MJoSHT
Malaysian Journal of Science, Health & Technology

[mjosht.usim.edu.my]

USIM

Article

Leveraging Inception V3 and VGG16 for Accurate Identification of Residential and Commercial Zones

Valliappan Raman[1], Putra Sumari[2], Prabhavathy M[1], Sundresan Perumal[3]

[1]*Department of Artificial Intelligence and Data Science, Coimbatore Institute of Technology, Coimbatore, Tamilnadu 641014, India.*
[2]*School of Computer Science, Universiti Sains Malaysia, 11800 Penang, Malaysia.*
[3]*Faculty of Science and Technology, University Sains Islam Malaysia, 71800 Nilai, Negeri Sembilan, Malaysia.*

Correspondence should be addressed to:
Sundresan Perumal; sundresan.p@usim.edu.my

Article Info
Article history:
Received: 18 April December 2024
Accepted: 30 September 2024
Published:7 October 2024
Academic Editor:
Shahrina Ismail

Malaysian Journal of Science, Health & Technology

MJoSHT2024, Volume 10, Issue No. 2
eISSN: 2601-0003

https://doi.org/10.33102/mjosht.v10i2.396

***Abstract* — Image classification of land using aerial scene classification has become increasingly common across the world. With the power of Convolutional Neural Networks (CNNs), the identification of various city township areas using satellite imagery has become more efficient as compared to the previous manual labeling method. The objective of this research is to build a convolutional neural network model for residential and commercial area identification. The primary aim of this identification system is to support urban planning, land use management, and infrastructure development by providing accurate, automated insights into the spatial distribution of residential and commercial zones. By replacing manual mapping processes, this system can greatly improve the efficiency in analyzing urban landscapes, monitor urban growth, and assist in making informed decisions related to sustainable city planning and zoning regulations. In this research, Inception V3 and VGG16 were adopted to develop two transfer learning models for the identification system. The Inception V3-based model achieved the highest overall accuracy value of 100%, showing its effectiveness in the accuracy of residential and commercial area identification. The proposed CNN model achieved an accuracy of 99%, while the VGG-16 model, with all configurations being frozen, achieved 99% accuracy.**

Keywords — CNNs; Transfer Learning; Satellite Image

I. INTRODUCTION

The rapid growth of cities in Malaysia has resulted in an increased demand for efficient and accurate identification of residential and commercial areas. This demand arises from various applications, such as urban planning, real estate development, and resource allocation [1]. In addressing this need, studies have focused on the development of identification systems for residential and commercial areas utilizing satellite imagery. Understanding land usage through the use of satellite imaging or aerial photography has been a popular subject of extensive research. Both offer a bird's-eye view of the planet and are employed in geography research and land surveying.

Aerial scene classification provides a high-level interpretation of aerial images by assigning a semantic label to the entire scene or a significant portion of the scene [2]. This approach aims to categorize the image based on its content, such as identifying whether it depicts a residential area, a forest, or a commercial district. On the other hand, pixel-based

or object-based image classification approaches focus on assigning labels to individual pixels or groups of pixels (objects) within the image. These methods aim to classify each pixel or object independently based on its spectral characteristics, texture, shape, and other relevant features. Birohmatin et al. used this pixel-based or object-based image classification for the land cover analysis of Bogor City in 1996 and 2016 to study the development of the city [3].

Aerial scene classification is typically performed using deep learning models or traditional machine learning algorithms that operate on the entire image. Scene interpretation and image categorization have both been successfully implemented using deep learning, notably Convolutional Neural Networks (CNNs). Convolutional, pooling, and fully connected layers make up CNN's multi-stage, biologically inspired design, which can be effectively taught under strict supervision [4]. The classification of land use and land cover has recently gained widespread adoption. New techniques for classifying land cover and land use based on high-resolution digital aerial photography are suggested by Yang et al. [5]. While the classification of land use is based on a CNN that takes an image patch of 256 x 256 pixels and returns a land use label, the classification of land cover is based on SegNet offers a class label for each pixel, receiving an overall accuracy of 85.7%.

Taking into consideration the research published in classifying land use and the land function from various countries, this research aims to build a residential and commercial area identification system for Malaysia satellite image datasets using Convolutional Neural Networks.

Two main contributions of this research can be summarized as it:

1. proposes a new convolutional neural network model by defining its own convolution layer and dense layer for the identification of commercial and residential areas.

2. adopts the existing models Inception V3 and VGG16 to develop two transfer learning models for the identification system. The dataset on all three models was trained and tested. Accuracy, precision, and recall rates were measured and compared according to the model's performance. **The structure of this paper is as follows**:

Section 2: Literature Review provides an overview of previous studies on CNNs, transfer learning, and land use classification, highlighting key methodologies and techniques used in the field.

Section 3: Proposed Work discusses the datasets, preprocessing steps, and the design of the custom CNN model. Additionally, it explains the transfer learning models built using Inception V3 and VGG16, along with the configurations applied.

Section 4: Experiment and Results presents the experimental setup, performance metrics, and comparative results of the CNN, Inception V3, and VGG16 models. It includes accuracy, precision, recall, and F1-score analysis, along with the confusion matrices for each model.

Section 5: Conclusion summarizes the findings, highlights the best-performing model, and discusses the practical applications of the system in urban planning and land use management. Future research directions are also proposed.

II. Literature Review

A. CNN

Machine learning's potent subfield of deep learning has constantly produced state-of-the-art results on a variety of data sets. Due to their outstanding performance, Convolutional Neural Networks (CNNs), a crucial feature of deep learning, have attracted a lot of attention in the area, particularly for applications focusing on images. CNNs use a deep feed-forward architecture and outperform fully connected networks in terms of generalization. Their design is inspired by hierarchical feature detectors, allowing them to learn abstract features and efficiently identify objects [6].

CNNs differ from traditional models in that they use weight sharing, which is a defining characteristic of CNNs. Weight sharing greatly lowers the number of training parameters required, improving generalization performance and reducing overfitting concerns. The classification and feature extraction phases are also integrated by CNNs, simplifying their implementation for big networks [6].

Convolution layers, pooling layers, and fully linked layers make up CNN's basic architecture. In Figure 1, a condensed CNN architecture is shown.

