Logistics Infrastructure Classification using CNN and Transfer Learning

LEE XUN,LEONG SHIN YEE,XIE XUHUI,TAN ZI XIN,KAM SAY HONG,HUANG XIN,WANG WEN YAN

Abstract: *Logistics Infrastructure plays an important role where it can help to improve the efficiency of the distribution and the movement of the goods. However, it is crucial to be able to identify those facilities correctly through the satellite image. In this paper, we have proposed a method to classify the satellite image by using Convolutional Neural Network (CNN), VGG16 and InceptionV3.*

**Keywords:** CNN, VGG16, InceptionV3, logistics infrastructure, satellite image

## 1.0 INTRODUCTION

Nowadays, logistics infrastructure is important in supporting the distribution of goods within supply chains. There are many different types of logistics infrastructure that we can see everyday and we are able to identify easily using our eyes or through photos. However, it poses several unique challenges including large intraclass diversity and high interclass similarity, large variance of object/scene scales, and the coexistence of multiple ground objects. Within-class diversity arises from variations in appearance, style, shape, scale, and distribution of ground objects, making accurate classification difficult. Additionally, imaging conditions, such as weather and illumination, contribute to within-class diversity. Interclass similarity occurs when the same objects appear in different scene classes or when there is semantic overlap between scene categories. The ambiguous definition of scene classes further compounds interclass dissimilarity. The large variance of object/scene scales is a challenge due to varying imaging altitudes and intrinsic size variations within categories. Furthermore, remote sensing images often contain multiple ground objects, adding complexity to classification tasks. Addressing these challenges requires sophisticated algorithms and techniques, including deep learning-based methods such as autoencoders, convolutional neural networks (CNNs), and generative adversarial networks (GANs) [1].

In this project, we will propose a model that is able to identify and classify the satellite image to the correct category. A total of three categories will be used in this model which are the harbour, parking lot and container. The objective of this project is to evaluate the effectiveness of the proposed model. The performance of the models have been evaluated by accuracy to ensure that the classification model is generated with satisfactory accuracy.

## 2.0 LITERATURE REVIEW

In recent years, convolutional neural networks and transfer learning techniques have been widely used in the detection and classification of satellite images [2]. As a valuable resource for analysing and monitoring transportation and logistics infrastructure, satellite images can effectively and accurately distinguish and classify different infrastructure elements using CNN and transfer learning of deep learning, thus providing numerous applications in the field of transportation and logistics.

Shabbir, Amsa, et al. conducted a study on satellite and scene image classification, employing transfer learning and fine-tuning of deep learning models. Their approach accurately distinguished different satellite images by utilising a pre-trained model initially trained on a large-scale dataset ImageNet. They subsequently fine-tuned the model using their dataset of satellite and scene images, effectively adapting the learned features to the target domain. The results demonstrated a high classification accuracy, indicating the efficacy of their approach [3]. In another research, Kim, Youngmin, et al. classified the image data of the damaged parcel boxes through Convolutional Neural Network (CNN) in Google Colaboratory notebook, and the experiment yielded promising results. To validate the effectiveness of the model, additional experiments were conducted using other algorithms such as VGG16 and ResNet50. Finally, they got the best results using CNN, which exhibits potential for identifying damaged parcel boxes[4].

The introduction of the Inception V3 architecture by Szegedy et al. in their paper "Rethinking the Inception Architecture for Computer Vision" brought significant advancements to the field of computer vision. The authors presented an improved version of the original Inception model, incorporating innovative design elements to address the challenges of large-scale image classification tasks. Inception V3 introduced the concept of factorized convolutions, which split larger convolutions into smaller ones to reduce computational complexity while maintaining or enhancing performance. It also incorporated batch normalization and aggressive regularization techniques to enhance training stability and prevent overfitting. The authors demonstrated the effectiveness of Inception V3 on the ImageNet dataset, achieving top-5 accuracy surpassing previous state-of-the-art models. Inception V3 has since become a widely used architecture for various computer vision applications, showcasing its capability to handle complex image classification tasks effectively. [5]

