimal-31 it is 95.41% (80% training ratio), on the NWPU-RESISC45 dataset it is 92.90 (10% training ratio), and on the NWPU-RESISC45 dataset it is 94.95 (20% training ratio).

In order to extract important characteristics from the image researchers suggested DenseNet-121 deep learning that was pre-trained using the transfer learning technique (Tan et al. 2023). DenseNet significantly reduces computational resources. Experiments are conducted using three benchmark datasets: urbansound8k, soundscapes1, and soundscapes2. With F1-scores of 80.7%, 87.3%, and 69.6%c respectively. Suggested pre-trained model DenseNet-121 with multilayer perceptron performs better than previous studies on the soundscapes1, soundscapes2, and urbansound8k datasets. DenseDsc and Dense2net are two CNN architectures that were proposed by the researchers (Li et al. 2021). These two CNNs are coupled in a compact manner which makes it easier for the networks to reuse features.Dense2net employed efficient group convolution, while DenseDsc adopted more effective separable convolution employing depth. Both of these methods increased the parameters' efficiency. The CIFAR and ImageNet datasets are used to assess the suggested methods. Using DenseDsc, the author obtained an accuracy of 74.2% on CIFAR-100 and 76.3% on ImageNet (top-1). Accuracy of 77% ImageNet utilising Dense2Net and 73.68% on CiFAR-100 were attained. Further advancements in image classification are yet possible.

3 Deep Learning Pre-Trained Model : DenseNet

In a traditional feed forward CNN, each convolution layer (aside from the input image) takes the output feature map produced by the previous convolution layer and sends it to the next convolution layer. There are thus L direct connections, one between each layer and the one that follows it, as shown in figure 1. As the number of layers of a CNN increases, the vanishing gradient problem arises. As the number of connections between the input and output levels increases, some information is lost or vanishes.

Consequently, the network's ability to learn efficiently is reduced. Therefore, training really deep networks was difficult. As they go deeper, CNNs transfer gradients and information in the opposite direction from the input layer, until the output layer is so big that it vanishes before it reaches the opposing side. DenseNets solves this issue since each layer has direct access to the loss function and gradients from the original input image. DenseNets solve this issue by simplifying layer-to-layer connectivity patterns and modifying the CNN architecture.

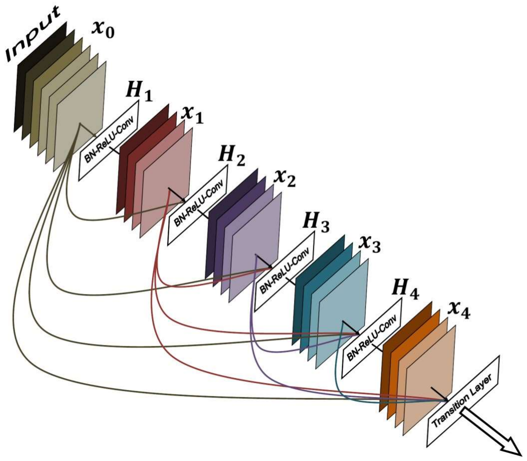


Figure 1: DenseNet Block Diagram

The network is referred to as a Densely Connected Network because each layer in the DenseNet architecture is closely related to every other layer. DenseNet is substantially more accurate and efficient to train when it contains L layers since it has L(L+1)/2 direct connections between layers between input and output. DenseNet is a feed forward network since all of its levels are directly connected to the front layers. The i^th layer's input can be the $(i-1)^th$, $(i-2)^th$, and $(i-n)^th$ layers' outputs. Batch normalisation is used by each directly connected layer in the network to normalise its input, reducing the absolute disparity between data.

All the levels are connected to ensure the optimum flow of information. The layers that come before them provide their feature maps and other inputs to all subsequent layers. At the moment widely used neural network topologies including

AlexNet (Krizhevsky, Sutskever, and Hinton 2012), ResNet (J. Zhang, Lu, et al. 2019), GoogleNet (C. Wang et al. 2019), VGG (variants include VGG-16 and VGG-19) (S. Zhou et al. 2018), and Inception (Pathak and Raju 2021) are used to classify remote sensing scenes.

Many methods based on these networks have limitations even though they significantly improve classification accuracy. These networks overfit because they are trained with limited data and have a large number of parameters (Li et al. 2021). Consequently, it is not sufficiently efficient to extract relevant semantic information from distant sensing images (J. Zhang, Lu, et al. 2019).

The deeper CNN DenseNet is proposed in this study as a solution to this issue. Because DenseNet has L(L+1)/2 direct and shorter connections between layers near input and output, it is more efficient at training and attaining better accuracy (Huang et al. 2017).

DenseNet is a feed forward network, which means that each layer is directly connected to the front layers, claims Zhang (2019). The ith layer's input can be the $(i_1)^th$, $(i-2)^th$, and $(i-n)^th$ layers' outputs. The network's direct connection of each layer reduces the absolute difference between data by normalising each layer's input using batch normalization (Imbriaco et al. 2021). All the levels are connected to ensure the optimum flow of information. Additional inputs from all previous layers are used to send its feature maps to all subsequent layers (Li et al. 2021).

DenseNet-201 is a densely connected convolutional neural network with 201 deep layers. The 224*224 input image is passed through several convolutional layers in DenseNet-201, and each convolutional layer uses channel-wise concatenation to send its feature maps to all succeeding layers. The depth of the ImageNet models is represented by variations with distinct layers. Every DenseNet variation uses an input image with three colour channels and dimensions of 224 by 224. DenseNet has multiple variations, including DensetNet-169, DenseNet-121, and DenseNet-201. The only difference between these variants is the number of layers, as seen in figure2 (Gao et al. 2021; J´egou et al. 2011).

3.1 Components of DenseNet

Bottleneck layers, growth rate, density blocks, and connection are DenseNet's four core components. Concatenated feature maps from all previous levels are used as inputs in each layer. The $i^{t}h$ layer receives the feature maps of all preceding layers, x0,...., xi-1. Redundant feature maps are removed and features are reused. Therefore, DenseNet requires less parameters than traditional CNNs.

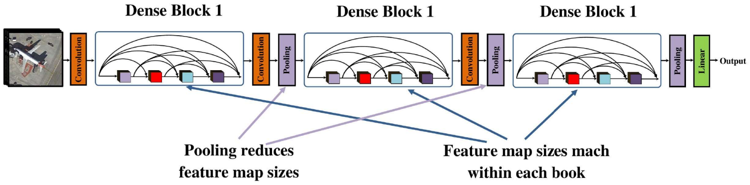


Figure 2: Dense Blocks

DenseNets concatenate the input feature maps with the output feature maps of the layer. Each layer has access to its predecessor's feature maps. Consequently, each layer adds new information to the corpus of knowledge.

$$Xl = Hl([x0,x0........Xl-1]) \quad (1)$$

$[x0,x1 < ldotp.......Xl-1]inthiscase$ stands for the feature-maps' concatenation, which is the result of all the layers before layers (0,...,l-1). For additional implementation, the several inputs of H_l are concatenated into a single tensor.

3.2 Dense Block Structure

All of the feature maps from the previous layers will be used as input for the i^{th} layer of DenseNets. Concatenation of feature maps from previous layers results in a feature map explosion. Dense blocks are made in order to solve the feature map explosion problem. A predetermined number of layers are contained within each DenseNet. When feature map sizes differ, it is impossible to group them together. Only feature maps of the same size that have been concatenated from the previous layers may be concatenated.