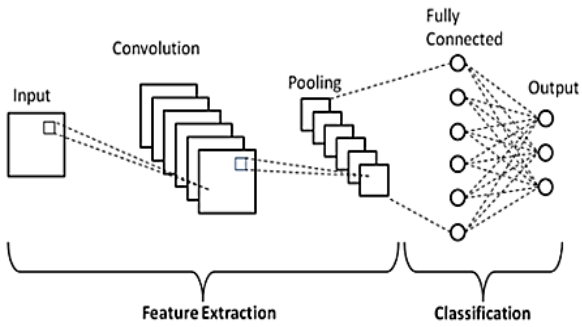

Figure 1. Basic CNN architecture for classification

In computers, images are represented as interconnected pixels. Each pixel or collection of pixels can represent different visual attributes, such as edges, shadows, or patterns. Convolutions are employed to detect these patterns effectively. The convolution layer extracts the feature from input images. It employs receptive fields to capture local correlations in the image. Each neuron in the layer is connected to neurons in the previous layer, and the weight vector associated with a receptive field is shared across all spatial locations. By sliding the weight vector over the input image, the convolution operation generates feature maps representing distinct features. This phenomenon of local receptive fields significantly reduces the number of trainable parameters and enhances feature detection capabilities.

Following the convolution layer, the pooling layer is responsible for downsampling the feature maps. It reduces the size of these maps while maintaining important information [7]. This is achieved by dividing the feature maps into windows and applying a pooling function, such as max-pooling. By selecting a window and performing pooling, the elements within the window are combined to generate an output vector. This pooling process greatly reduces the number of parameters in the network and introduces translation invariance.

In a CNN design, the completely connected layer is often found at the bottom. For CNN, it acts as the classifier. A vector created by flattening the feature maps from the convolutional or pooling layers above serves as the input to the fully connected layer [7]. The features of the input image that were extracted are shown in this vector. The final output of the CNN, which might be the predicted class or any other desired output, is represented by the output of the completely connected layer.

B. Transfer Learning

Transfer learning has emerged as an effective technique in machine learning and artificial intelligence, which allows models to leverage knowledge learned from one source domain to another target domain. In the context of convolutional neural networks (CNNs), the transfer learning process involves the use of pretrained models, such as VGG-16 and Inception V3, within convolution neural networks (CNNs) to improve the efficiency and accuracy in image classification tasks.

With transfer learning, CNN models can benefit from pretrained models that have been trained on large-scale datasets such as ImageNet. A study has been performed by Yosinki et al. (2014) on the transferability of features previously learned from the ImageNet dataset through the use of different fine-tuning strategies, and their work showed that lower layers that capture more generic features are broadly transferrable. However, higher layers which capture more specialized features are less transferable [8]. VGG16 was developed by K.Simonyan et al. (2015) and was well known for its deep architecture with 16 layers [9] and Inception V3 proposed by C.Szegedy et al. (2016) with 48 layers deep introduced the concept of inception modules to capture features at different scales [10].

Transfer learning approaches for image classification include feature extraction and fine-tuning. In the feature extraction approach, the pretrained model such as VGG16, ResNet, and Inception V3 plays the role of feature extractors by freezing the pretrained layers and allowing only the classification layers on top to be replaced or added and trained on a new dataset. A study by Razavian et al. (2014) demonstrated that superior performance can be obtained using VGG16 as a feature extractor when compared to training from scratch across several image classification tasks [11]. According to Zhang et al. (2016), their study demonstrated the efficacy of using the pretrained model ResNet as a feature extractor in task-specific and fine-grained image classification tasks [12].

In a fine-tuning approach, the pretrained layers are partially or fully unfrozen, and the entire network is then trained on the new task-specific dataset. Yosinski et al. investigated the impact of fine-tuning and found that fine-tuning the earlier layers of VGG16 could enhance performance across image classification tasks [8].

Transfer learning with VGG16 and Inception V3 has attracted wide applications in multiple image classification fields. Tan et al. (2019) demonstrated that transfer learning with VGG16 achieved state-of-the-art superior performance on the ImageNet dataset and outperformed previous approaches and models [13]. Besides that, transfer learning with VGG16 and Inception V3 as pre-trained models have been widely employed in localization tasks and object detection. The region-based CNN (R-CNN) framework was introduced by Girshick et al. (2014), which makes use of the transfer learning approach with VGG16 for object detection. The research conducted showcased outstanding performance on well-known benchmark datasets, including MS COCO and PASCAL VOC. [14].

III. PROPOSED WORK

A. Dataset Description and Preprocessing

Data sets are crucial for machine learning. They provide the foundation for training models, extracting useful features, evaluating performance, and making predictions. The image dataset was collected from two distinct sources: the Mendeley Data website and the satellite imagery relevant to Malaysia that was collected manually. The dataset includes 2500 commercial area images, 2783 dense residential area images, and 1829 sparse residential area images from Mendeley Data, along with 100 commercial area images, 105 dense residential area images, and 106 sparse residential area images that were manually collected. All the images in the dataset were resized to 224x224x3 and converted into NumPy arrays. Additionally, the pixel values were normalized to enhance the convergence and stability of the model during training. The data set was divided into 70%/30% training and testing datasets, respectively. The testing dataset was then further divided into two, so that the validation data holds 15% of overall and testing data holds another 15%. The image samples for each class in the dataset are shown in the following Figure 2.

Figure 2. Sample from the dataset: (a) commercial area, (b) dense residential area, (c) sparse residential area

B. CNN Model

The principle of Convolutional Neural Networks (CNN) has shown remarkable results in image processing and classification tasks [15]. In this project, the convolutional layers for feature extraction and nonlinear transformation were used. The convolutional layer uses a filter or convolutional kernel to slide over the input image or feature map, weighing and summing the local regions to generate a new feature map. For an input feature map I and a convolution kernel F, the convolution operation can be expressed as:

$$(I * F)_{i,j} = \sum_{m} \sum_{n} I_{i+m,j+n} \cdot F_{m,n}$$

$I_{i+m,j+n}$ denotes the value of the input feature map at the position $(i+m, j+n)$ and $F_{m,n}$ denotes the value of the convolution kernel at the position (m, n).

The CNN model design of this project also adopted the idea of the Deep Residual Network (ResNet for short). Specifically, it follows the Identity Block design of ResNet and improves on it. The core idea of ResNet is to introduce a residual mechanism in the network. The output of each convolutional layer is the sum of the input and the residual, i.e., $y = F(x) + x$, where x is the input and $F(x)$ is the residual. In this way, the gradient can be propagated directly to the next layer by short-circuiting the connection. This can effectively alleviate the common gradient vanishing problem in deep neural networks [16].

