


Satellite image segmentation using UNet++ with Vgg19 deep learning model

 Yaragorla Raju^{1*}, M. Narayana²

^{1,2}Department of Electronics and Communication Engineering, Anurag University, Venkatapur, Ghatkesar Rd, Hyderabad, Telangana, 500088; raju.everest8848@gmail.com (Y.R.) narayanaece@anurag.edu.in (M.N.)

Abstract: Satellite image segmentation is an essential step in many applications, including urban planning, disaster response and environmental monitoring. The problem though is that existing methods suffer from high failure rates because of the intrinsic complexity and variation found in satellite images. This research uses deep learning UNet++ and Vgg19 models to construct advanced satellite image segmentation method as we propose a brand-new approach of our own. In this study, the effective segmentation method combines the powerful feature extraction capability of Vgg19 model which is improved version based on Unet++ approach and data route aggregation module are adopted to provide complex detail in satellite images and contextual information. Implementing deep network models using known architectures will help increase accuracy and efficiency in situations where training datasets can be limited. Apparently, the approach was tested and bench-marked over a set of datasets for several visual contexts confirmed by extensive testing which resulted in an increased precision, recall along with F1 scores.

Keywords: Deep learning, Satellite image segmentation, Semantic segmentation, Unet++, VGG19.

1. Introduction

Understanding of satellite image segmentation is essential for geospatial data analysis and remote sensing. This consists of segmenting satellite images to break them down into contextual patches that correspond to a particular type of object in the scene [1]. Applications of this segmentation process are wide-ranging and include land cover classification, urban planning, disaster management and environmental monitoring [2]. Ultimately, our goal was to automatically read satellite data (data interpretation and analysis) where given a image all the pixels are classified into certain classes namely vegetation, buildings, water bodies, roads & barren land [3]. Satellite image segmentation, as the basic component of many fields relying on geospatial data for decision-making purposes, is highly effective in various applications whose precision directly influences the overall result.

Ultimately, our goal was to automatically read satellite data where given a image all the pixels are classified into certain classes namely vegetation, buildings, water bodies, roads & barren land [3]. Satellite image segmentation, as the basic component of many fields relying on geospatial data for decision-making purposes, is highly effective in various applications whose precision directly influences the overall result. This is a result of differences in captured satellite images, as they are taken at different altitudes and under various environmental conditions; hence one algorithmic approach may not work well across all datasets. This also leads to the need for powerful algorithms that work with a large input dataset, which can truly represent slight variations in pixel intensities.

One of the major limitations in segmentation of satellite images is their inconsistent occurrence. There is high variation of images even within the same class due to unique geographical features, weather patterns or just different sensors used [5]. This unpredictability has implications for traditional image processing techniques, which require hand-crafted rules and functions to specify the boundaries of objects [6].

The other issue that is seen with satellite image segmentation is the problem of handling so much data. Thousands or more of satellite images are produced, and as a time-consuming task has to be manually labelled for training models in machine learning [7]. This not only makes labelled data scarce, an essential ingredient to train regular supervised machine learning models. To address this issue, automated and scalable solutions should be used to make use of the large amount of unlabeled experience data [8]. These works can remedy the production of incomplete pictures and improve segmentation efficiency in novel or difficult situations.

This is where deep learning excels in solving the problem of satellite image segmentation. In deep learning, Convolutional Neural Networks (CNNs) have demonstrated impressive results by being able to learn hierarchical representations from visual data. Deep learning is unique among traditional methods as it does not require complicated handmade feature extraction process and allows the network to learn discriminative features directly from pixel intensities. This article presents a new fusion of modern deep learning frameworks like UNet++ and Vgg19 for improving satellite image segmentation in this perspective. The Vgg19 model helps boost accuracy of the system to a good extent in settings where our data happens to be less, by using pre-trained weights with small input perturbations for noise finding neural networks up to this point. Utilizing this blend of UNet++ and Vgg19 is a very productive way for aerial image segmentation accurately.

2. Literature

Kunhao Yuan et al [9] proposed an original DCNN model MC-WBDN (multichannel water body detection network) which outperforms the state-of-the-art DCNN-based WBD techniques by utilizing these three advanced features: a multichannel fusion module, Enhanced Atrous Spatial Pyramid Pooling and Space-to-Depth/Depth-to-Space operations.

Zhuokun Pan et al [10] presented the urban village mapping paradigm which is build by U-net deep learning architecture. The study was carried out in Guangzhou City, China. For this study, worldview satellite imagery with 8 pan-sharpened band is employed at spatial resolution of 0.5 m and building boundary vector layer used in the research process. This world image scene encompasses ten sites of urban villages. Urban village six and four selected sites were subjected to training and testing of the deep neural network model. Both building segmentation and classification models are trained and tested.

Ramesh Raghavan et al [11] demonstrated approach enhances the convolutional neural network for pixel-level image classification. Meanwhile, the Mask-RCNN (Mask Region-based Convolutional Neural Networks) performs image extraction and the masking problem which can be resolved so that it is easier and faster in preventing this matter. It involves training the model with a much larger set of data points, which makes it more complicated. Finally, a sophisticated image augmentation technique has been added in the preprocessing to enhance this algorithm towards scalability.