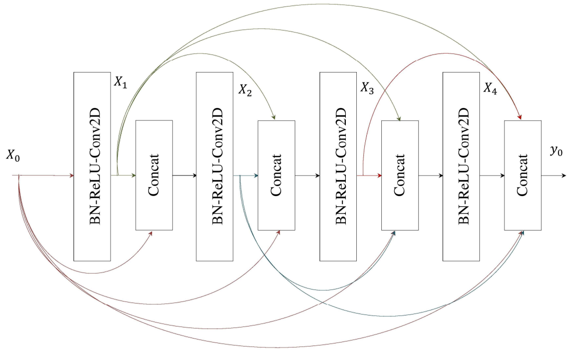


Figure 3: Connectivity of Layers

Dense blocks are used to separate DenseNets. While the number of filters may vary from block to block, the feature map size in these blocks stay constant. The blocks are separated by transitional layers. A dense block's output is sent to the transition layer. This transition layer is in charge of downsampling, which involves making feature maps smaller. With transition layers, there are half as many channels as there were before. This transition reduces the size of feature maps by using 1 × 1 convolution and Max pooling.

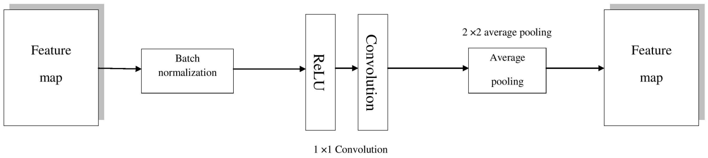


Figure 4: Transition Layer

H is described in the equation 1 as a composite function that applies three successive operations: a Convolution (Conv), a Rectified Linear Unit (ReLu), and Batch Normalisation.

Concatenation is made possible by feature maps having the same size, as seen in figure 4. Down sampling is the process of reducing the size of feature maps using a transition layer that consists of convolution and pooling. Growth rate is the rate at which feature maps expand in size after passing through thick layers, each of which adds ”k” features to preexisting features.

The network’s growth rate, which measures the amount of data added to each network layer, is represented by this ”k”. If each function H1 produces k feature maps, then the $l^t h$ layer has $kl = k0 + k \times (l - 1)$ input feature maps.$K0$ represents the number of channels in the input layer. Since each layer produces k output feature-maps, the number of inputs for later layers may be quite large. A 1 × 1 convolution layer is employed as a bottleneck layer prior to each 3 × 3 convolution. It improves the efficiency and speed of calculations.

To determine which of the four suggested datasets is the best variant to utilise as a feature extractor, experiments have been conducted on each DenseNet variant, including DenseNet-169, DensetNet-121, and DensetNet-201, for feature extraction of remote sensing images.

In conclusion, by adding new information to the network’s common knowledge, the input picture and the output of each DenseNet layer within each Dense Block are concatenated. The 32 new feature maps from the preceding layer are being added to by each layer. For this reason, after six levels, we move from 64 to 256. Additionally, Transition Block divides the volume and feature map count in half by performing 1x1 convolutions with 128 filters and then a 2x2 pooling with a stride of 2.

3.3 Unique Features of DenseNet Pre-Trained Model

The compact network structure of the DenseNet network is one of its distinctive properties when compared to other pre-trained networks. Higher computational and memory efficiency results from compact structures (Salim et al. 2023).

- The efficiency of the network is improved by feature reuse through concatenation of feature maps learnt by several layers, which enhances variety in the input of following layers (Albelwi 2022).
- Lower parameter countable the easy training of model. Reusable features can be learnt with very few kernels (Abai and Rajmalwar 2019).
- It also avoids over fitting and vanishing gradient Connections from early to late layers. Vanishing gradient problem occurs while training deep neural networks with gradient based learning method and back propagation. During each iteration of training each of the neural network weight receives an update proportional to the partial derivative of the error function with respect to current weight. While training in some of the cases the gradient becomes vanishing small thus effectively preventing the weight from changing its value. Vanishing gradient may completely stop the neural network from further training (Riasatian et al. 2021). DenseNet overcomes the situation of vanishing gradient well.
- Furthermore DenseNet have small number of convolution kernels that generates a large number of reusable feature maps by dense connections which makes the network deeper yet does not increase number of parameters (J. Zhang, Lu, et al. 2019).

4 Experimented Datasets

Remote sensing image retrieval system has been promoted with the advancement of new feature extraction methods experimented over new datasets. Benchmark datasets are the basis for implementing RSIR methods and performance evaluation. The literature has committed the recognizable progress for developing benchmark datasets for RSIR system. Datasets have two different categories according to retrieval modes. First is Unisource retrieval and second is cross source retrieval. In unisource retrieval both the query and retrieved images belong to same source. In cross source retrieval, both the query and retrieved images belong to two different sources. popularly used remote sensing datasets are explained in further sections.

4.1 UC-Merced Land Use Dataset

Dataset is also termed as UCM/UC-Merced. UCM contains 100 images having 21 classes. UCM is small in size having restricted classes. These classes belong to land use/land cover classification. All images have identical spatial resolution having measures of 256x256 pixels. There is collection of large aerial images obtained from US Geological Survey from USGS National Map Urban Area with spatial resolution of 0.3 meters. Classes in dataset include: airplanes, agricultural areas, forests, freeways, beaches, buildings, intersections, baseball diamonds, harbors, mobile home parks, chaparral, rivers, runways, sparse and dense residential areas, overpasses, parking lots, tennis courts, and storage tanks.Some overlapping classes are there in the dataset including medium, dense and sparse residential areas (Jiang et al. 2018) (Dong and Q. Zhang 2019) (C¸IKLAC¸ANDIR and Semih n.d.).

4.2 AID

A vast collection of aerial images gathered from Google Earth Imagery is called AID. The spatial resolutions of the scene images range from 0.5 to 8 meters.Airport, baseball pitch, undeveloped land, bridge, beach, centre, church, business, dense residential, forest, desert, meadow, farmland, industrial, mountain, medium residential, park, parking, pond, port, resort, playground, railway station, river, school, sparse residential, square, stadium, storage tanks and viaduct are among the classes in the dataset. It has thirty classes and 10,000 images. Each class contains between 220 and 420 (600X600) pixel images (Xia et al. 2017), (W. Zhang, Tang, and L. Zhao 2019).

4.3 PatterNet

The Google Map API of a few US cities gathered high-resolution images from Google Earth photography to create the very big dataset known as PatterNet. There are 800 256 x 256 pixel images in 38 classes. Aircraft, beach, baseball field, bridge, basketball court, charparral, cemetery, Christmas tree farm, crosswalk, closed road, coastal mansion, dense residential, ferry terminal, football field, forest, harbour, golf course, freeway, nursing home, intersection, mobile home park, parking lot, parking space, railway, river, runway, runway marking, shipping yard, solar, panel, sparse residential, storage tank, tennis court, swimming pool, transformer station and wastewater treatment plant are some of classes contained in PatterNet.

For the classification and retrieval of remote sensing images, PatterNet is an excellent collection with tagged data.Deep learning methods that require a lot of labelled data might benefit greatly from this labelled data (W. Zhou et al. 2018), (Yishu Liu, Yingbin Liu, et al. 2020).