Three blocks were identified in the G1 CNN model. The identity block is a common module in the ResNet. In the identity block of this model, the input is the first batch normalized, and ReLU activated, and then the number of channels is adjusted by a 1x1 convolutional layer. Then, another batch normalization and ReLU activation are performed, and a dilated convolution layer with variable parameters extracts the features. The convolutional kernel size and dilation rate of this convolutional layer can then be used as parameter inputs, followed by another batch normalization, ReLU activation, and 1x1 convolution. The original input is also subjected to a 1x1 convolution and batch normalization to ensure that the shape is consistent with the previous convolution output, which is then summed with the convolved output and activated by the ReLU. Typical convolution operations focus only on local neighborhoods of pixels, for example, a 3x3 or 5x5 region. Dilated convolution, however, introduces a new parameter known as the dilation rate to expand the receptive field of the convolution kernel [17]. If the dilation rate is denoted as d, the distance between each element in the convolution operation becomes d. The mathematical representation of dilated convolution is presented as follows. Assume that the input is an image I of dimensions W x H, and the convolution kernel is a k x k matrix F, with a dilation rate of d. The dilated convolution operation can be represented as:

$$o_{i,j} = \sum_{m=0}^{k-1} \sum_{n=0}^{k-1} I_{i+m\cdot d, j+n\cdot d} \cdot F_{m,n}$$

In this equation, $o_{i,j}$ is an element of the output feature map. $I_{i+m\cdot d, j+n\cdot d}$ denotes the value of the input feature map at the position $(i + m \cdot d, j + n \cdot d)$ and $F_{m,n}$ denotes the value of the convolution kernel at the position (m, n). It can be seen here that the dilation convolution inserts $d - 1$ voids between each element of the convolution kernel, allowing the convolution kernel to cover a larger perceptual field. It is also worth noting that in dilated convolution, the number of elements in convolution kernel F does not increase when the dilation rate increases. In other words, the complexity of the convolution kernel F does not increase even if the dilation rate d increases. This means that although the dilated convolution increases the receptive field, it does not increase the parameters or calculation of the model. This allows the model in this research to receive more contextual information while maintaining a higher resolution, which facilitates the image classification task.

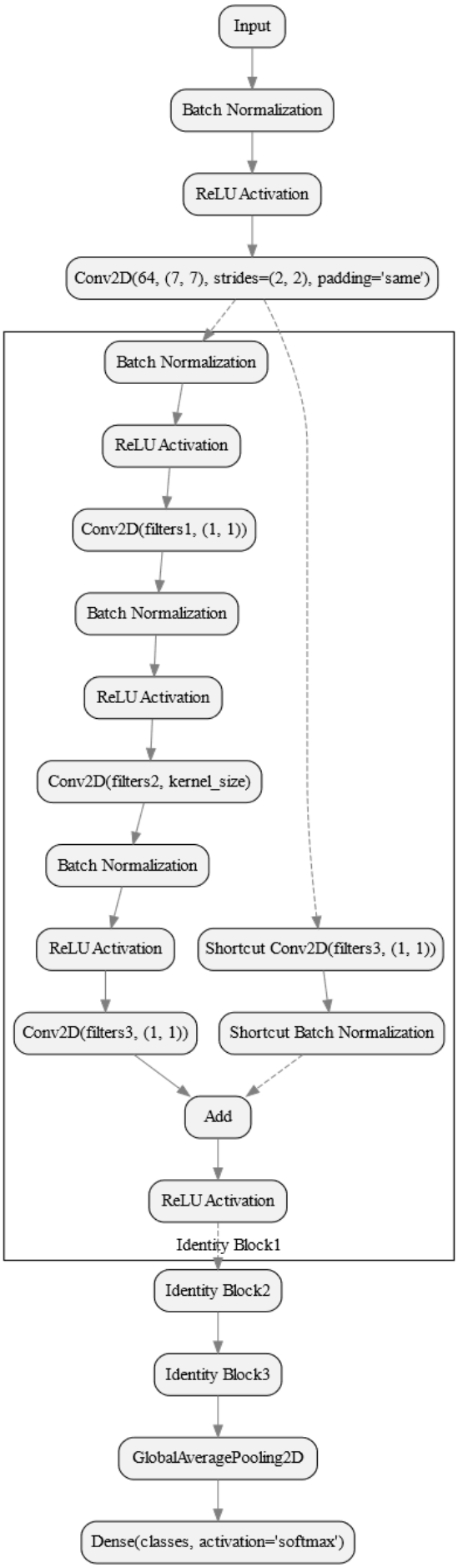

Figure 3. G1 CNN Structure

The specific structure of this project is shown in Figure 3. First, the input is batch normalized, ReLU activated, and the feature extraction is performed through a 7x7 convolutional layer. Then, three identity blocks perform the feature extraction and adjustment, with the second identity block having a convolutional kernel size of 5 and a dilation rate of 2, and the third identity block having a convolutional kernel size of 3 and a dilation rate of 3. Following this, the feature map is converted into a one-dimensional feature vector using global average pooling, and a fully connected layer is used for classification.

C. Transfer Learning using InceptionV3

The transfer learning strategy is built around the InceptionV3 model. Known for its remarkable performance in a variety of image classification tasks, this deep convolutional network design has gained widespread recognition. Convolutional, pooling, and fully connected layers, among others, are included in the numerous layers. InceptionV3 features inception modules that employ parallel convolutional operations with different kernel sizes, as shown in Figure 4 [18]. These modules facilitate the capturing of both local and global features, enabling effective learning and representation of complex patterns in images.

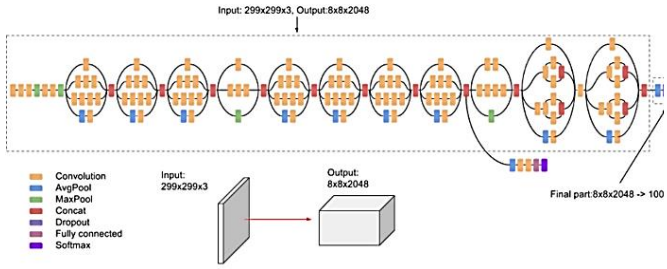

Figure 4. InceptionV3 Architecture [18]

The pre-trained layers of InceptionV3 up to the convolutional layers were obtained from Tensor flow Keras by excluding the fully connected layers. Due to the predefined input size of the InceptionV3 pretrained model as 299x299 pixels, adjustments were required for compatibility with input images of size 224x224 pixels. Therefore, the tensor input was configured to accommodate the specific dimensions of the input images. Following the base model, a global spatial average pooling layer was added. This layer reduces the spatial dimensions of the features obtained from the base model and provides a fixed-length feature vector. It enables the model to capture global information from the extracted features.