The work by Simonyan and Zisserman introduced VGG16, a deep convolutional neural network architecture that has made substantial contributions to the field of computer vision. The VGG16 model garnered attention for its simplicity and depth, featuring 16 weight layers. By utilizing small 3x3 convolutional filters throughout the network, VGG16 achieved impressive performance on image recognition tasks, including the challenging ImageNet Large-Scale Visual Recognition Challenge dataset. The decision to increase the depth of the network while keeping the filter size small enabled VGG16 to learn highly discriminative and intricate features, leading to superior results compared to earlier models. Consequently, VGG16 has become widely recognized and influential, frequently employed as a feature extractor or pretraining backbone in various computer vision applications. Its remarkable accuracy and transferability have solidified its reputation as a prominent model in the field [6].

Overall, these studies highlight the potential of CNNs and transfer learning in the field of satellite image classification, especially transportation and logistics infrastructure. By leveraging these techniques, researchers can achieve accurate and efficient classification of satellite images.

## 3.0 METHODOLOGY

**3.1 Dataset Description**

The dataset consists of a total of 7813 satellite images and divided into 3 categories which are the Harbor with 2611 images, Containers with 2600 images and Parking Lot with 2602 images. The dataset is built up with 2 parts: the original dataset from Mendeley Data [7] which consists of the images in the United States and the Malaysia dataset which is collected from Google and satellite images. All the images have been resized to 224\*224 pixels. In this paper, we have split them into 3 sets which are the Training Set (60%), Validation Set (20%) and Test Set (20%).

The sample images are shown as follow:

 Aerial view of a marina with boats

Description automatically generated A group of containers in a warehouse

Description automatically generated

Figure 3.1.1 Sample images of the dataset. Starting from left: parking lot, harbour, and container

**3.2 Data Preprocessing**

The images are then being augmented to provide a more robust result. Table 1 shows the list of augmentation techniques performed using keras library.

| Parameters | Value | Description |
| --- | --- | --- |
| rotation\_range | 40 | Random rotation between -40 and +40 degrees is applied to the images. |
| width\_shift\_range | 0.2 | Random horizontal and vertical shifting of the images by a maximum of 20% of their width and height, respectively. |
| height\_shift\_rang |
| shear\_range | 0.2 | Random shearing transformation applied to the images. |
| zoom\_range | 0.2 | Random zooming applied to the images, allowing zooming in or out by a maximum of 20%. |
| horizontal\_flip | True | Random horizontal and vertical flipping of the images, respectively. |

Table 3.2.1 Data augmentation performed

**3.3 Experiment Setup**

Three models are proposed in this paper, including the CNN model, VGG16, and InceptionV3. Hyperparameter tuning are performed at the search space shown below:

| Hyperparameters | Search space |
| --- | --- |
| Learning rate | 0.1, 0.01, 0.001, 0.0001 |
| Optimizers | Adam, SGD, RMSProp, Adadelta, Adagrad, Adamax, Nadam |

Table 3.3.1 Hyperparameter search space

The hyperparameter is selected by running one epoch on each hyperparameter and selecting the one that leads to highest accuracy.

For VGG16 and Inception Model, additional trials of freezing and unfreezing layers are run, and results are being recorded. By constantly fine tuning the models, the best hyperparameter such as optimizers, learning rate, layer freezing, and number of epochs is obtained and brought forward for comparison.

**3.4 Architecture of proposed model**

**3.4.1 CNN**

In the CNN algorithm of this paper, three convolutional layers are first added through the Conv2D layer (the first convolutional layer has 128 filters, each of size 5x5; the second convolutional layer has 64 filters, each of size 3x3. The weight regularization term of this layer is L2 regularized and the regularization factor is set to 0.00005; the third convolutional layer has 32 filters, each of size 3x3, again using L2 regularisation), each convolutional layer has a different number and size of filters, and uses ReLU as the activation function. After each convolutional layer, the feature map was downsampled by a maximum pooling layer to reduce the size of the feature map. Subsequently, the data is normalized using a batch normalization layer to improve the stability and training effectiveness of the network.