4.4 NWPU-RESISC45

The NWPU-RESISC45 benchmark dataset is used to classify scenes using remote sensing. There are forty-five scenario classifications. Each class contains 700 images. A total of 31,500 256 by 256 pixel images are included.NWPU (Northwestern Polytechnic University) was the one who created it. The spatial resolution of the photographs varies from 0.2 to 30 meters. Airport, aeroplane, basketball court, baseball diamond, beach, bridge, church, chaparral, circular farmland, cloud, commercial area, dense residential, desert, forest, freeway, ground track field, golf course, harbour, intersection, industrial area, island, lake, medium residential, meadow, mobile home park, mountain, flyover, parking lot, palace, railway station, rectangular farmland, roundabout, river, runway, snowberg, sea ice, ship, sparse residential, storage tank, stadium, thermal power are the scene classes included in NWPU-RESISC45 (Yishu Liu, Yingbin Liu, et al. 2020), (Jiang et al. 2018).

Table 1: Remote Sensing Datasets

DATASET SELECTED	SCENE CLASSES	TOTAL IMAGES CONTAINED IN DATASET	IMAGE SIZE (Pixels)	SPATIAL RESOLUTION (m)
UC-Merced	21	2100	256×256	0.3
AID	30	10000	600×600	8-0.5
NWPU-RESISC45	45	31500	256×256	30-0.2
PatterNet	38	30400	256×256	4.69-0.06

5 Evaluation Metrics

Evaluating the algorithms is an essential part of any research. Well known evaluation metrics used by researchers are: Confusion Matrix, Accuracy, Precision, Recall and F1-Score.

5.1 Confusion Matrix

Confusion Metrics is used for binary classification problem like two classes: YES or NO as presented in table**??**

Table 2: Confusion Matrix

n=165	Predicted No	Predicted Yes
Actual: NO	50	10
Actual: YES	5	100

When the classifier predicted yes and the actual output was also yes, these are known as true positives.When the classifier predicted no and the actual output was no, these are known as true negatives. When a classifier predicted yes but the actual result was no, this is known as a false positive. When the classifier predicted no and the actual result was yes, this is known as a false negative.

5.2 Accuracy

The ratio of the number of predictions to the total number of predictions is known as accuracy. The percentage of correct predictions made by our model is known as accuracy. The number of real positive and negative results divided by the total number of samples is known as accuracy.

$$Accuracy = \frac{TruePositives + TrueNegatives}{TotalSamples} \quad (2)$$

5.3 Precision

Precision indicates the ratio of actual positive results by the number of positive results as predicted by the classifier.

$$Precision = \frac{TruePositives}{TotalPositives + FalsePositives} \quad (3)$$

5.4 Recall

Recall indicates the ratio of correct positive results by the number of all actual samples.

$$Recall = \frac{TruePositives}{TruePositives + FalseNegatives} \quad (4)$$

5.5 F1-Score

F1 Score is the Harmonic Mean between precision and recall.

$$F1Score = \frac{2P \times R}{P + R} \quad (5)$$

The F1 Score range is [0, 1].It shows how accurately the classifier is inferring the number of cases that it correctly classifies and does not miss a sizable number of instances.The F1 Score is the equilibrium of recall and precision. High accuracy combined with low recall produces incredibly precise answers, but it has the drawback of missing many hard-to-classify occurrences. The better the selected model performs, the higher the accuracy, precision, recall, and F1 Score values.

5.6 Training Accuracy

The accuracy of a model based on the dataset samples it was trained on. Generally the accuracy of model increases with increasing number of epochs.

5.7 Validation Accuracy

Also termed as testing accuracy, computed on the dataset on which the model has not been trained. Model has not seen these dataset examples during training. It indicates the generalization of the model. Validation accuracy is expected to be either less than or close to the training accuracy. The significance difference between validation accuracy and training accuracy indicates under fitting and overfiting respectively.

5.8 Training Loss

It is used to update the model's parameters to minimize loss on training data. It represents how well the model is fitting the training data. It decreases over time as model learns from data.

5.9 Validation Loss

During training the portion of data not used for training is kept aside as a validation set. It measures how well the model is performing on validation set. It monitors model generalizes to unseen during training. If validation loss increases while training loss decreases it represents overfitting.

6 Remote Sensing Image Classification based on Pre-Trained Models

The investigations are based on selected pre-trained models like VGG-19, Inception-v3 and DenseNet etc. Out of all these models, one best model with highest testing accuracy and least test loss achieved for image classification may be adopted for Remote Sensing classification and retrieval system.

7 Remote Sensing Image Classification based on Pre-Trained Models

In order to determine the best dataset for remote sensing picture classification, the studies are based on a pre-trained model called DenseNet, along with its versions, DenseNet-169, DenseNet121, and DenseNet-201 using four remote sensing datasets: UC-Merced, AID, NWPU-RESISC-45 and PatterNet.

7.1 General Flow of Transfer Learning Algorithm for Remote Sensing Image Classification

- Load the remote sensing datasets selected for experiments

Division of the dataset into a train, test and validation splits like 70% and 30% ratios is

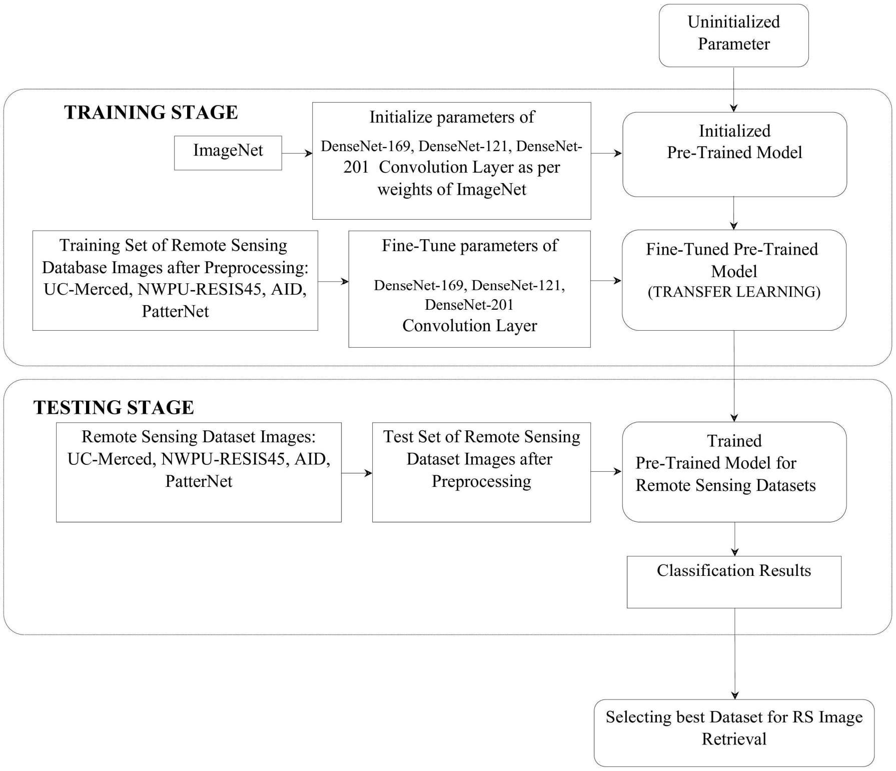

Uninitialized Parameter
TRAINING STAGE
ImageNet
Initialize parameters of DenseNet-169, DenseNet-121, DenseNet-201 Convolution Layer as per weights of ImageNet
Initialized Pre-Trained Model
Training Set of Remote Sensing Database Images after Preprocessing: UC-Merced, NWPU-RESIS45, AID, PatterNet
Fine-Tune parameters of DenseNet-169, DenseNet-121, DenseNet-201 Convolution Layer
Fine-Tuned Pre-Trained Model (TRANSFER LEARNING)
TESTING STAGE
Remote Sensing Dataset Images: UC-Merced, NWPU-RESIS45, AID, PatterNet
Test Set of Remote Sensing Dataset Images after Preprocessing
Trained Pre-Trained Model for Remote Sensing Datasets
Classification Results
Selecting best Dataset for RS Image Retrieval

Figure 5: Remote Sensing Image Classification based on Pre-Trained Models performed. Train dataset consists of images to train the model. Validation dataset contains the images to validate the model in every epoch. They are used to obtain the training and validation accuracies and loss in every epoch while training the model. Test dataset contains the completely unseen images of Remote Sensing dataset. • Set the size of input image given as input to pre-trained model(224,224).