Jakub Nalepa et al [12] provide a series of simulated scenarios based on different atmospheric conditions and noise pollution that might occur in any future imaging satellite. In this paper, we also demonstrate the impact of these corrections on spectral-spectral and spectral-spatial convolutional neural networks generalization performance in hyper-spectral image segmentation.

Thanh Tam Nguyen et al [13] employed a spatio-temporal-spectral deep neural network of high spatial resolution imagery at the pixel level for an entire year and each event in time provides a novel multi-time high-resolution classification method to map paddy fields. The Landsat 8 data is used for our case study as it has exceptional spatial resolution which supports development of the technique and evaluation.

Sultan Daud Khan et al [14] designed a novel hybrid deep learning model which merges the characteristics of DenseNet and U-Net architectures. This categorization is carried out to pixelwise cover land. By using long-range connections of U-Net to concatenate the encoder and decoder paths; keeping some low-level features, we achieve padding information detection results that are better than existing methods.

BipulNeupane et al [15] analyzed and perform meta-analyses of the recent publications on research questions, data sources pre-processing-augmentation techniques training settings architectures, backbones, frameworks, optimizers, loss functions, other hyper-parameters performance measurement.

Arsalan Tahir et al [16] specific to assessing the state-of-play with deep learning algorithms at recognising objects within satellite imagery. Thus a dataset of satellite imagery is made using the convolutional neural network based framework; faster RCNN (faster region-based convolutional neural networks), YOLO, SSD (single shot detector) and other frameworks like SIMRDWN (Satellite Imagery Multiscale Rapid Detection with Windowed Networks) to do object detection. In addition, we use the set of satellite images created to test different blasting methods in terms of quality and speed.

Dmitry Rashkovetsky et al [17] detailed a means to leverage satellite imagery acquired in visible, infrared and microwave regions of the spectrum for differentiating fire-affected areas. We applied this workflow to assess the suitability for fire detection in four open-access satellite data sources: Sentinel 1 (C-SAR); Sentinel-2 multispectral; Sentinel -3 sea and land surface temperature; Terra / Aqua MODIS. He retrained the same single-input convolutional neural network on a new dataset (new data) for each of these before testing them on held-out test samples.

Ekrem Saralioglu et al [18] employed a 3D-2D CNN network, which incorporates both spectral and spatial information to help generate accurate land cover maps out of GSD satellite images. Based on the SH model for landscape, a Worldview-2 satellite image was used that incorporated 9 land use categories (houses and highways) in addition to ORLs depicting product agricultural areas along with tea farms as well as hazelnut plantations.

3. Proposed Method

This proposed methodology is based on the integration of VGG19 deep learning model with UNet++ Framework in order to segment satellite images precisely. It combines VGG19 and UNet++, two state-of-the-art deep-learning frameworks to deal with the intricacies of satellite imagery. The critical issues include the presence of diverse lighting conditions, different terrains and required accuracies for Earth surface structures extractions. The method exploits these architectures to improve the performance of impact segmentation, which is an early developmental stage that has so far proven challenging as it exhibits high detail in satellite imagery.

3.1 U-Net++

UNet++, also called UNet ++ or Nested U-Net, is an improved version of the original architecture which uses more complex segmentation tasks. It is able to keep the basic ideas of U-Net and improve in various forms. Moreover, the proposed U-Net++ alterations are practical and can tune-in better to complex segmentation problems especially in medical images type for critical areas. The architecture now includes extra pathways and dense connections, which help improve feature extraction as well provide better segmentation results in non-homogeneous space.

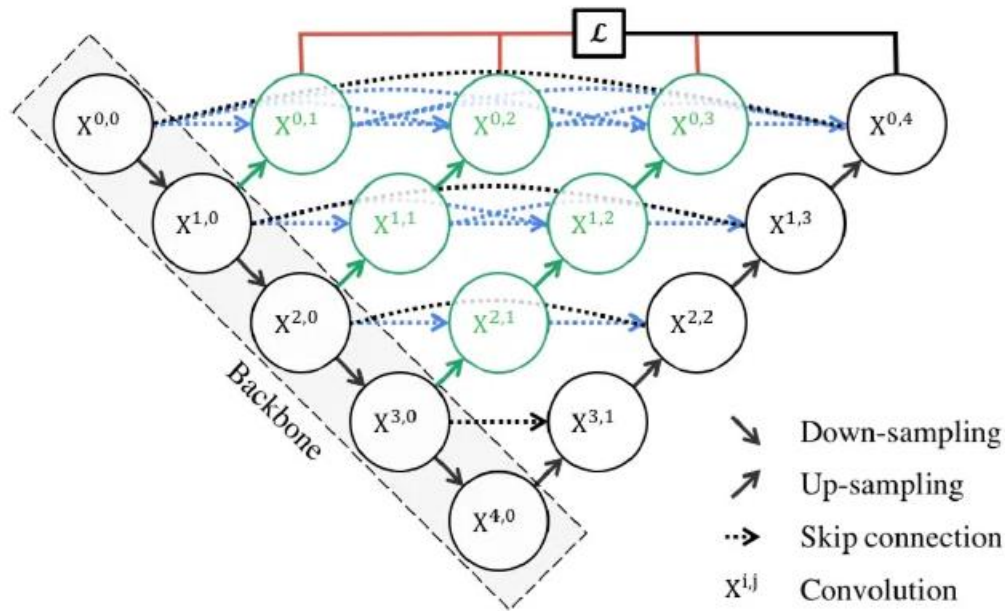


Figure 1.
General architecture of U-Net++.