Next, the multidimensional feature map is flattened into a one-dimensional vector via the Flatten layer in preparation for the fully connected layer. In the fully connected layer, there is a hidden layer with 224 neurons and an activation function of ReLU. Subsequently, the output of 50% of the neurons is randomly discarded through the Dropout layer to prevent overfitting.

Finally, there is an output layer with 4 neurons that maps the input to the probability distribution of the classification task using the softmax activation function.

**A screenshot of a computer code

Description automatically generated**

Figure 3.4.1.1 Summary of Proposed CNN Model

**3.4.2 VGG-16**

VGG-16 is used as a transfer learning model for our image recognition model. VGG-16 is a pre-trained convolutional neural network using a collection of over 14 million images with 1000 categories. It contains 13 convolutional layers and 3 fully connected layers with a total of 138 million parameters. A 3x3 filter is used for convolution, ReLu as the activation function, and 5 max-pooling layers are used in VGG-16. VGG-16 has 3 fully connected layers with the first two layers having 4096 channels each, and the third layer is the class label layer. Figure 3.4.2.1 illustrates the architecture of VGG-16. [8]

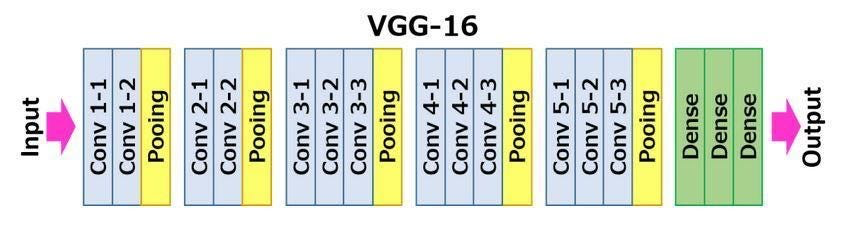


Figure 3.4.2.1 VGG-16 Architecture

In the proposed VGG-16 model, we freeze all convolutional layers and replace the label layer with 3 classes which are Container, Harbour, and Parking Lot. The features from the last max-pooling layer are flattened, followed by two dense layers. First dense layer with 4096 channels, and second dense layer with 1072 channels. Both dense layers use the ReLu activation function. Then, a Dropout layer with a drop rate of 0.2 is introduced to prevent overfitting. Finally, the SoftMax activation function is used for the class label layer in our image recognition model. Figure 3.4.2.2 shows the summary of the proposed VGG-16 model.