Generally pre-trained models accepts the input image of 224×224 of Remote Sensing dataset but actually the dataset images are of different dimensions. Thus, the dataset images must be resized according to required dimensions.

- Model Training
  1. Model building
     (a) Load pre-trained base model with its pre-trained weights.
     (b) Customize the mode: Changing last layer according to number of classes of containedin dataset.

(c) Predicting Classes: Multiclass classification is done using Softmax activation functionat dense output layer.

2. Compiling Model

   Categorical cross entropy function is evaluated using Adam optimizer. Adam optimizer decides the best training rate itself for compiling model.

3. Fitting the Model

   Now model is fit to train. Early stopping is used to stop training the model if further validation loss gradually increases.

4. Performance Evaluation

   Accuracy and loss learning curves are plotted. Accuracy, precision, recall, F1-scores are analyzed. General flow of transfer learning is presented in figure 8.

8 Train, Validation, Test Dataset Details on variants of DenseNet

Every DenseNet variation is learnt using 100 epochs. The following tables compare four benchmark datasets' total parameters, non-trainable parameters, validation accuracy, validation loss, precision, recall, and f1-scores. Test Accuracy measures the predictive performance of the model by evaluating how well its predictions match the actual data. The test loss metric which calculates the cumulative mistakes on the validation set after each iterative optimisation shows how effective the model is.

8.1 DenseNet-169: AID

The results of the trials on the AID dataset are shown in Figure5.These investigations yielded a test accuracy of 87%, indicating that the model's predictions were extremely accurate. Furthermore, figure5 displays the associated test loss, which was measured at 2.0396 and suggests a relatively larger degree of error in the model's predictions.

8.2 DenseNet-169: UC-Merced

Figure6 displays the outcomes of the tests conducted using the UC-Merced dataset. The model's predictions were quite accurate, as evidenced by the 90% test accuracy obtained from these studies. Additionally, the comparable test loss of 1.2614 is shown in figure 6 suggesting a comparatively smaller degree of error in the model's predictions.

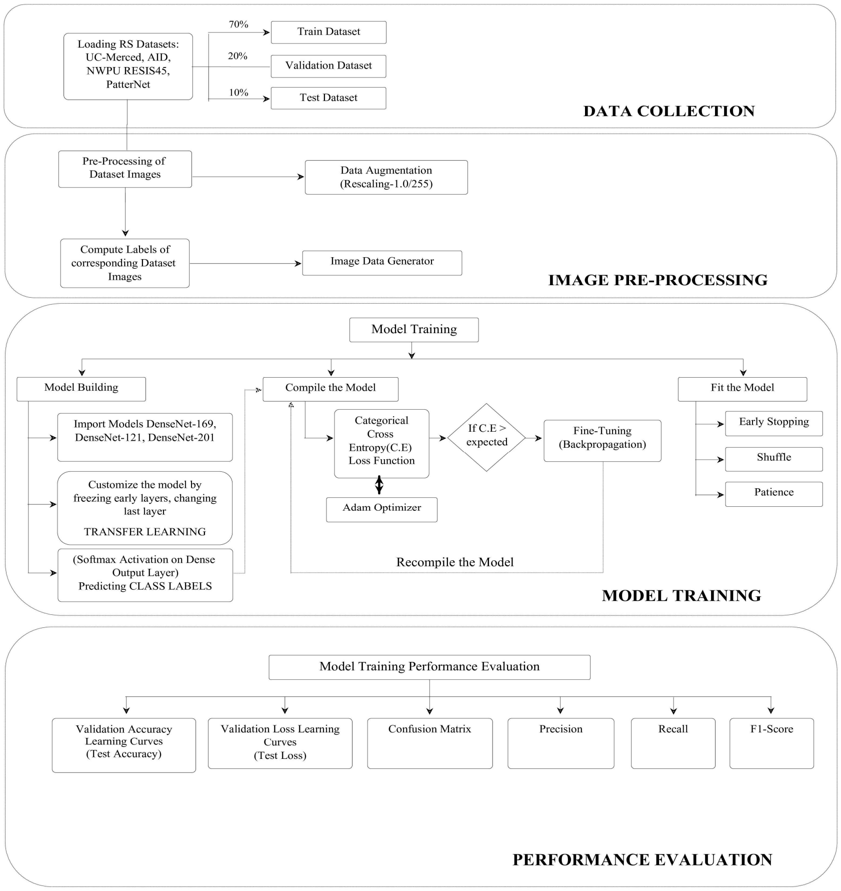
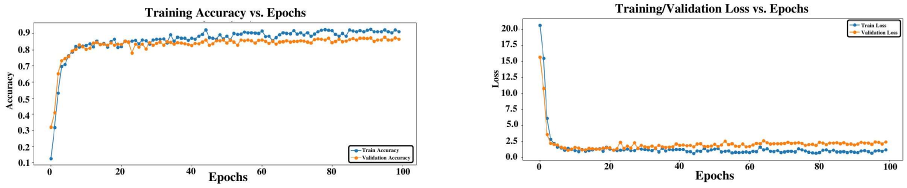
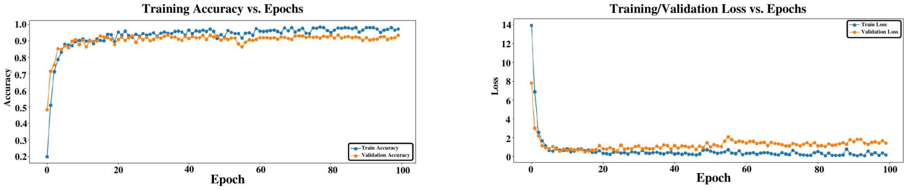

Loading RS Datasets: UC-Merced, AID, NWPU RESIS45, PatterNet
70%
20%
10%
Train Dataset
Validation Dataset
Test Dataset
DATA COLLECTION
Pre-Processing of Dataset Images
Data Augmentation (Rescaling-1.0/255)
Compute Labels of corresponding Dataset Images
Image Data Generator
IMAGE PRE-PROCESSING
Model Training
Model Building
Compile the Model
Fit the Model
Import Models DenseNet-169, DenseNet-121, DenseNet-201
Customize the model by freezing early layers, changing last layer
TRANSFER LEARNING
(Softmax Activation on Dense Output Layer) Predicting CLASS LABELS
Categorical Cross Entropy(C.E) Loss Function
Adam Optimizer
If C.E > expected
Fine-Tuning (Backpropagation)
Recompile the Model
Early Stopping
Shuffle
Patience
MODEL TRAINING
Model Training Performance Evaluation
Validation Accuracy Learning Curves (Test Accuracy)
Validation Loss Learning Curves (Test Loss)
Confusion Matrix
Precision
Recall
F1-Score
PERFORMANCE EVALUATION