U-Net++ still sticks to the core encoder-decoder architecture of U-net and makes modifications for better segmentation performance. To this end, a major change is the use of multi-scale skip connections that help capturing local and global context more robustly. This enables U-Net++ to carry out more precise segmentations by taking both fine details and larger imaging context into account. The encoder is responsible for down-scaling the input image using convolutional and pooling layers to extract hierarchical features across various levels. The decoder then upscale these feature maps to give the final reconstruction of segmentation mask and improve it further upon iteration. This basic structure is similar to the U-Net, while in case of U-Net++, there are additional connections providing a more easy flow of information through-out the network. It directly leads to more effective use of data from various image regions and improves the segmentation accuracy.

The key innovation in U-Net++ is its nested architecture, where each encoder-decoder stage consists of stacked UNet modules. These multi-scale skip connections enable the network to collect and aggregate information from different levels of abstraction for enhanced segmentation performance when objects are in various scales. In the decoding phase feature maps, produced by corresponding encoder stages are aggregated to produce more information for segmentation. U-Net++ build a network with feature from different scale concatenated such that it understand both fine, detailed and more abstract stuff at high-level. With such layer-by-layer design, the network can perform even better in segmentation on complex structures. Overall, U-Net++ is specifically designed to enhance the segmentation potential of neural networks and seems also well suited for tasks involve multi-scale feature integration that require precise boundaries.

3.1.1. Functionality

In straight words, U-Net++ is focusing on extracting most local and global guidance from different regions of an input image. This is achieved by using multi-scale skip connections as well stacking multiple U-Net modules, in order to help the model better capture spatiotemporal orientation and structure of objects that change rapidly with size or shape during its progress. With the help of these multi-level skip connections in both up-sampling and downscaling processes, it reduces loss information by maintaining among original image features at different scaling leading to a perceptible increase in pixel-wise segmentation accuracy.

U-Net++ is a more hierarchical design model and it can capture long-range dependencies of image information. Thus, U-Net++ can capture both global and local contexts of different spatial distributions in the images to adapt to changes such as target object size, location or appearance. In addition, this network has multi-scale skip connections that benefit segmentation tasks involving: a) detecting fine and coarse objects, and b) simple or complex object shapes.

The U-Net++ is also shown to perform very well in other biomedical image segmentation tasks and it was applied actively not only semantic segmentation but also satellite imagery interpretation, remote sensing application or natural scene analysis area. It has been the tool of choice for segmentation among researchers and practitioners due to how flexible, whilst still efficient it is. U-net++ is a better U-net model than the original one and resolves some limitations about more context powerful/better segmentation accuracy. It is not just suitable for computer vision tasks but also can be applied to many medical image domains supported by its architecture.

3.2. VGG19

The VGG19 is a deep convolutional neural network (CNN) from the Visual Geometry Group. An extension of the VGG16 architecture with additional higher resolution layers and a downscaled pooling operation to achieve better performance for image recognition, made by Karen Simonyan and Andrew Zisserman in "Very Deep Convolutional Networks for Large-Scale Image Recognition" (2014).

Detailed analysis of VGG19 architecture is given here:

3.3. Input Layer

The input layer of VGG19 takes care to receive and process the image data before it is passed through The Network. VGG19 model was developed with the goal that it should work on input images of size 224×224 pixels allowing a more consistent way to create and use image data sets. Images are either in colour (RGB three channels) or gray scale (single channel). RGB images have the shape $224 \times 224 \times 3$ where each of red, green or blue channels has a dimensionality for RGB colour maps and gray scale is in form of one 2D array with dimensions (also flattened) to make its appearance similar. We use the input layer so that image size is always fixed when every input goes through these; it makes training as well inference simple.

After resizing the input images, RGB pixel values are normalized and turned into numerical tensors. Basically each pixel in the image becomes a input feature, where its intensity level being treated as value of that feature. This input tensor is then flowed through the convolutional layers which initiates feature extraction. This is an important contribution to the model since it guarantees that all images are processed equally, promoting thus some concretion with those images which come from a different dataset. VGG19 is able to outperform across a spectrum of visual recognition tasks due the fact its input size and format are standardised, freeing up resources in training for learning patterns opposed to adjusting weights at each layer.

3.3.1. Convolutional Layers

The convolutional layers form the base of VGG19 and take care for feature extraction from input images. Convolutional layers hold learnable weights by executing several filters or kernels on each image which in the process will learn 2D patterns (like edges, gradients) and so forth. VGG19 uses small 3×3 size filters which allow them to capture the spatial pattern within an image. These filters are then drawn across the entire image one by one, where they create feature maps that give high importance to a particular structure such as an edge or curve. Convolutional layers operate in blocks. There can be any number of convolutions stacked together forming a block. These layers capture more complex features as they attune to the progression, capturing finer and finer patterns ranging from simple edges at very early levels of layer 1 up to parts of objects in higher level concepts.