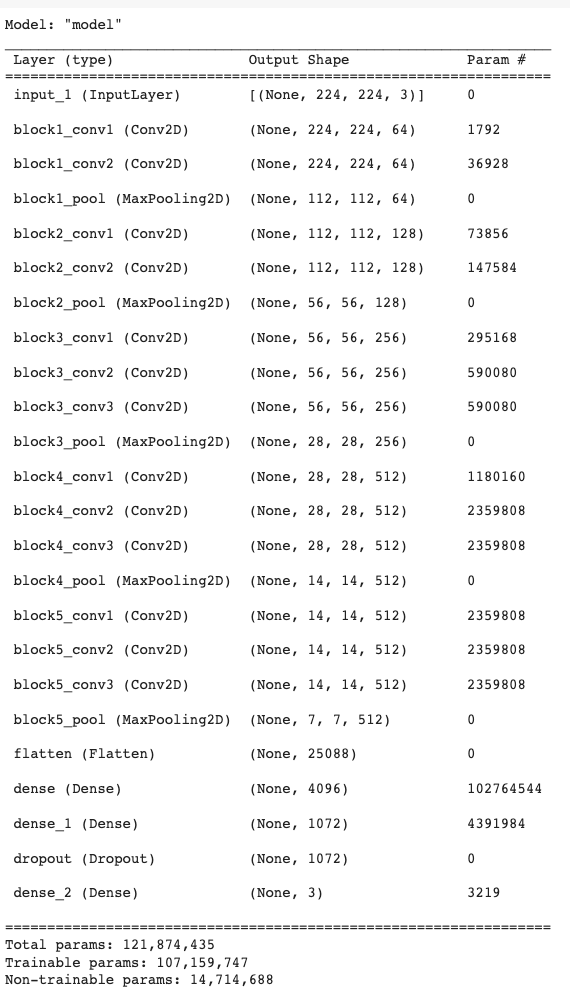


Figure 3.2.2.2 Summary of Proposed VGG-16 Model

**3.4.3 Inception**

InceptionV3 is a deep convolutional neural network architecture designed for image classification tasks. It incorporates a unique "inception module" that enables efficient and parallel processing of different filter sizes within the same layer, facilitating the capturing of both fine-grained and high-level features. This architecture utilizes multiple layers with varying receptive fields, allowing for hierarchical feature extraction and representation learning. InceptionV3 also incorporates techniques like batch normalization and regularization to enhance training stability and prevent overfitting. With its innovative design and advanced features, InceptionV3 has demonstrated impressive performance on various image recognition benchmarks and has become a popular choice in computer vision research and applications [5].

**A diagram of a block diagram

Description automatically generated**

Figure 3.4.3.1 The Framework of Xception

Inception is able to classify images into 1000 objectives where it is able to have rich features representation for a wide range of images. Figure 3.4.3.1 also shows that the input of the image for the Inception model is 299x299x3 where it uses 3x3 convolution layers.

The proposed Inception Model has frozen all convolutional layers and the classifiers labels have been changed into 3 layers. The data have been processed through 94 convolutional layers that freeze in Inception Model, then it will spatial pooling by the average value across each feature map's spatial dimensions to reduce it spatial dimensions before it process by the Dense Layer with 1024 neurons and the ReLU activation function. Lastly, the SoftMax function will distribute the input images into our preset labels.

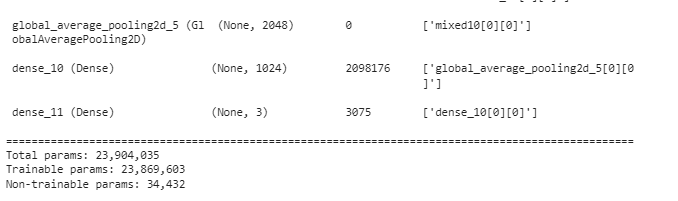


Figure 3.4.3.2 The Output of Inception Model

**3.5 Evaluation metrics**

In this project, accuracy is used to evaluate the performance of the model. Given the dataset is balanced, accuracy can provide a fair comparison among the models. The formula of accuracy is given as:

where TP = True Positive, TN = True Negative, FP = False Positive, FN = False Negative

## 4.0 RESULT & DISCUSSION

We had trained all the models with different optimizers, learning rates, and number of epochs to select the best model. Table 4.0.1 below records the best result and its parameters from each respective model.

| **Model** | **Total Parameters** | **Total Layers (Convo + FC)** | **Optimizer** | **Learning rate** | **Epochs** | **Freeze layer** | **Test** | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Loss** | **Accuracy (%)** |
| CNN | 5,641,699 | 3 + 2 | Adagrad | 0.001 | 20 | NA | 0.0743 | 98.08 |
| VGG-16 | 121,874,435 | 13 + 3 | Adamax | 0.001 | 20 | Freeze all convo layer | 0.0477 | 98.91 |
| Inception | 23,904,035 | 94 + 2 | Adadelta | 0.001 | 20 | Freeze all convo layer | 0.0478 | 99.29 |

\*Convo represents convolutional layer, FC represents Fully Connected Layer

Table 4.0.1 Loss and Accuracy of Each Model

Based on Table 4.0.1, we can observe that the InceptionV3 model achieves the highest accuracy with 99.29% accuracy followed by VGG16 (98.91%) and CNN model (98.08%). Comparing the training and testing accuracy, we can confirm that there are no overfitting occurs.

The high accuracy of Inception V3 may be attributed to:

1. Dimensionality Reduction: Inception architectures use dimensionality reduction techniques, such as 1x1 convolutions, to reduce the number of input feature maps and computational complexity. This allows the network to capture both local and global information effectively, enabling better feature representation [9].
2. Multiple Filter Sizes: Inception architectures utilize parallel convolutional layers with different filter sizes (e.g., 1x1, 3x3, 5x5). This allows the network to capture features at different scales and resolutions, enabling better modeling of various object sizes and shapes [9].
3. Regularization: Inception architectures incorporate regularization techniques, such as batch normalization and dropout, which help in reducing overfitting and improving generalization performance [10].