Figure 6: Remote Sensing Image Classification based on Transfer Learning



(a) Accuracy: 0.8769 (b) Loss: 2.0396

Figure 7: DenseNet-169: AID Dataset

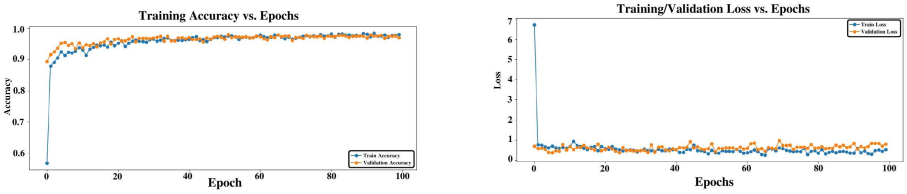


(a) Accuracy: 0.9048 (b) Loss: 1.2614

Figure 8: DenseNet-169: UC-Merced Dataset

8.3 DenseNet-169: PatterNet

Figure 7 displays the outcomes of the experiments conducted on the PatterNet dataset.As a consequence of these studies, the test accuracy was 96.3%, which indicates that the model's predictions are highly accurate. The related test loss measure was 1.0238, which indicates that the model's predictions are somewhat moderately inaccurate.



(a) Accuracy: 0.9632 (b) Loss: 1.0238

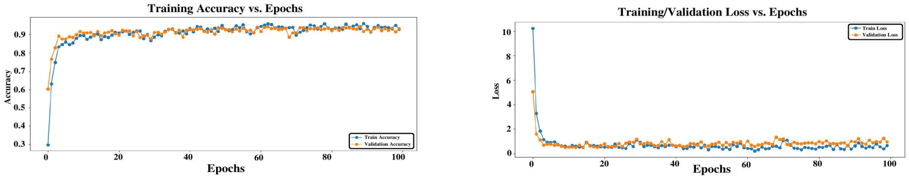
Figure 9: DenseNet-169: PatterNet Dataset

8.4 DenseNet-169: NWPU-RESISC45

Figure 8 displays the outcomes of the tests conducted using the NWPU-RESISC45 dataset.As a result of these trials, the test accuracy was 93.2%, which indicates that the model's predictions were highly accurate. The corresponding test loss was 0.8443, which indicates that the model's predictions were significantly less inaccurate.

Systematic analysis of Test Accuracy and Test Loss Scores on proposed remote sensing datasets AID, UC-Merced, NWPU-RESISC45 and PatterNet with the variants of the pre-trained models is presented in this section.

Table 3 presents the precision, recall, F1-score and accuracy scores as observed on DenseNet-169.



(a) Accuracy: 0.9325 (b) Loss: 0.8443

Figure 10: DenseNet-169: NWPU-RESISC45 Dataset

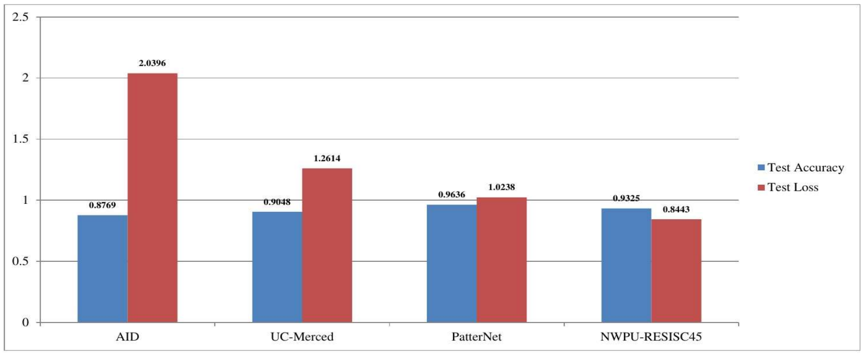
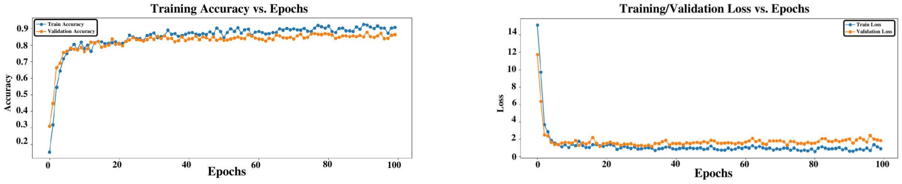


Figure 11: DenseNet-169: Test Accuracy Test Loss Analysis AID, UC-Merced, ,PatterNet, NWPURESISC45

8.5 Observing Learning Curves of Train Accuracy and Train Loss Scores on DenseNet-121 8.6DenseNet-121: AID

The results of the studies are shown in DenseNet-121 Figure 10, which yields an accuracy of 88%. This outcome shows that the model performed rather well in predicting the test data's class labels. Similarly, a wider range for loss optimisation was indicated by the equivalent test loss, which was found to be 1.6608.



(a) Accuracy: 0.8838 (b) Loss: 1.6608

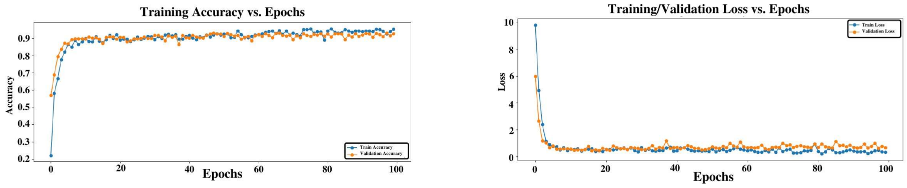
Figure 12: DenseNet-121: AID Dataset

Table 3: DenseNet-169: Investigations on Precision, Recall, F1-Scores on AID, UC-Merced, PatterNet and NWPU-RESISC45 Datasets

Dataset	Precision	Recall	f1-Score	Accuracy	Support
UC-Merced	0.91	0.90	0.90	0.90	210
AID	0.89	0.88	0.88	0.88	1007
NWPU-RESISC45	0.93	0.93	0.93	0.93	1067
PatterNet	0.97	0.96	0.96	0.96	3040

8.7 DenseNet-121:UC-Merced

DenseNet-121 The results of studies on the UC-Merced dataset are shown in Figure 11, which yielded a test accuracy of 93%. This result shows that the model performed reasonably well in predicting the test data's class labels. Similarly, a lower range for loss optimisation was indicated by the matching test loss, which was found to be 0.4508.

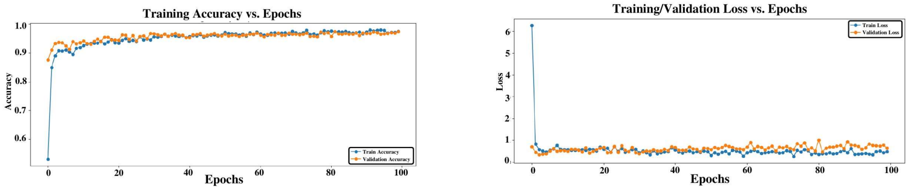


(a) Accuracy: 0.4508 (b) Loss: 0.9381

Figure 13: DenseNet-121: UC-Merced Dataset

8.8 DenseNet121: PatterNet

The results show that the DenseNet-121 model performs reasonably well in categorising the test images, with a test accuracy of 97% in figure 12. Furthermore, a matching test loss of 0.6076 was found, indicating a comparatively reduced degree of error in the model's predictions for this specific dataset.