Convolutional layers in the VGG19 model are accompanied by an activation function, Rectified Linear Unit (ReLU), which injects non-linearity into our architecture. This non-linearity is essential for the neural network to learn intricate and abstract patterns which would not be able without adding up all of those linear transformations. Due to the presence of several convolutional layers and ReLU

activation, VGG19 is able to automatically learn a hierarchy of features from simple edge patterns at lower levels to more sophisticated texture representations at deeper layers. Each After each convolutional block, we have a max-pooling layer that is used to reduce the dimensionality of the feature maps but retaining relevant information. This architecture makes VGG19 very good at image recognition and is widely used in tasks like object detection, Image classification ETC.

3.3.2. MaxPooling Layers

MaxPooling MaxPooling layers are an essential ingredient of the VGG19 architecture, responsible for down-sampling feature maps along their spatial dimensions while maintaining only their most prominent features. Max-polling layers are used for sub-sampling the feature maps after every convolutional block. This layer works by taking the max value from a 2×2 region in the feature map hence reducing its height and weight to half. The max-pooling layer, by downsampling the feature maps for us, greatly reduces the computational load on our network and at the same time helps in retaining key features required to classify any image correctly. This process helps by making the network learn spatial hierarchies again and makes it generalize well across different image scales.

Using max-pooling also results in more translation invariance, i.e., the network becomes less sensitive to where objects are located (but it still has no idea what object is there). So max-pooling helps to make our feature robust and discriminatory by only focusing on like important features: things such as edges or shapes etc. VGG19 utilised max-pooling layers between convolutional blocks to enable the network to induce spatial invariance over higher and deeper feature space I formation. This balance of dimensionality reduction and feature extraction allows VGG19 to work well with large, high-dimensional data sets which we have in visual tasks.

3.3.3. Fully Connected Layers

The fully connected layers in VGG19 are located at the last stage of the neural network, and they utilize features which were extracted by convolutional stages to make decisions on if a region belongs to one class or not. These layers are fully connected, which means there is a connection from each neuron in one layer to all neurons on the next. This dense connectivity helps the network to aggregate and make sense of hierarchical representations learned by convolutional layers. The first two fully connected layers in VGG19 have 4096 neurons each, and act to reduce the high-dimensional feature representations into a more compressed form. This will help the network to learn patterns, interactions between features which are important for accurate prediction.

The final fully connected layer of VGG 19 is the 1000 neurons, which corresponds to its output image classes when it was trained with ImageNet dataset. The resulting value is a probability distribution represented by softmax function and the sum of their probabilities will be 1. It helps the model to give a probability score regarding each class that how much chance is there for an input image of certain category. Those fully connected layers are essential and take all the learned features convert them into one of most significant predictions, allowing VGG19 to perform well in tasks like image classification or object detection.

3.3.4. Activation Function (ReLU)

VGG19 uses ReLU (Rectified Linear Unit) as an activation function in each layer of the convolutional and fully connected layers excluding output. The ReLU introduces non-linearity in the network which is crucial while learning complex patterns from data. It is defined as $f(x) = \max(0, x)$, which simply sets any positive values to directly be that 'x', or zero otherwise. This complex yet intuitive concept prevents the vanishing gradient problem, which will not be discussed here; in short it is just that we would like to avoid cases where gradients become too small values or infinitesimally close such that they have no effect when updating weights happens during back propagation.

The use of ReLU in VGG19 speeds up the convergence during training as it helps learn faster. Unlike sigmoid or tanh, ReLU does not saturate which in other words mean that the gradients can flow through neurons even when using large inputs. This makes the training more efficient and less susceptible to getting stuck. ReLU is now one of the most used bits and pieces in this jigsaw, its

simplicity and effectiveness can be understood through VGG19 where it has tremendous success when applied to image recognition/classification etc. RELU, by injecting the layer with non-linearity helps in functionality making it easier for modelling to learn and represent complex features of data.

3.3.5. Dropout Layer

VGG19 uses a dropout to prevent overfitting, which means the network is good in training data but bad on new unseen test data. VGG19 uses dropout between the fully connected layers during training. Randomly "drops out" a percentage of neurons in layer to allow the network to converge on more robust features and not rely too heavily on specific neurons. This makes the model generalize more well on new data as it cannot rely too much on a single neuron or group of neurons during training.

Dropout, as with VGG19 here, has increased in importance due to its ability to enhance generalization performances (especially for large datasets and difficult patterns). The dropout rate (usually between 0.2 and 0.5) specifies which neurons to turn off as a fraction of the total number of that layer's input neuron count. Dropout is used in training mode because it forces the network to learn from a broader set of features mitigating overfitting. However, during inference (when the model is being used for predictions) all neurons are active and so it can make full use of its capacity. This ensures that activations takes place much faster and is also more robust, which ultimately makes VGG19 a better performer especially in large image classification problems.