Next, a fully connected layer with 1024 nodes and ReLU activation is introduced. This dense layer serves as an intermediate step to further process and transform the features. The use of ReLU activation helps introduce non-linearity into the model. Finally, a logistic layer with softmax activation is added. This layer produces the final output predictions for the three classes: dense residential areas, sparse residential areas, and commercial areas. The softmax activation ensures that the outputs are normalized probabilities, representing the likelihood of each class.

In the conducted research, three strategies were employed to freeze and fine-tune the trainable layers of the network.

TABLE I. CONFIGURATION FOR INCEPTIONV3

Configuration	**Freeze all conv layer**	**Semi freezing**	**Unfreeze all**
Trainable Parameter	Customized fully connected layer	Half of the convolutional layers & connected layer	All Layer
Number of Trainable Parameters	2,101,251	19,468,227	23,869,603

In the first strategy, all convolutional layers, including those in the InceptionV3 base model, were frozen. The trainable parameter was set to False for these layers, ensuring that their weights remained fixed throughout training. This approach leveraged the pre-trained convolutional layers as feature extractors while training only the newly added dense layers for the specific area identification task.

The second strategy involved a semi-freezing approach, where approximately half of the convolutional layers were frozen, starting from the earlier layers of the network. This retained some fine-tuning while keeping a substantial portion of the pre-trained weights fixed. The goal was to strike a balance between leveraging the learned representations from the pre-trained layers and adapting to the specific task requirements.

In the third strategy, all layers of the network were unfrozen, including both the convolutional layers from InceptionV3 and the newly added dense layers. This allowed training the entire network from scratch using the dataset. Unfreezing all layers provided maximum flexibility for the network to learn and adapt to the area identification task, albeit requiring a larger training dataset to avoid overfitting.

Using the three different strategies, the performance of trainable layer freezing was assessed for the classification of dense residential areas, sparse residential areas, and commercial areas in satellite imagery. However, unfreezing all layers requires a larger dataset to prevent overfitting. When the dataset is limited, there is a risk of the model memorizing the training examples instead of generalizing well to unseen data.

In the context of this research, due to the limited dataset of satellite images depicting dense residential areas, sparse residential areas, and commercial areas, unfreezing all layers was not preferred. Instead, the semi-freeze approach was considered more suitable. By applying the semi-freeze strategy, the advantages of the learned representations from the pre-trained layers were retained while allowing substantial fine-tuning of the weights to adapt to the task at hand. The adoption of this approach aimed to strike a balance between leveraging the valuable features extracted by the pre-trained InceptionV3 model and enabling the network to learn task-specific patterns from the limited dataset of satellite images.

D. Transfer Learning using VGG16

Due to the enormous number of parameters and lower number of layers when compared to other deeper networks like InceptionV3 and ResNet, the VGG16 is regarded as a broad and shallow convolutional neural network. The 16 in VGG16 stands for 16 weighted layers. 13 convolutional layers, 5 max-pooling layers, and 3 dense layers make up VGG16's total of 21 layers, as depicted in Figure 5.

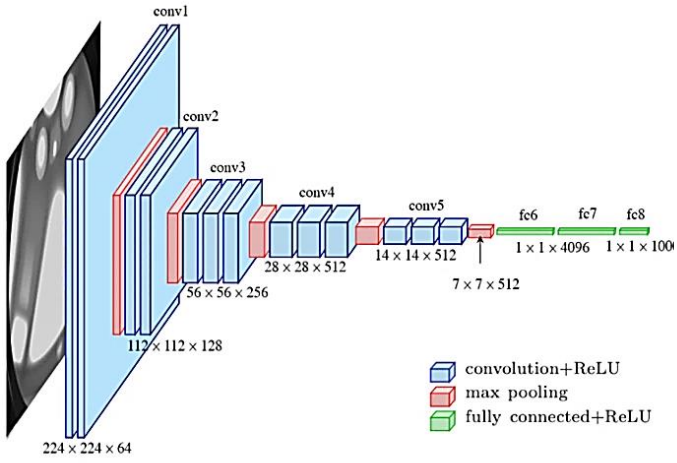

Figure 5. VGG-16 Architecture [9]

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	[(None, 224, 224, 3)]	0
block1_conv1 (Conv2D)	(None, 224, 224, 64)	1792
block1_conv2 (Conv2D)	(None, 224, 224, 64)	36928
block1_pool (MaxPooling2D)	(None, 112, 112, 64)	0
block2_conv1 (Conv2D)	(None, 112, 112, 128)	73856
block2_conv2 (Conv2D)	(None, 112, 112, 128)	147584
block2_pool (MaxPooling2D)	(None, 56, 56, 128)	0
block3_conv1 (Conv2D)	(None, 56, 56, 256)	295168
block3_conv2 (Conv2D)	(None, 56, 56, 256)	590080
block3_conv3 (Conv2D)	(None, 56, 56, 256)	590080
block3_pool (MaxPooling2D)	(None, 28, 28, 256)	0
block4_conv1 (Conv2D)	(None, 28, 28, 512)	1180160
block4_conv2 (Conv2D)	(None, 28, 28, 512)	2359808
block4_conv3 (Conv2D)	(None, 28, 28, 512)	2359808
block4_pool (MaxPooling2D)	(None, 14, 14, 512)	0
block5_conv1 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv2 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv3 (Conv2D)	(None, 14, 14, 512)	2359808
block5_pool (MaxPooling2D)	(None, 7, 7, 512)	0
global_average_pooling2d (Gl	(None, 512)	0
dense (Dense)	(None, 1024)	525312
dense_1 (Dense)	(None, 3)	3075

Total params: 15,243,075

Figure 6. Summary of the VGG16-Custom Model

However, only the convolutional layers and dense layers are considered the weight layers which are trainable and sum up to 16 layers. Before execution of the transfer learning on the VGG16 'ImageNet' pre-trained model, the original 3 dense layers (fully connected layers) from the pre-trained model are replaced with a few customized layers to form the VGG16-Custom Model for the satellite image classification task.

The customized layers include a global average pooling 2D layer, which reduces the spatial dimensions of the feature map and computes the average value for each channel, resulting in a fixed-length vector. After that, a dense layer that outputs a 1024-dimensional vector is added with the ReLU activation function applied to introduce non-linearity to the output. Lastly, the dense layer with 3 units is added as the model is going to classify 3 classes (Commercial Area, Dense Residential Area, and Sparse Residential Area). The summary of the VGG16-Custom Model network is shown in Figure 6. There are a total of 15 trainable layers with a total of 15243075 parameters in the network.