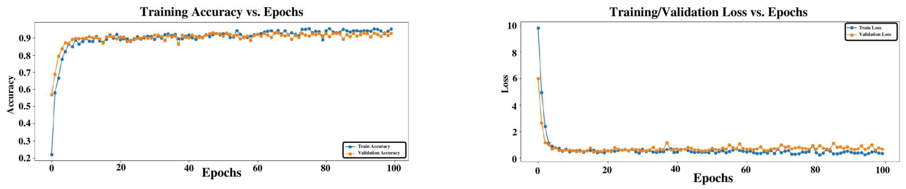


(a) Accuracy: 0.9727 (b) Loss: 0.67076

Figure 14: DenseNet-121: PatterNet Dataset

8.9 DenseNet-121: NWPU-RESISC45

DenseNet-121 Figure 13 displays the outcomes of the tests conducted using the NWPU-RESISC45 dataset.With a test accuracy of 92.2%, these experiments demonstrated that the model's predictions were highly accurate. A comparatively small degree of error in the model's predictions was shown by the corresponding test loss, which was assessed at 0.7884.



(a) Accuracy: 0.9228 (b) Loss: 0.7884

Figure 15: DenseNet-121: NWPU-RESISC45 Dataset

9 Analysis of Train Accuracy and Train Loss Scores on Proposed Datasets based on Investigations using DenseNet201

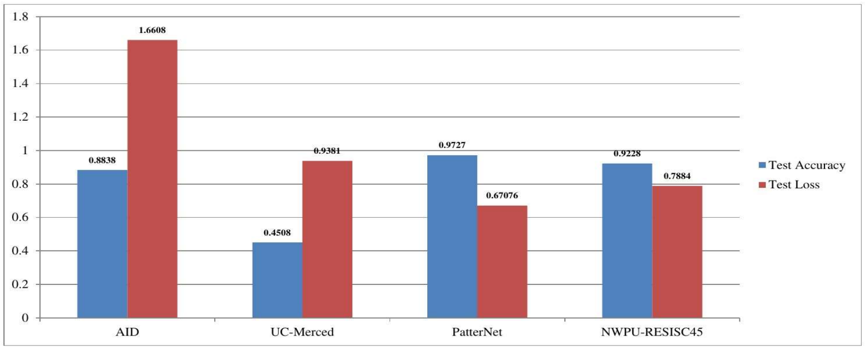
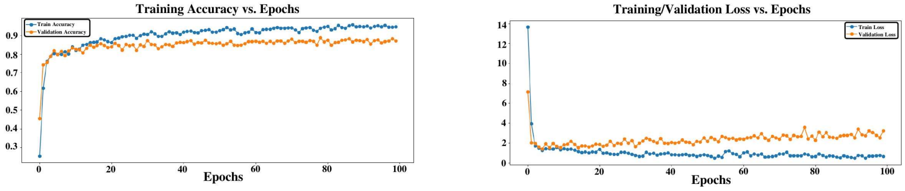


Figure 16: Test Accuracy Test Loss Analysis DenseNet-121: NWPU-RESISC45,PatterNet, UCMerced and AID datasets

DenseNet-121 is trained on 100 epochs. Comparison of total parameters, non-trainable parameters, validation accuracy, validation loss, precision, recall and F1-scores on four benchmark datasets is represented in following tables.

9.1 DenseNet-201: AID

The results obtained with the experiments carried out on the AID dataset are presented in figure 15.These experiments resulted in test accuracy of 86.6%, indicating a high level of accuracy in the model's predictions. Corresponding test loss was measured at 2.709 indicating a relatively higher level of error in the model's predictions.



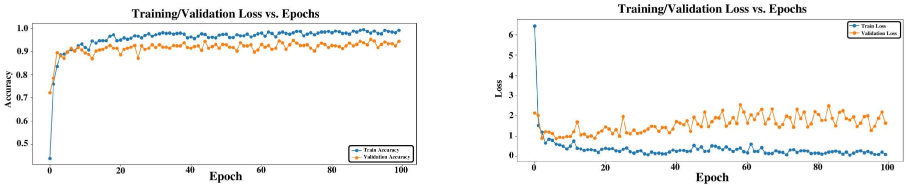
(a) Accuracy: 0.8669

(b) Loss: 2.7096

Figure 17: DenseNet-201: AID Dataset

9.2 DenseNet-201: UC-Merced

DenseNet-201 Figure 16 displays the outcomes of the tests conducted on the UC-Merced dataset on DenseNet-201.A test accuracy of 94% was obtained from these studies, suggesting that the model's predictions were highly accurate. A comparatively larger degree of error in the model's predictions is indicated by the equivalent test loss of 1.323.



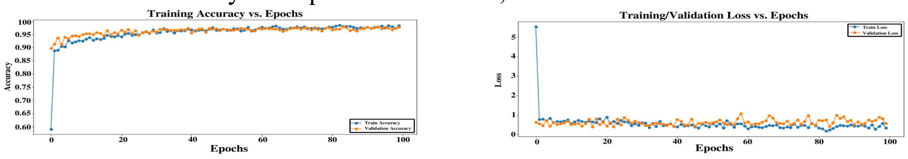
(a) Accuracy: 0.9429

(b) Loss: 1.3238

Figure 18: DenseNet-201: UC-Merced Dataset

9.3 DenseNet 201: PatterNet

The results of the studies are shown in Figure 17, which shows a test accuracy of 97.6%. This outcome shows that the model performed rather well in predicting the test data's class labels. Similarly, a lower range for loss optimisation was indicated by the equivalent test loss, which was found to be 0.640.



(a) Accuracy: 0.9760 (b) Loss: 0.6402

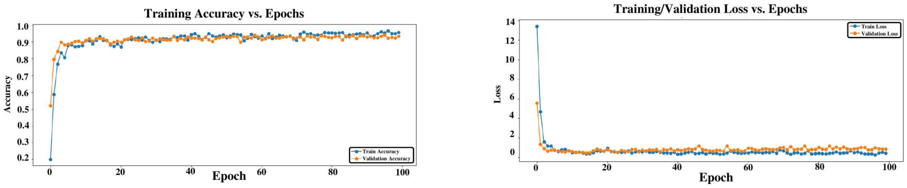
Figure 19: DenseNet-201: PatterNet Dataset

9.4 DenseNet 201: NWPU-RESISC45

DenseNet-201 Figure 18 displays the outcomes of the tests conducted using the NWPU-RESISC45 dataset.The model's predictions were quite accurate, as evidenced by the 90% test accuracy obtained from these studies. Additionally, the comparable test loss, which was 0.8955, is shown in figure 13, suggesting a comparatively higher degree of error in the model's predictions.