3.3.6. Output Layer

In the VGG19 Model, this is called the output layer; it performs logic to assess predictions (features) extracted by previous layers. 1000 is that corresponding to 1000 classes (i.e., ImageNet training), because VGG19 was trained on the ImageNet dataset. The output layer consists of a neuron per class, and the network predicts into which class an input image belongs based on the highest activation. The output layer uses a softmax activation function that obtains the probability distribution over all classes from the raw output. In other words, it helps the model understand the possible likelihood of each class and selected the most probable one.

In the case of predictions, the input image, after passing through input, convolutional, maxpooling, and fully connected, goes to the output layer, where the final prediction is generated. This is called the softmax function and it implies that sum of all probabilities equals to 1, which relaxes calculation a bit easier in terms interpreting model prediction. The class having highest probability is then selected which becomes the predicted label of input image. The correct inference of these meaningful predictions and the high-level features that are learned in a hierarchical manner, at each layer, provide a unique capability — Class-specific information is distilled into taking decisions from their local receptive fields to global correlations.

3.3.7. Functionality

The main application to use VGG19 is in feature extraction from input images. Its deep network architecture enable hierarchical representations of visual data, moving from differential gradients and textures in the early layers (closer to input) towards more abstract shapes or objects as they proceed deeper. This property makes it a versatile classifier for image processing problems. One of its popular use cases is a pre-trained model for transfer learning. The model's weight values are initialized using a pre-trained model (e.g., ImageNet) and the network is fine-tuned for small datasets, allowing easy learning with less data. They benefit from the good generalisation of the model, so it works well in contexts with limited training data.

One of the other main work for which VGG19 is known named as Image Classification. In this process, for the input images it is trained and for these it will predict to which class label they belongs using features learned from training. The last layer of the network is a softmax and hence outputs probabilities to put one particular class as label having highest probability value. Despite the fact that VGG19 is not built for object detection, its property to learn general features renders it compatible with object-detection architectures. For example, the feature maps produced by VGG19 may serve as input to

object detection models like Region Convolutional Neural Networks (R-CNNs) or Single Shot Detectors (SSDs), thereby enabling easy identification of objects.

The VGG19 model is extremely useful, not only in classification and object detection but also all types of semantic segmentation tasks. Satellite Image recognition on the other hand represents an even more difficult task for which standard image classification methods are not sufficient given that, in addition to classifying and recognizing images you have to do it “filling” with recognition each pixel of the input satellite photo. The feature maps from VGG19 can be used with decoder networks to get segmentation masks that give where different objects in an image. The network produces feature maps which qualify as the input for constructing these masks, indicating VGG19's versatility in processing pixel level tasks. Deep feature extraction from VGG-16, one of the most famous models used in computer vision projects for semantic segmentation.

In the end, by accommodating all of these basic details, VGG19 is one among very strong and powerful convolutional neural network (CNN) structure that may be widely practiced for photo classification to object detection/recognition along with semantic segmentation. With this ability to generate hierarchical feature representations, it is useful for object detection or similar task of computer vision; because an effective detector should output both the regions that contain objects and their categories in a region. VGG19 is among the most widely used architectures in deep learning, popular amongst researchers and practitioners because it has proven to be very effective at solving a lot of different image challenges. Therefore, it has become one of the vital tools for various professionals working on visual data processing.

3.4. Proposed Unet++/VGG19 Model

The UNet++/VGG19 model is a hybrid architecture that fuses the advantages of both models, aiming to improve satellite image segmentation. This was done by marrying the stacked skip connection of UNet++ with the very powerful feature extraction ability of VGG19 to achieve a desired balance between clear segmentation and detailed features. An improved version of the UNet architecture for biological image segmentation tasks; referred to as an evolution on UNet, naming it UNet++. It has nested, tightly coupled skip connections to enable it to address the shortcomings of original UNet model by enhancing feature propagation and alleviating semantic gap between encoder layers and decoder side. VGG19, on the other hand is state-of-the-art convolutional neural network (CNN) which pretrained with high-resolution natural images and their ability to learn very complex hierarchical features from visual data makes them more specialized for pattern recognition in satellite image.

The UNet++/VGG19 model uses VGG19's pre-trained weights typically trained on a large dataset like ImageNet which results in faster convergence and more generalisation when performing satellite image segmentation. However, pre-training VGG19 captures important low-level features like edges or textures (or shapes), that are essential to differentiate between different land cover classes in the satellite images. This is particularly valuable in the task of segmenting satellite images, where objects can come with many unique sizes and shapes, and be placed into complex and noisy surroundings. The pre-trained weights allow the model to begin with some started learning features, speeding up training and boosting performance in satellite image segmentation.

By grabbing on VGG19 this multi-scale information, can help UNet++ to draw better sharp boundaries for satellite images. Another reason lies in VGG19's wider convolutional layers that include small receptive fields which help capture certain subtle details (microscopic patterns) harder to notice for some deeper, additional architecture. This mechanism of extracting better features and the use of nested skip connections in UNet++, allows the model to carry high-resolution, higher level semantics through all stages: this helps getting more accurate segmentation results. Therefore, the high performance of UNet++/VGG19 in capturing both local and global features can make it an ideal choice for applications such as land cover classification, urban planning or disaster response where reliable delimitation of different types of terrains is needed.