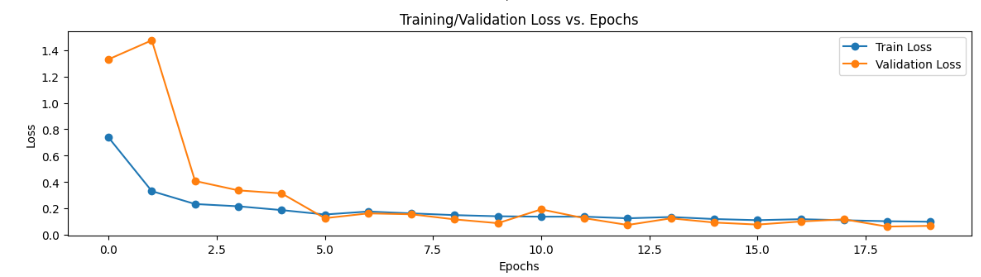
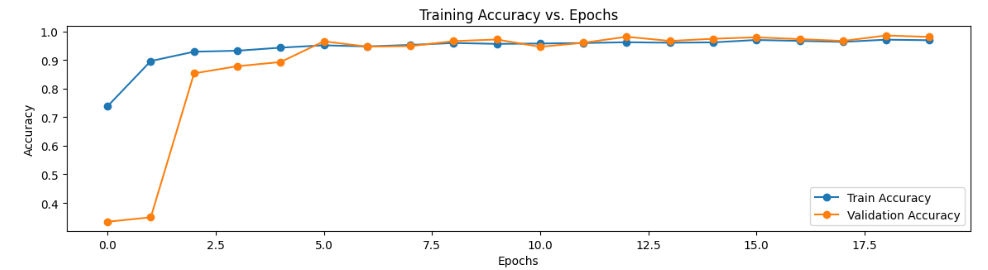


Figure 4.0.1 Graphs depict the training / validation accuracy and loss across epochs in proposed CNN model