Table 4: DenseNet-201: Investigations on Precision, Recall, F1-Scores on AID, UC-Merced, NWPURESISC45 and PatterNet Datasets

Dataset	Precision	Recall	f1-Score	Accuracy	Support
UC-Merced	0.95	0.94	0.94	0.94	210
AID	0.90	0.89	0.89	0.89	1007
NWPU-RESISC45	0.93	0.93	0.93	0.93	1062
PatterNet	0.98	0.98	0.98	0.98	3040



(a) Accuracy: 0.9275 (b) Loss: 0.8956

Figure 20: DenseNet-201: NWPU-RESISC45 Dataset

10 Analysis of Train Accuracy and Train Loss Scores on DenseNet-201

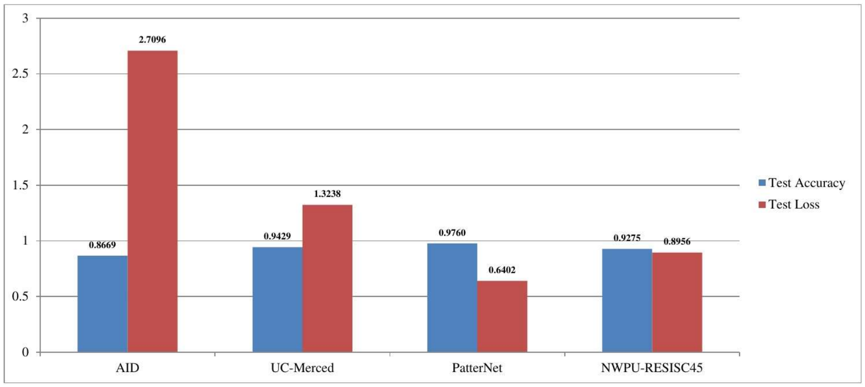


Figure 21: Test Accuracy Test Loss Analysis DenseNet-201: NWPU-RESISC45,PatterNet, UCMerced and AID datasets

11 Analysing DenseNet-169, DenseNet-121 and DenseNet201 evaluations with existing approaches and syntactic analysis

Table 5: Performance Analysis of DenseNet used as feature extractor and State-of-the-art Machine Learning and Deep Learning feature extraction methods adopted for Image Classification in recent past decade.

Year	Author	Mehodology	Observations		
			DATASET	Evaluation Metrics	Score
2014	G.Sumbul	DAS-RHDIS	UC-Merced	Accuracy Precision Recall F1-Score	56.8% 65.3% 70% 67.5%
			BigEarthNet	Accuracy Precision Recall F1-Score	62.7% 77.7% 75.7% 76.7%
2015	C.Szegedy	Inception-v3	ImageNet	Top-1 Error Top-5 Error	17.2% 3.58%
2016	B.Chaudhari	ARGM	2100 aerial orthoimagery	Precision Recall ANMR	59.76% 70% (Top-20) 60% 0.5748
2017	Do & Cheung	SASH Supervised	CIFAR 10 MNIST NUS-WIDE	MAP MAP MAP	53.17% 75.48% 64.01%
		SASH Unsupervised	CIFAR 10 MNIST NUS-WIDE	MAP MAP MAP	77.22% 63.31% 45.05%
2017	J.Li	DMINTIR	Oxford 5k	Mean Average Precision	85.34%
			Paris 6k	Mean Average Precision	81.75%
2018	Jabeen	SURF based FREAK	Corel 1K Corel 1.5K Caltech 256	Accuracy Accuracy Accuracy	86% 83% 38%
			GID	Accuracy	77.74%
2020	M.K. Alsmadi	YCbCr + GLCM	Corel	Average Precision Average Recall	90.15 18.03
2021	G. Li, M.Zhang	DenseDsc	CIFAR-10 CIFAR-100 ImageNet	Accuracy Accuracy Accuracy	94.05% 74.24% 76.3%
		Dense2Net	CIFAR-10	Accuracy	94.19%
			CIFAR-100	Accuracy	73.68%
			ImageNet	Accuracy (Top-5) Accuracy (Top-1)	76.3% 77.1%
2022	Z.Zhang	MKANet kam Class	RSSCN7	Precision mAP	0.8963 (89.6%) 0.9167 (91.6%)
		MobileNetV3	RSSCN7	Precision mAP	0.8573 (85.7%)

2019
Dong
ResNet-101
ISPRS
Accuracy
86.67%

TECHNIQUE	OBSERVATIONS		
	DATASET	EVALUATION METRICS	SCORE
Proposed	UC-Merced	Precision	0.91
		Recall	0.90
		F1-score	0.90
		Accuracy	0.90
		Test Accuracy	0.9048
		Test Loss	1.2614
	AID	Precision	0.89
		Recall	0.88
		F1-score	0.88
		Accuracy	0.88
		Test Accuracy	0.8769
		Test Loss	2.0396

					0.8586 (85.8%)
		MKANet Class	OPTIMAL-31	Precision	0.7447 (74.4%)
				mAP	0.7399 (73.9%)
				ANMR	0.0325 (32.5%)
2023	P.S.Tan	DenseNet-121	Soundscapes1	F1-score	80.70%
			Soundscapes2	F1-score	87.30%
			Urban-Sound8k	F1-score	69.60%

Table 6: Evaluations done on DenseNet-169, DenseNet-121 and Densenet-201

Method (DenseNet-169)			
	NWPU-RESISC45	Precision	0.93
		Recall	0.93
		F1-score	0.93
		Accuracy	0.93
		Test Accuracy	0.9325
		Test Loss	0.8443
	PatterNet	Precision	0.97
		Recall	0.96
		F1-score	0.96
		Accuracy	0.96
		Test Accuracy	0.9632
		Test Loss	1.0238

Proposed Method (DenseNet-121)	UC-Merced	Precision	0.95
		Recall	0.94
		F1-score	0.94
		Accuracy	0.94
		Test Accuracy	0.9381
		Test Loss	0.4508
	AID	Precision	0.90
		Recall	0.88
		F1-score	0.88
		Accuracy	0.88
		Test Accuracy	0.8838
		Test Loss	1.6608
	NWPU-RESISC45	Precision	0.93
		Recall	0.92
		F1-score	0.92
		Accuracy	0.92
		Test Accuracy	0.9228
		Test Loss	0.7884
	PatterNet	Precision	0.97
		Recall	0.97
		F1-score	0.97
		Accuracy	0.97
		Test Accuracy	0.9727
		Test Loss	0.6076
Proposed Method (DenseNet-201)	UC-Merced	Precision	0.95
		Recall	0.94
		F1-score	0.94
		Accuracy	0.94
		Test Accuracy	0.9429
		Test Loss	1.3238
	AID	Precision	0.90
		Recall	0.89
		F1-score	0.89
		Accuracy	0.89
		Test Accuracy	0.89
		Test Loss	2.7096
	NWPU-RESISC45	Precision	0.93
		Recall	0.93
		F1-score	0.93
		Accuracy	0.93
		Test Accuracy	0.9275
		Test Loss	0.8956

PatterNet	Precision	0.98
	Recall	0.98
	F1-score	0.98
	Accuracy	0.98
	Test Accuracy	0.9760
	Test Loss	0.6402

Table 5 and 6 presents the performance analysis of various state of the art feature extraction methods adopted by researchers in past decade compared with the proposed pre-trained model DenseNet on proposed remote sensing datasets.

12 Complexity and Challenges

DenseNet-169 has a depth of 338 and a size of 57 MB. DenseNet-121 has a depth of 242 and is 33 MB in size. DenseNet-201 has a depth of 402 and a size of 80 Mb. Variants of DenseNet have demonstrated greater computational efficiency than VGG-19 and Inception Networks (GoogLeNet/Inception v1), both in terms of the quantity of parameters produced by the network and the financial cost paid (memory and other resources). Care must be made when training DeneNet networks with large and complex sizes to ensure that the computational benefits are maintained.

13 Contribution of the research work proposed

The content-based remote sensing information retrieval system could be helpful in forestry and agriculture, among other industries. Agricultural fields may benefit from CBIR's ability to identify damaged crops just by looking at them from above. The impacted area can be remotely sensed to track deforestation. The RSIR system is also used in geosciences, where satellites may collect images to determine the earth's geological characteristics at a specific time and location. Through remote sensing of planet positions, astrologers can get information regarding planet motions. The minerals that are present on planets can be recorded by scientists. To predict the weather, the weather forecasting department may employ remote sensing.