In general, the application of UNet++ to VGG19 contributes important findings in satellite image segmentation. Thus, the model is great for high-accuracy and precision-demanding tasks which benefits from optimal feature propagation as in UNet++ along with robust feature extraction through VGG19.

Due to its efficiency in carrying out computational and analytical operations, as well capturing contextual information at multiple scales for multi-sensor image fusion of complex visual data; geographic community find it the best suited technology especially for environmental monitoring, urban mapping applications disaster management etc. Combined with superior architectural foundations, the UNet++VGG19 model makes a formidable tool to help address limitations around satellite image segmentation in diverse and dynamic surroundings.

4. Experimental Results

In this section, the complete results of performing simulation over suggested methodology is presented. The Kaggle is the source of this investigation dataset details.



(a)



(i)



(b)



(ii)



(c)



(iii)



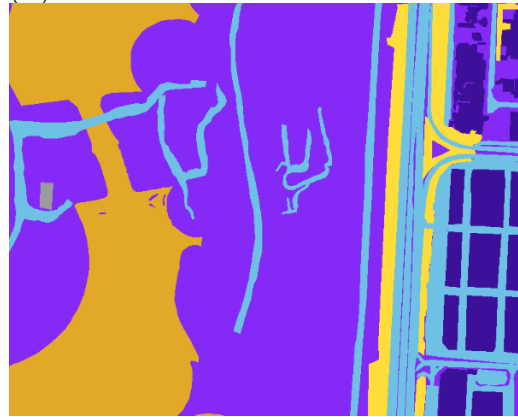
(d)



(iv)



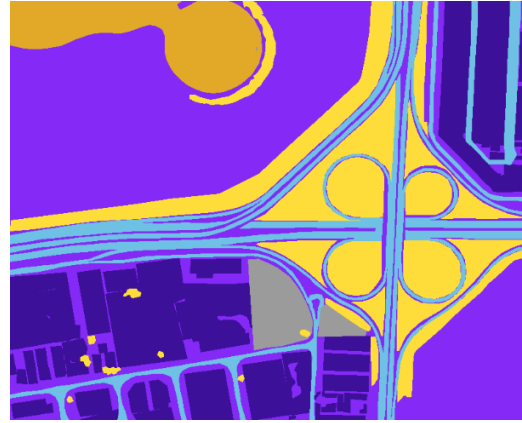
(e)



(v)



(f)



(vi)

Figure 2.
Sample images from dataset.

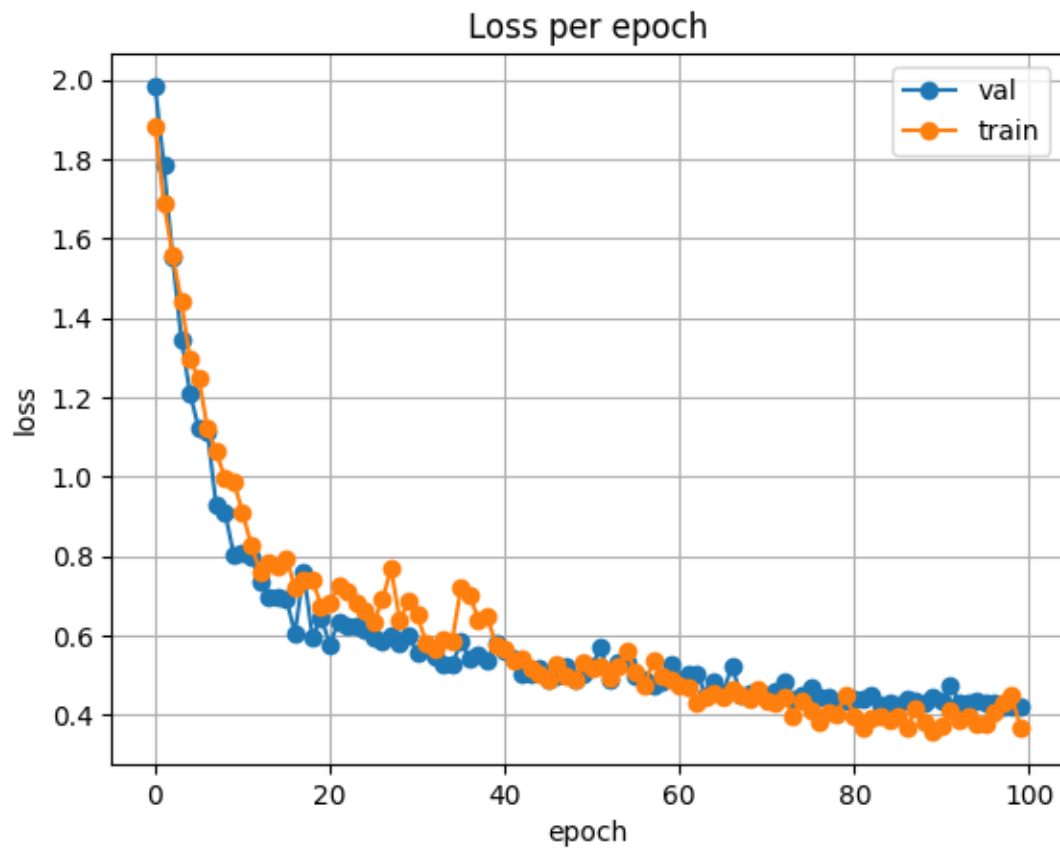


Figure 3.
Train loss Vs validation loss.