There are 3 different configurations of layer-wise fine-tuning made for the transfer learning on the VGG16-Custom Model. The differences between the 3 configurations are shown in TABLE II.

TABLE II. DIFFERENCE BETWEEN THE 3 CONFIGURATIONS OF TRANSFER LEARNING

Configuration	**Freeze All**	**Unfreeze Half of Parameters**	**Unfreeze All**
Trainable Layers	The customized layers only	Conv5 layers and customized layers	All layers
Number of Trainable Parameters	528,387	7,607,811	15,243,075

The freeze-all configuration is to only make the customized layers from the VGG16-Custom Model to be the trainable layers. The rest of the layers are frozen and the execution of the training does not change the weights and biases for those layers. This is to make full use of the pre-trained model and train on the customized classifier to solve the classification problem.

The unfreeze half of the parameters configuration is to freeze the layers before the Conv5 layers. The intention for this configuration is to train almost half of the total number of parameters (7,607,811 parameters). It is not suitable to freeze up to the center of the network straightaway (layers before Conv4) as the network goes deeper. The number of parameters also increases. Freezing the layers before Conv4 will cause most of the parameters to become trainable.

The unfreeze all configuration is basically to train all the convolutional and fully connected layers by retuning the weights and biases of the layers. This is because the satellite image datasets used in this experiment differ from the ones in ImageNet, which consists of 1000 classes of general objects. There is no satellite image in the dataset. The pre-trained weights and biases in the convolution layers might not be useful to extract the features for the satellite image to generate good classification results. Hence, no layer is frozen for this configuration.

The layer-wise fine-tuning of the VGG16-Custom Model is executed separately based on the 3 configurations stated above. The classification report of the VGG16-Custom Model for each of the configurations is recorded and then compared.

E. Hyperparameter

1) Optimizer: After constructing the G1 CNN model, the model needs to be trained to learn the intrinsic patterns of the data. Backpropagation and gradient descent algorithms were used in combination with the Adam optimizer for model optimization. The Adam optimizer combines both momentum and adaptive learning rate strategies of gradient descent to obtain faster convergence and better generalization performance [19].

2) Loss Function: The goal of model training is to minimize the cross-entropy loss function, which is commonly used in multi-classification problems. For a given true label y and model prediction $\hat{y}$, the cross-entropy loss function is defined as:

$$L = -\sum_{i} y_i \cdot \log(\hat{y}_i)$$

where i denotes the category index, y_i and $\hat{y}_i$ denote the i th element of the true label and the predicted label, respectively.

3) Batch Size: Deciding the batch size for training a CNN model is crucial as it impacts both training time and generalization of performance. After careful experimentation, a batch size of 32 was chosen for the CNN model trained on a moderate-sized dataset of approximately 7,423 images distributed across three classes. This selection aimed to strike a balance between diverse samples in each update and computational efficiency.

Using a smaller batch size could introduce more noise in gradient estimates, while a larger batch size might sacrifice generalization performance due to reduced diversity. Through empirical experimentation and training dynamics monitoring, it was determined that a batch size of 32 achieved the desired balance. It provided sufficient diversity to prevent overfitting, efficient GPU utilization, stable gradient estimates, and consistent convergence behavior, resulting in improved generalization performance.

4) Number of Epochs: Another critical hyperparameter that requires careful attention while training deep learning models is the number of epochs, which refers to the number of full iterations across the entire training dataset. The optimal number of 100 epochs is determined by taking into account a variety of parameters.

The complexity of the problem and the size of the dataset played a significant role. In the experiment, a dataset of approximately 7423 images distributed across three classes was used. While the dataset was not extremely large, it contained sufficient variability and diversity to require an adequate number of epochs for the model to capture the underlying patterns effectively.

In addition to determining the appropriate number of epochs, an early stopping strategy is introduced to further optimize the training process and prevent overfitting. If the performance does not improve after a certain number of epochs, the training process is then terminated. The training procedure will be terminated if the validation loss does not reduce for 10 consecutive epochs. The best-performing version of the model is kept by automatically restoring the model weights at the best validation loss [20].

IV. EXPERIMENT AND RESULT

A. CNN

Achieving an F1 score of 99%, the CNN model in this investigation demonstrates exemplary performance across all three categories of satellite imagery. The categorical representations are as follows: '0' for commercial zones, '1' for densely populated residential zones, and '2' for sparsely populated residential zones. The specifics of the model's performance are detailed in the subsequent classification report and confusion matrix, visually represented in the ensuing Figure 7 and 8.

	precision	recall	f1-score	support
0	0.99	0.98	0.99	390
1	0.99	0.99	0.99	433
2	0.98	0.99	0.98	290
accuracy			0.99	1113
macro avg	0.99	0.99	0.99	1113
weighted avg	0.99	0.99	0.99	1113

Figure 7. CNN Classification Report

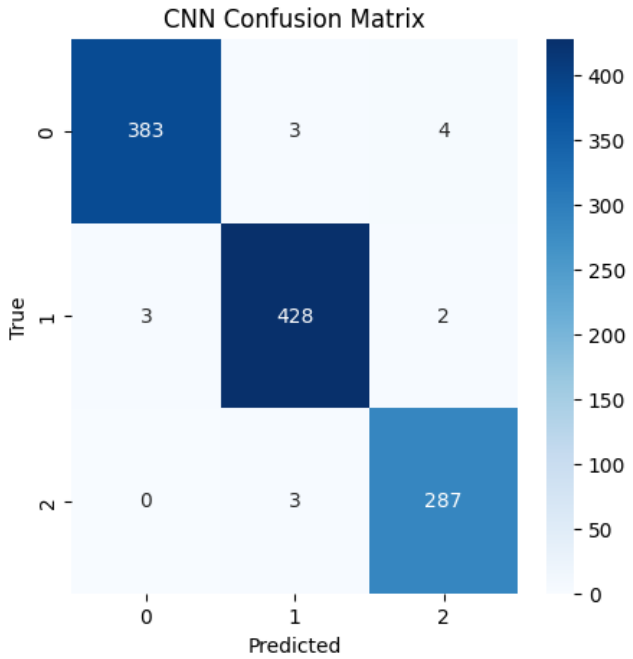

Figure 8. CNN Confusion Matrix

The precision of the CNN model's training set in this study demonstrates a consistent enhancement in correlation with an increase in the training iterations. Notwithstanding the observed intermittent fluctuations, the precision of the validation set exhibits a discernible progressive upward trajectory (see Figure 9).