14 Conclusion

The goal of DenseNet is to maximise the flow of information between the network's intermediary levels. It immediately connects every layer. The key component of DenseNet is Dense Block. The primary characteristic is that every network layer is interconnected. Each layer's input is derived from the output of every layer before it. Each layer in a dense block architecture is directly connected to input and loss, which encourages information transmission and lowers the gradient, allowing the network to converge more effectively.

In order to classify remote sensing images, this paper experiments with transfer learning using variations of the DenseNet pre-trained deep learning model. The four benchmark datasets previously indicated are used for the evaluations. In light of the aforementioned findings, DenseNet-201 speeds up training and gets rid of over-fitting. With fewer parameters, DenseNet-201 increases the remote sensing scene's classification accuracy. Additionally, overfitting can be avoided and training accelerated by using pre-training parameters that were trained on ImageNet.

Comparing with other datasets, the PatterNet dataset is found to be the most appropriate for classifying remote sensing images. This conclusion is based on the remarkable low validation loss of 0.640 and the excellent accuracy of up to 97%. The distinctive features of the PatternNet dataset, such as its extensive collection of high-resolution images on a huge scale, are largely responsible for its appropriateness.

References

1. Abai, Zoheb and Nishad Rajmalwar (2019). "Densenet models for tiny imagenet classification". In: *arXiv preprint arXiv:1904.10429*.
2. Albelwi, Saleh A (2022). "Deep architecture based on DenseNet-121 model for weather image recognition". In: *International Journal of Advanced Computer Science and Applications* 13.10.
3. Alzu'bi, Ahmad, Abbes Amira, and Naeem Ramzan (2017). "Content-based image retrieval with compact deep convolutional features". In: *Neurocomputing* 249, pp. 95–105.

4. C¸IKLAC¸ANDIR, Fatma YAS¸AR and UTKU Semih (n.d.). “Performance Comparison of CNN Based Hybrid Systems Using UC Merced Land-Use Dataset”. In: *Dokuz Eylu¨l Universitesi Mu¨hendislik¨ Faku¨ltesi Fen ve Mu¨hendislik Dergisi* 25.75 (), pp. 725–737.
5. Dong, Yunya and Qian Zhang (2019). “A combined deep learning model for the scene classification of high-resolution remote sensing image”. In: *IEEE Geoscience and Remote Sensing Letters* 16.10, pp. 1540–1544.
6. Fan, Lili, Hongwei Zhao, and Haoyu Zhao (2020). “Distribution consistency loss for large-scale remote sensing image retrieval”. In: *Remote Sensing* 12.1, p. 175.
7. Gao, Jian et al. (2021). “Multiview graph convolutional hashing for multisource remote sensing image retrieval”. In: *IEEE Geoscience and Remote Sensing Letters* 19, pp. 1–5.
8. Huang, Gao et al. (2017). “Densely connected convolutional networks”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4700–4708.
9. Imbriaco, Raffaele et al. (2021). “Toward multilabel image retrieval for remote sensing”. In: *IEEE Transactions on Geoscience and Remote Sensing* 60, pp. 1–14.
10. J´egou, Herv´e et al. (2011). “Aggregating local image descriptors into compact codes”. In: *IEEE transactions on pattern analysis and machine intelligence* 34.9, pp. 1704–1716.
11. Jiang, Shulong et al. (2018). “A novel framework for remote sensing image scene classification”. In: *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* 42, pp. 657–663.
12. Karimpouli, Sadegh and Pejman Tahmasebi (2019). “Image-based velocity estimation of rock using convolutional neural networks”. In: *Neural Networks* 111, pp. 89–97.
13. Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E Hinton (2012). “Imagenet classification with deep convolutional neural networks”. In: *Advances in neural information processing systems* 25.
14. Li, Guoqing et al. (2021). “Efficient densely connected convolutional neural networks”. In: *Pattern Recognition* 109, p. 107610.
15. Liu, Yishu, Liwang Ding, et al. (2020). “Similarity-based unsupervised deep transfer learning for remote sensing image retrieval”. In: *IEEE Transactions on Geoscience and Remote Sensing* 58.11, pp. 7872–7889.
16. Liu, Yishu, Yingbin Liu, et al. (2020). “Remote-sensing image retrieval with tree-triplet-classification networks”. In: *Neurocomputing* 405, pp. 48–61.
17. Paoletti, Mercedes E et al. (2018). “A new deep convolutional neural network for fast hyperspectral image classification”. In: *ISPRS journal of photogrammetry and remote sensing* 145, pp. 120–147.
18. Pathak, Debanjan and USN Raju (2021). “Content-based image retrieval using feature-fusion of GroupNormalized-Inception-Darknet-53 features and handcraft features”. In: *Optik* 246, p. 167754.
19. Riasatian, Abtin et al. (2021). “Fine-tuning and training of densenet for histopathology image representation using tcga diagnostic slides”. In: *Medical image analysis* 70, p. 102032.
20. Risojevi´c, Vladimir and Vladan Stojni´c (2021). “Do we still need ImageNet pre-training in remote sensing scene classification?” In: *arXiv preprint arXiv:2111.03690.*
21. Salim, Farsana et al. (2023). “DenseNet-201 and Xception pre-trained deep learning models for fruit recognition”. In: *Electronics* 12.14, p. 3132.
22. Tan, Pooi Shiang et al. (2023). “Pre-trained DenseNet-121 with Multilayer Perceptron for Acoustic Event Classification.” In: *IAENG International Journal of Computer Science* 50.1.
23. Wang, Cheng et al. (2019). “Pulmonary image classification based on inception-v3 transfer learning model”. In: *IEEE Access* 7, pp. 146533–146541.
24. Wang, Jie et al. (2020). “Object-scale adaptive convolutional neural networks for high-spatial resolution remote sensing image classification”. In: *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 14, pp. 283–299.

25. Wang, Yameng, Shunping Ji, and Yongjun Zhang (2021). "A learnable joint spatial and spectral transformation for high resolution remote sensing image retrieval". In: *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 14, pp. 8100–8112.
26. Xia, Gui-Song et al. (2017). "AID: A benchmark data set for performance evaluation of aerial scene classification". In: *IEEE Transactions on Geoscience and Remote Sensing* 55.7, pp. 3965–3981.
27. Zhang, Jianming, Xiaokang Jin, et al. (2020). "Spatial and semantic convolutional features for robust visual object tracking". In: *Multimedia Tools and Applications* 79, pp. 15095–15115.
28. Zhang, Jianming, Chaoquan Lu, et al. (2019). "A full convolutional network based on DenseNet for remote sensing scene classification". In: *Mathematical Biosciences and Engineering* 16.5, pp. 3345–3367.
29. Zhang, Wei, Ping Tang, and Lijun Zhao (2019). "Remote sensing image scene classification using CNN-CapsNet". In: *Remote Sensing* 11.5, p. 494.
30. Zhou, Shuren et al. (2018). "Improved VGG model for road traffic sign recognition". In: *Computers, Materials & Continua* 57.1, pp. 11–24.
31. Zhou, Weixun et al. (2018). "PatternNet: A benchmark dataset for performance evaluation of remote sensing image retrieval". In: *ISPRS journal of photogrammetry and remote sensing* 145, pp. 197–209.