Figure 3 also shows the training and validation loss over 100 epochs, indicating how much better or worse our model is performing across different epochs.

- **Training Loss:** When the model is being trained, it tries to reduce the error in on training data set. In the fig above, you can see that training loss is decreasing rapidly in first 20 epochs and then slowly this indicates algorithm is learning patters from data and improving itself on its

prediction. Training loss going down means the model is fitting the training data, but that does not imply it's doing good. But if training loss is decreasing but validation set, it will be overfitting in which case. In this case, make sure to monitor both the losses so that the model learns in an appropriate balanced fashion.

- **Validation Loss:** The validation loss which represents how good our model can generalize over unseen new data. It is initially high, just like the training loss and drops in parallel to lower values of training loss demonstrating you do better. This is quite common as the validation set will be different from training and this would help in testing how well or bad does our model generalize on unseen data. If the validation loss starts diverging from training loss, you may be overfitting instead of generalizing. If, however, the validation loss starts rising but training loss still decreases your model likely needs regularization to avoid overfitting and become more generalizable so that it performs well on unseen data.

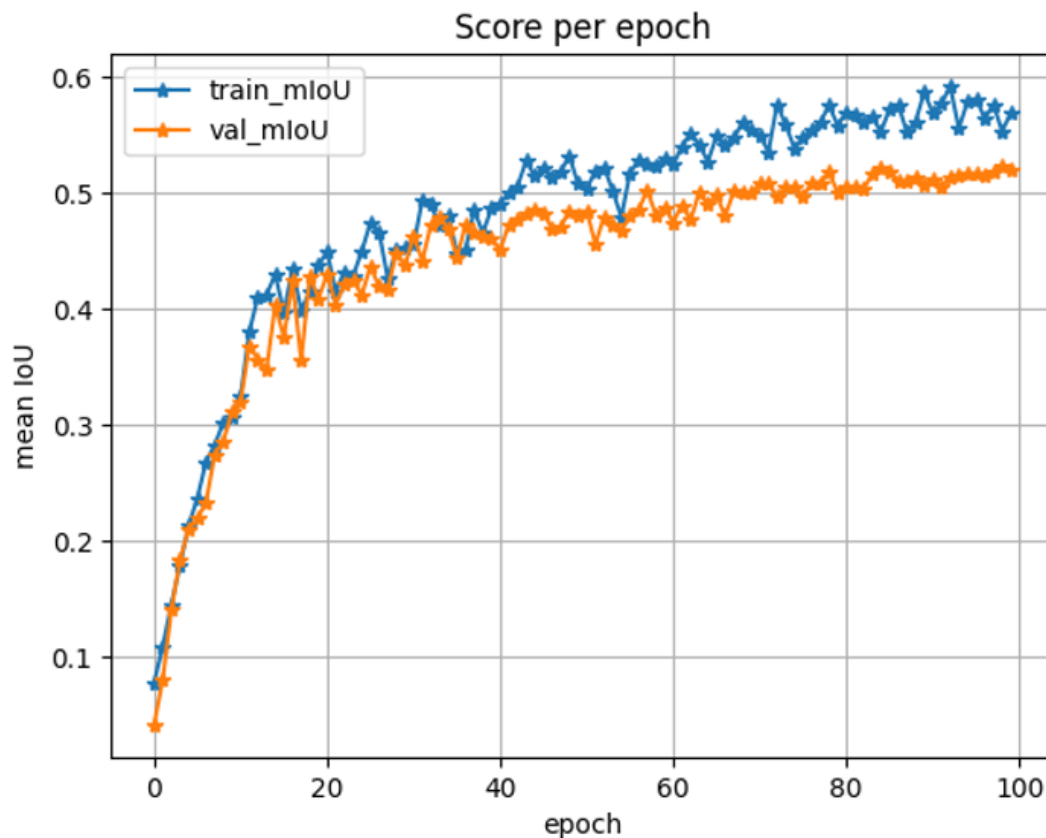


Figure 4.
Train mIoU Vs Validation mIoU.

The mean Intersection over Union (mIoU) metric for the training and validation datasets is shown as a function of 100 epochs in Figure 4. mIoU, the intersection-over-union (iou) is a common metric to evaluate segmentation models by computing overlapping region of predicted mask and ground truth. Both the train and validation mIoU values increases slowly to begin with, however the training one gradually improves faster. After around 20 epochs, they both hover to training mIoU of ~ 0.55 and Validation mIoU about to reach plateau a bit below ~ 0.5 . This is an indication that the model in fact learning reasonable segmentation of images on both training and validation sets. The near values of train and validation mIoU indicate no major overfitting — the model is generalize well. Although, small

variations on validation mIoU can also be attributed to the randomness involved when dealing with unseen data.