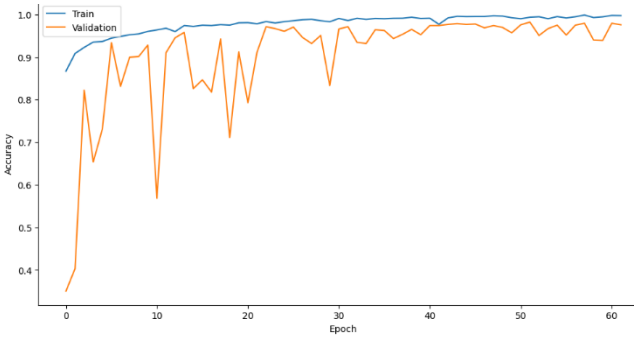

Figure 9. Accuracy on Training and Validation Set of CNN

B. *InceptionV3-Custom*

The performance of the different freezing strategies was evaluated by examining the classification report metrics, including precision, recall, and F1-score, as shown in TABLE III below.

TABLE III. CLASSIFICATION REPORT OF INCEPTIONV3-CUSTOM MODEL TRAINED UNDER THE 3 CONFIGURATIONS

Configuration	Classification Report				
Freeze all conv layer		precision	recall	f1-score	support
	0	0.98	0.97	0.98	390
	1	0.97	0.98	0.98	433
	2	0.98	0.99	0.98	290
	accuracy			0.98	1113
	macro avg	0.98	0.98	0.98	1113
	weighted avg	0.98	0.98	0.98	1113
Semi freezing		precision	recall	f1-score	support
	0	1.00	1.00	1.00	390
	1	1.00	1.00	1.00	433
	2	1.00	1.00	1.00	290
	accuracy			1.00	1113
	macro avg	1.00	1.00	1.00	1113
	weighted avg	1.00	1.00	1.00	1113
Unfreeze all conv layer		precision	recall	f1-score	support
	0	0.96	0.99	0.98	390
	1	0.99	0.97	0.98	433
	2	1.00	0.99	0.99	290
	accuracy			0.98	1113
	macro avg	0.98	0.98	0.98	1113
	weighted avg	0.98	0.98	0.98	1113

After these 3 models were built, they were evaluated by testing the model with isolated testing data to avoid overfitting. The results demonstrated that the semi-freeze approach outperformed the other strategies. With the semi-freeze approach, precision, recall, and F1-scores for dense residential areas, sparse residential areas, and commercial areas reached higher values compared to freezing all layers or unfreezing all layers. This outcome suggests that the semi-freeze approach effectively balanced leveraging the pre-trained weights while adapting to the specific task at hand, yielding more accurate and reliable classification results for the satellite imagery dataset.

The freezing strategy of unfreezing all layers yielded relatively poorer results compared to the semi-freeze approach. This outcome can be attributed to the limited size of the dataset used to train this model. Unfreezing all layers introduced a significant number of trainable parameters, increasing the model's capacity to fit the training data more closely. However, with a limited dataset, the risk of overfitting became more evident. The model might have excessively adapted to the idiosyncrasies of the training samples, resulting in reduced generalization ability and lower performance on unseen data.

Similarly, the strategy of freezing all layers demonstrated lower performance comparatively. With the pre-trained model trained on the ImageNet dataset, which primarily consists of labels related to animals and general objects, there was a disparity between the pre-trained labels and the specific labels in the satellite imagery dataset. When all layers are frozen, the model heavily relies on the learned representations from the pre-trained model, which may not directly align with the features and patterns present in satellite imagery depicting dense residential areas, sparse residential areas, and commercial areas. The pre-trained model's features may be more biased towards recognizing animals and general objects rather than the specific features relevant to this task.

Therefore, the semi-freeze approach will be the most reliable as it could benefit from the learned representations while enabling the model to adapt and learn task-specific patterns from the satellite imagery dataset in this research. The confusion matrix and accuracy of the InceptionV3 with semi-freeze will be shown in Figure 10 below:

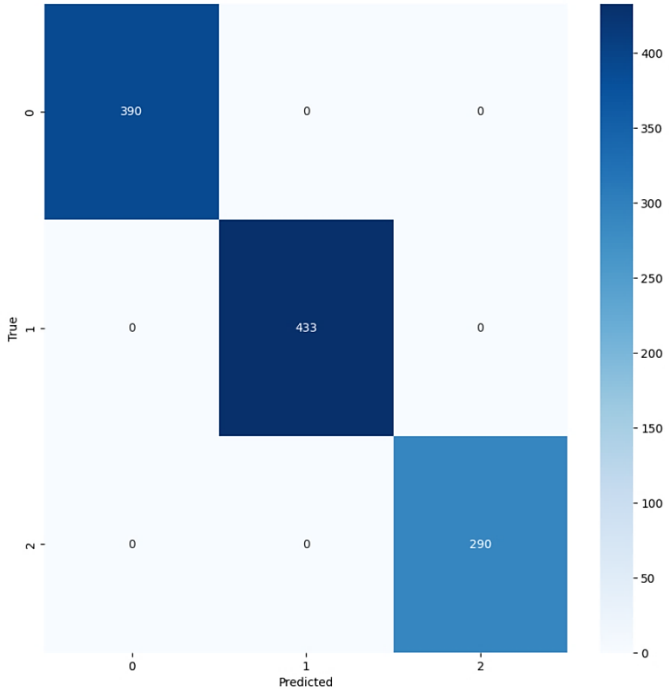

Fig 10. Confusion Matrix of InceptionV3-Custom Model Trained with Semi-Freeze Configuration

C. *VGG16-Custom*

The performance evaluation of the model trained can be determined by analyzing the classification report generated using the testing data.

TABLE IV below shows the comparison of the classification report generated after the testing phase of the trained VGG16-Custom Model under the 3 configurations of layer-wise fine-tuning.

TABLE IV. CLASSIFICATION REPORT OF VGG16-CUSTOM MODEL TRAINED UNDER THE 3 CONFIGURATIONS.

Configuration	Classification Report				
Freeze All		precision	recall	f1-score	support
	0	0.98	0.98	0.98	390
	1	0.98	0.99	0.98	433
	2	1.00	0.99	0.99	290
	accuracy			0.99	1113
	macro avg	0.99	0.99	0.99	1113
	weighted avg	0.99	0.99	0.99	1113
Unfreeze Half of Parameters		precision	recall	f1-score	support
	0	0.98	0.97	0.97	390
	1	0.98	0.97	0.98	433
	2	0.97	1.00	0.98	290
	accuracy			0.98	1113
	macro avg	0.98	0.98	0.98	1113
	weighted avg	0.98	0.98	0.98	1113
Unfreeze All		precision	recall	f1-score	support
	0	0.91	0.87	0.89	390
	1	0.81	0.90	0.85	433
	2	0.92	0.83	0.87	290
	accuracy			0.87	1113
	macro avg	0.88	0.86	0.87	1113
	weighted avg	0.87	0.87	0.87	1113

The high performance in terms of overall precision, recall, F1-score, and accuracy of the freeze-all configurations can be attributed to the fact that the fully connected layers were trained specifically for the task at hand, taking advantage of the pre-trained convolutional layers of VGG16 for feature extraction. Since the fully connected layers are closer to the classification task, they may have learned class-specific representations effectively, resulting in accurate predictions.