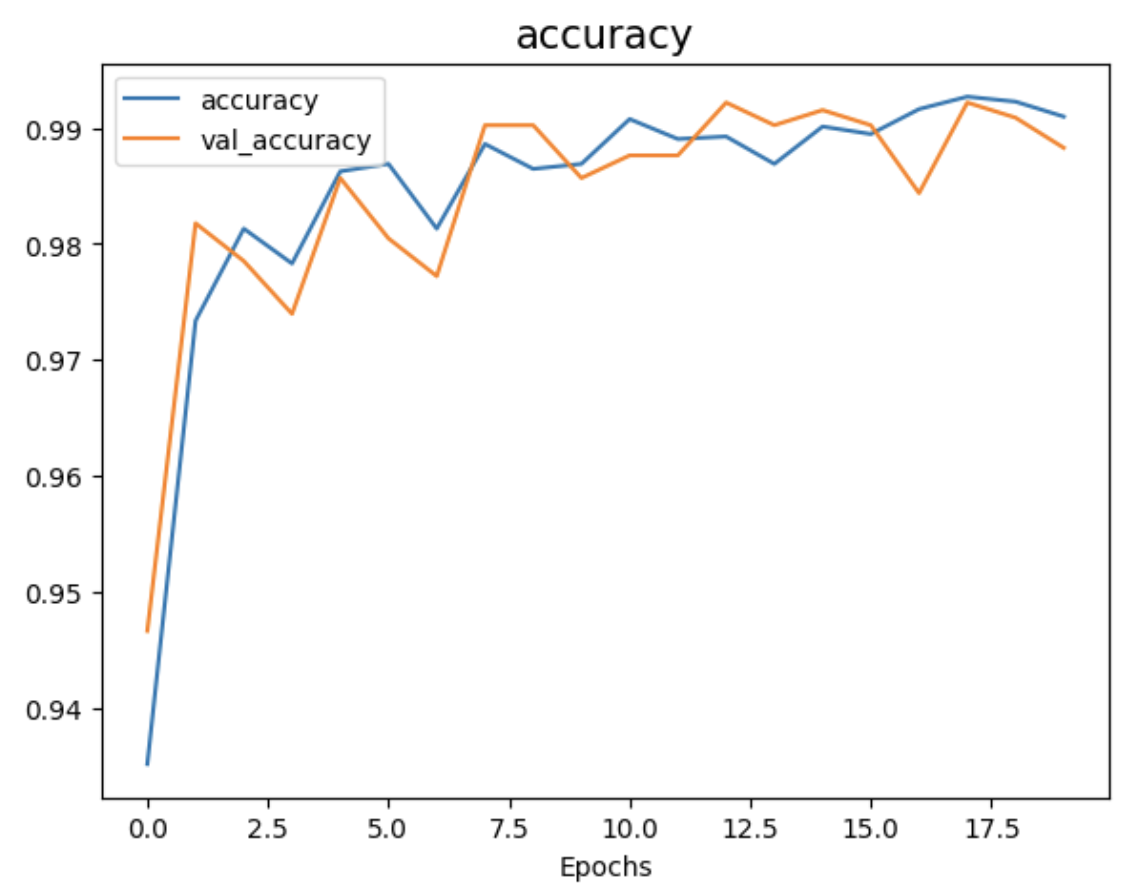
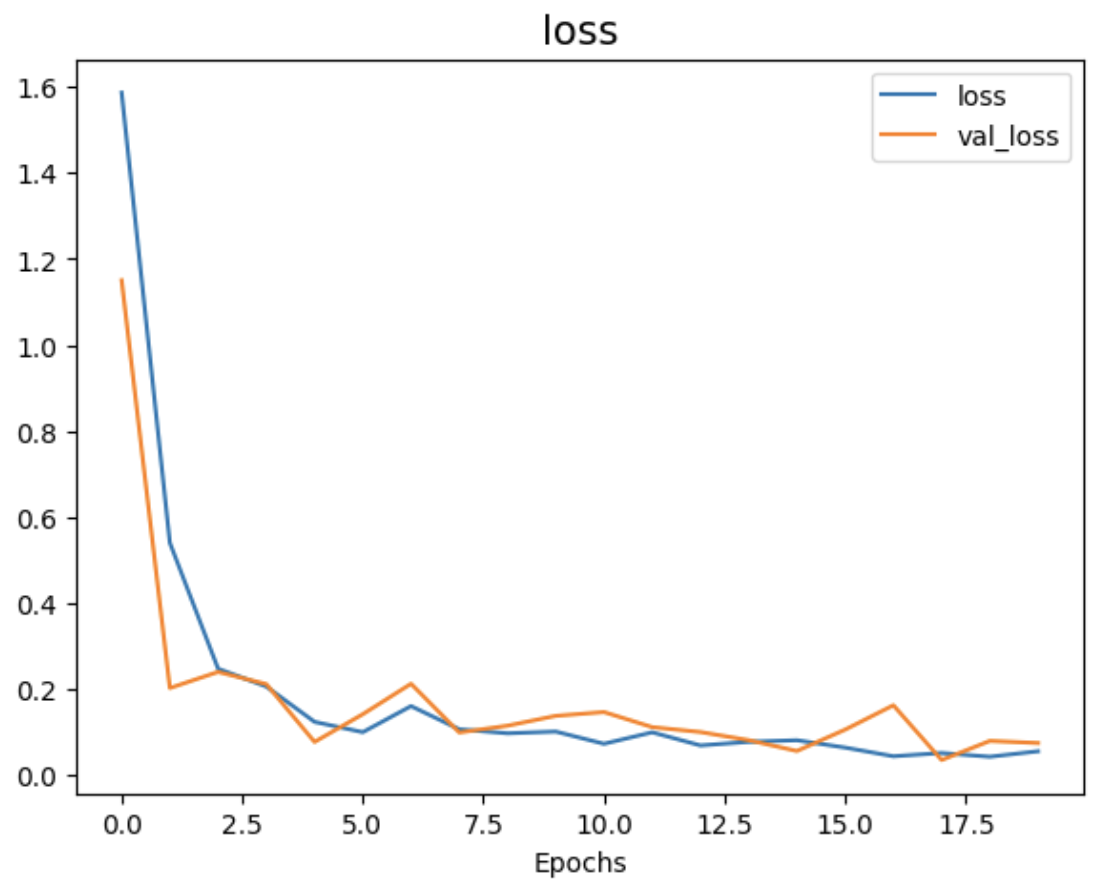
 