The performance of the unfreeze-all configuration appears to be the lowest compared to the other two, with an accuracy of 87%. The decrease in performance might be due to overfitting, as training the entire model can result in a large number of trainable parameters. With a limited amount of training data, the model may have become too specific to the training examples, leading to a drop in generalization performance on unseen data. Additionally, training the convolutional layers from scratch may require more training data or computational resources to learn meaningful representations effectively.

Overall, the results suggest that the freeze-all configuration, training only the fully connected layers, achieved the highest performance, likely due to the fine-tuning of the more task-specific layers. The unfreeze half of the parameters configuration, training layers after Conv4, also performed well but showed a slight decrease in performance, possibly due to the deeper layers requiring more data to generalize effectively.Training the entire model with all configurations unfrozen resulted in lower performance, likely due to overfitting or the need for additional training data and resources to optimize the convolutional layers effectively.

The confusion matrix and the accuracy of the VGG16-Custom Model Trained with Freeze All Configuration are shown in Figure 11to 13.

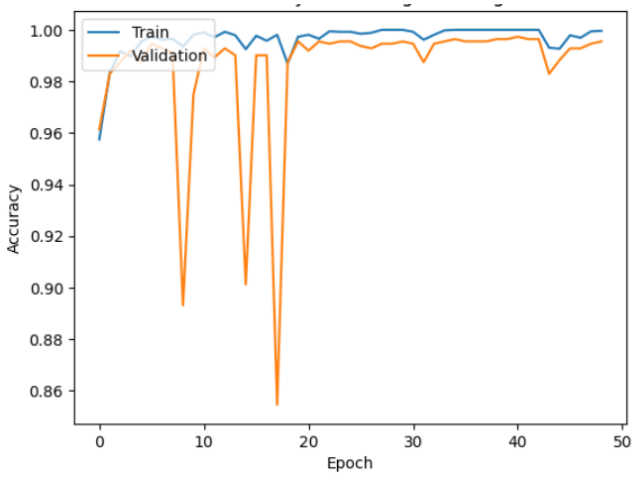

Figure 11. Accuracy on Training and Validation Set of InceptionV3-Custom Model Trained with Freeze All Configuration.

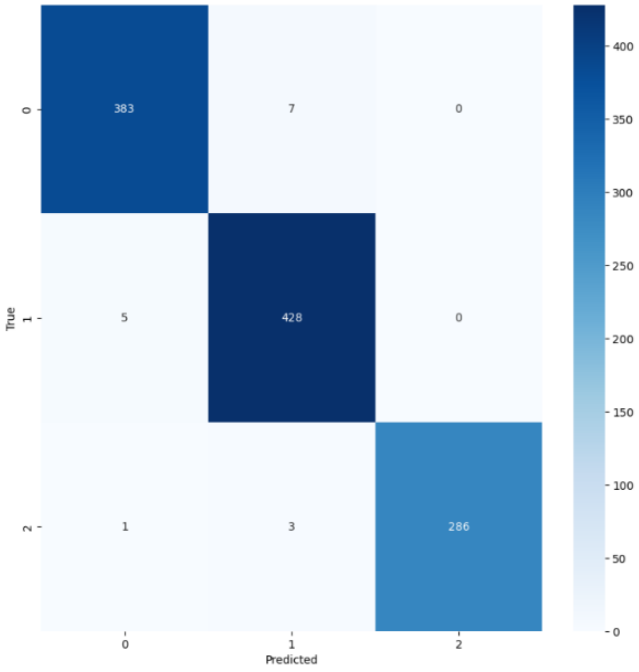

Figure 12. Confusion Matrix of VGG16-Custom Model Trained with Freeze All Configuration

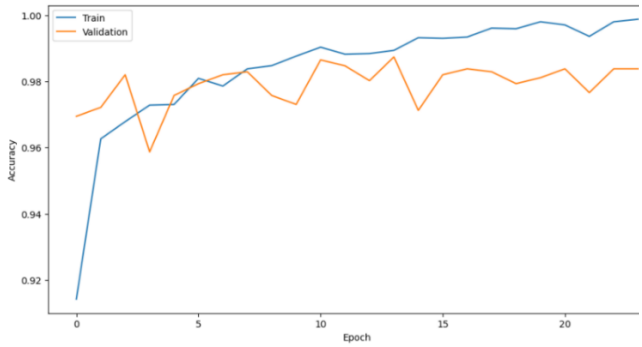

Figure 13. Accuracy on Training and Validation Set of VGG16-Custom Model Trained with Freeze All Configuration

Here is a tabular (TABLE VI) format comparing the results of different CNN models for aerial scene classification and similar tasks. The table includes the model names, key features, and performance metrics (e.g., accuracy) from relevant references.

TABLE VI. Comparisons of Benchmarking Performance

Model	Year	Key Features	Benchmarking Performance
EfficientNet	2019	Compound scaling, optimized model depth, width, resolution	Top-5 Accuracy: 97.1% on ImageNet
Vision Transformers (ViTs)	2020	Transformer architecture for image classification	Top-1 Accuracy: 88.5% on ImageNet
Hybrid CNN-Transformer	2023	Combination of CNN and Transformer models	Accuracy: 96.8% on urban land use classification
Multi-Scale CNNs	2024	Multi-scale feature extraction for varying object sizes	Accuracy: 98.2% on high-resolution satellite images

V. CONCLUSION

In this study, three models were developed for classifying dense residential areas, sparse residential areas, and commercial areas in satellite imagery. The models included a proposed CNN model, a VGG-16 model with freeze-all configuration, and the InceptionV3 model with a semi-freeze approach.