Figure 4.0.2 Graphs depict the training / validation accuracy and loss across epochs in proposed VGG16 model

A graph with blue and orange lines

Description automatically generated A graph of a training and validation loss

Description automatically generated

Figure 4.0.3 Graphs depict the training / validation accuracy and loss across epochs in proposed InceptionV3 model

Based on Figure 4.0.1, Figure 4.0.2 and Figure 4.0.3, we can observe that the model with deeper architecture tends to reach convergence faster than shallow architecture. This may due to the model with deeper architecture enable the extraction of features at multiple levels of abstraction. Lower layers capture low-level features like edges and textures, while higher layers capture higher-level features like shapes and objects. This hierarchical feature extraction helps the model to understand and represent the data in a more comprehensive and expressive manner.

However, it is necessary to note that our proposed CNN has 5.6 millions parameters which indicates that it requires less computational time and resources compared to the VGG16 (121 millions) and Inception V3 model (23 millions). With its fairly well accuracy, one can use our proposed CNN model if there are restrictions in computational resources.

## 5.0 CONCLUSION

In this project, a logistic infrastructure satellite image classification model has been proposed which is using the CNN, VGG16 and InceptionV3. All models have achieved a fairly good accuracy, with Inception V3 having the highest accuracy (99.29%), followed by VGG16 (98.91%) and CNN model (98.48%). With these high accuracies the models can be implemented to use for classification. However, it is worth to note that our proposed CNN model has the least parameters that lead to efficient computing speed and memory usage. In future research, the project can be extended by augmenting the dataset to encompass a broader range of categories relevant to the research scope.

## 

## REFERENCES

[1] Cheng, G., Xie, X., Han, J., Guo, L., & Xia, G. S. (2020). Remote sensing image scene classification meets deep learning: Challenges, methods, benchmarks, and opportunities. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *13*, 3735-3756.

[2] Brewer, Ethan, et al. "Predicting road quality using high resolution satellite imagery: A transfer learning approach." Plos one 16.7 (2021): e0253370.

[3] Shabbir, Amsa, et al. "Satellite and scene image classification based on transfer learning and fine tuning of ResNet50." Mathematical Problems in Engineering 2021 (2021): 1-18.

[4] Kim, Youngmin, et al. "Delivery Invoice Information Classification System for Joint Courier Logistics Infrastructure." Computers, Materials & Continua 75.2 (2023).

[5] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2818-2826).

[6] Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

[7] Qi, Xiaoman; Zhu, Panpan; Wang, Yuebin; Zhang, Liqiang; Peng, Junhuan; Wu, Mengfan; Chen, Jialong; Zhao, Xudong; Zang, Ning; Mathiopoulos, P.Takis (2021), “MLRSNet: A Multi-label High Spatial Resolution Remote Sensing Dataset for Semantic Scene Understanding”, Mendeley Data, V3, doi: 10.17632/7j9bv9vwsx.3

[8] Great Learning, “Everything you need to know about VGG16,” Medium, https://medium.com/@mygreatlearning/everything-you-need-to-know-about-vgg16-7315defb5918 (accessed Jul. 7, 2023).

[9] Garcia-Garcia, A., Orts-Escolano, S., Oprea, S., Villena-Martinez, V., & Garcia-Rodriguez, J. (2017). A review on deep learning techniques applied to semantic segmentation. *arXiv preprint arXiv:1704.06857*.

[10] Yu, Z., Dong, Y., Cheng, J., Sun, M., & Su, F. (2022). Research on Face Recognition Classification Based on Improved GoogleNet. *Security and Communication Networks*, *2022*, 1-6.

[11] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). Imagenet classification with deep convolutional neural networks. *Communications of the ACM, 60*(6), 84-90.