TABLE VII. Accuracy of Proposed Model and Pre-trained Models

Model	Accuracy
Proposed CNN	99.0%
InceptionV3(Customized)	100.0%
VGG16(Customized)	99.0%

The InceptionV3-based model achieved the highest accuracy with 100%, showcasing its effectiveness in inaccurate area identification. The proposed CNN model achieved an accuracy of 99%, while the VGG-16 model with freeze-all configuration achieved 99% accuracy. Three of the models demonstrated strong performance of their effectiveness for the area identification task. The highest accuracy achieved by the InceptionV3-based model can be attributed to its unique architecture, designed with multiple parallel branches that allow for the extraction of rich and diverse features at different scales and levels of abstraction. Further research can explore factors influencing these high accuracies and investigate the models' transferability to similar tasks or different regions.

CONFLICT OF INTEREST

The authors declare that there is no conflict of interest regarding the publication of this paper.

ACKNOWLEDGEMENT

We thank everyone who provided insight and expertise that greatly assisted the research.

REFERENCES

[1] Tobi, S. U. M., Jasimin, T. H., & Rani, W. N. M. W. M. (2020). Overview of Affordable Housing from Supply and Demand Context in Malaysia. In IOP Conference Series: Earth and Environmental Science (Vol. 409, No. 1, p. 012010). IOP Publishing.

[2] Xia, G. S., Hu, J., Hu, F., Shi, B., Bai, X., Zhong, Y., ... & Lu, X. (2017) outdated. AID: A benchmark data set for performance evaluation of aerial scene classification. IEEE Transactions on Geoscience and Remote Sensing, 55(7), 3965-3981.

[3] Amalisana, B., & Hernina, R. (2017, December) outdated. Land cover analysis by using pixel-based and object-based image classification methods in Bogor. In IOP Conference Series: Earth and Environmental Science (Vol. 98, No. 1, p. 012005). IOP Publishing.

[4] Hu, F., Xia, G. S., Hu, J., & Zhang, L. (2015) outdated. Transferring deep convolutional neural networks for the scene classification of high-resolution remote sensing imagery. Remote Sensing, 7(11), 14680-14707.

[5] Yang, C., Rottensteiner, F., & Heipke, C. (2018). Classification of land cover and land use based on convolutional neural networks. ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences 4 (2018), Nr. 3, 4(3), 251-258.

[6] Indolia, S., Goswami, A. K., Mishra, S. P., & Asopa, P. (2018). Conceptual understanding of convolutional neural network-a deep learning approach. Procedia computer science, 132, 679-688.

[7] Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., ... & Farhan, L. (2021). Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions. Journal of big Data, 8, 1-74.

[8] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson (2014) outdated, "How transferable are features in deep neural networks?", Advances in neural information processing systems , page 3320--3328.

[9] Simonyan, K., and A. Zisserman. “Very Deep Convolutional Networks for Large-Scale Image Recognition.” 3rd International Conference on Learning Representations (ICLR 2015) outdated, Computational and Biological Learning Society, 2015, pp. 1–14.

[10] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens and Z. Wojna, “Rethinking the Inception Architecture for Computer Vision,” 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016 outdated, pp. 2818-2826, doi: 10.1109/CVPR.2016.308.

[11] Razavian, Ali, Azizpour, HosseinSullivan, Josephine & Carlsson, Stefan. (2014) outdated. “CNN Features Off-the-Shelf: An Astounding Baseline for Recognition.”, 2014 IEEE conference on computer vision and pattern recognition workshops. 1403. 10.1109/CVPRW.2014.131.

[12] Zhang, N., et al. (2016) outdated. Deep learning vs. kernel methods: Performance for fine-grained classification. In Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR) (pp. 2166-2174).
[13] Kussul, N., Lavreniuk, M., Skakun, S., & Shelestov, A. (2017). Deep learning classification of land cover and crop types using remote sensing data. IEEE Geoscience and Remote Sensing Letters, 14(5), 778-782.
[14] Zhu, X. X., Tuia, D., Mou, L., Xia, G. S., Zhang, L., Xu, F., & Fraundorfer, F. (2017). Deep learning in remote sensing: A comprehensive review and list of resources. IEEE Geoscience and Remote Sensing Magazine, 5(4), 8-36.
[15] Huang, X., & Wang, K. (2020). Transfer learning for land use and land cover classification: A review. ISPRS Journal of Photogrammetry and Remote Sensing, 166, 104-122.
[16] Xu, Y., Xu, C., & Liu, C. (2021). Urban land use classification from high-resolution remote sensing images based on convolutional neural networks and graph embedding fusion. Remote Sensing, 13(1), 1-21.
[17] Sumbul, G., Charfuelan, M., Demir, B., & Bruzzone, L. (2019). BigEarthNet: A large-scale benchmark archive for remote sensing image understanding. IEEE International Geoscience and Remote Sensing Symposium, 5901-5904.
[18] Sun, Y., Liu, Y., & Li, H. (2022). Land use and land cover classification using transfer learning and deep convolutional neural networks. Journal of Remote Sensing, 2022, 53-64.
[19] Sherrah, J. (2016). Fully convolutional networks for dense semantic labelling of high-resolution aerial imagery. arXiv preprint.
[20] Li, W., Fu, H., Yu, L., Cracknell, A., & Gong, P. (2017). Deep learning based oil palm tree detection and counting for high-resolution remote sensing images. Remote Sensing, 9(1), 22.
[21] Zhu, Y., Zhang, Z., Lu, P., & Zhu, X. (2023). Aerial scene classification using transformer-based deep learning models and self-attention mechanisms. IEEE Transactions on Geoscience and Remote Sensing, 61, 1-12.
[22] Gong, C., Wang, F., & Yu, L. (2024). Multi-scale CNNs for land use and land cover classification in high-resolution satellite images. Remote Sensing of Environment, 298, 113457.
[23] Liu, Q., Li, W., Tang, Y., & Zhang, X. (2023). A dual-branch CNN architecture for fine-grained urban land use classification using high-resolution remote sensing images. Remote Sensing, 15(4), 1023.
[24] Chen, H., Sun, Y., & Zhang, H. (2023). Transfer learning-based CNN model for rapid urban expansion detection using multi-temporal satellite images. ISPRS Journal of Photogrammetry and Remote Sensing, 194, 88-100.
[25] Rao, D., & Patel, S. (2024). Enhanced aerial image classification using convolutional neural networks with data augmentation techniques. Journal of Applied Remote Sensing, 18(1), 012345.
[26] Mei, S., Hou, C., Li, D., & Zhang, W. (2023). Land use classification in complex urban environments with hybrid CNN-Transformer networks. IEEE Geoscience and Remote Sensing Letters, 20(6), 1234-1241.
[27] Wang, L., Xiong, Y., & Wang, J. (2023). Exploring the role of self-supervised learning in satellite image classification: A case study on urban land use detection. Remote Sensing, 15(7), 1923.