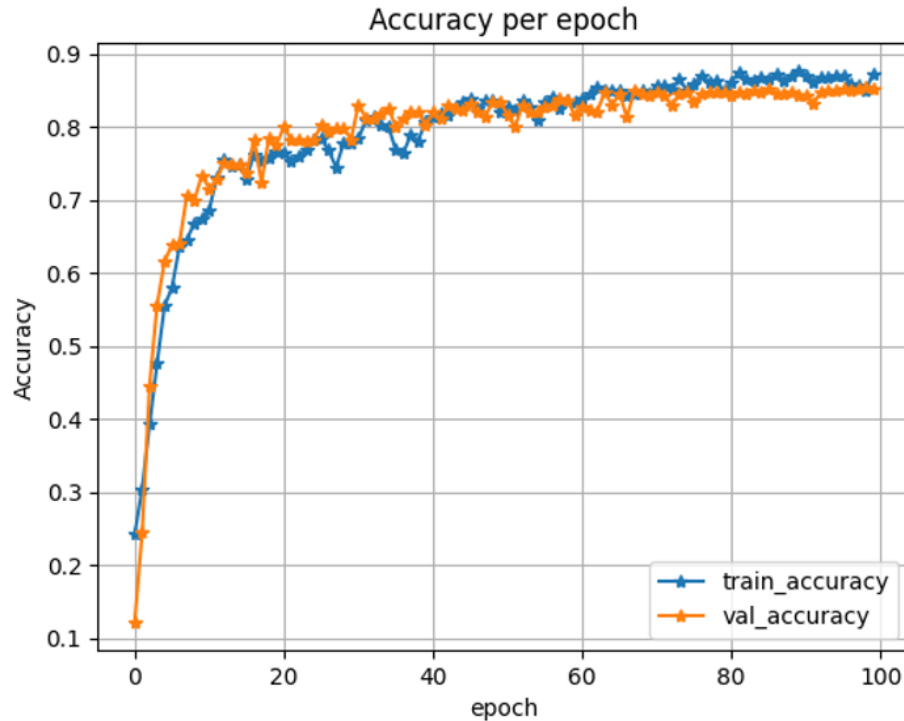


Figure 5.
Train accuracy Vs validation accuracy.

Figure 5 shows the accuracy of the model on training and validation datasets over 100 epochs.

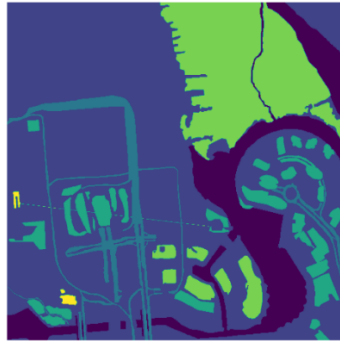
- **Training Accuracy** is the model accuracy on training dataset while each epoch. It specifies how well the model is learning from exposure to data. Notice the training accuracy grows rapidly in the first few epochs, since it depicts how well a model learned its parameters (weights) to minimize loss. Accuracy plateaus at around 20 epochs, indicating the model is close to optimal for correctly classifying training data. As long as the training accuracy keeps increasing, it means that the model is learning better to capturing more patterns in a data.
- **Validation Accuracy** used to compare the generalization capability of a model when it encounters some unseen data, which doesn't contain in training set. Training accuracy and loss behaviour can be seen in the above bar plot, with validation accuracy following a slightly smaller upward trend to again settle around the same point as training. Validation accuracy is close to training accuracy confirms that the model has not been overfitted and it will generalize well with out of sample data from unseen dataset. Nevertheless, oscillations around steady validation accuracy are expected (especially in early epochs) because the model is trained on one set of data and periodically validated against another.

Input image

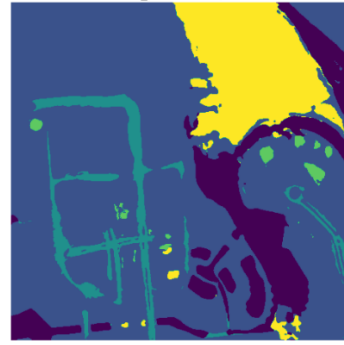


(a)

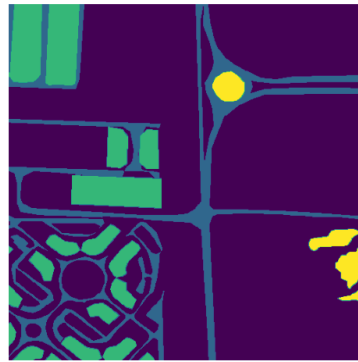
Ground truth image



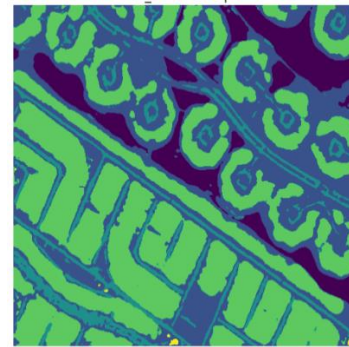
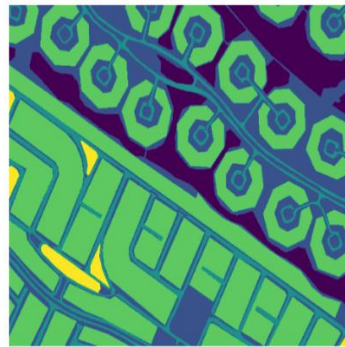
Result image



(b)



(c)



(d)

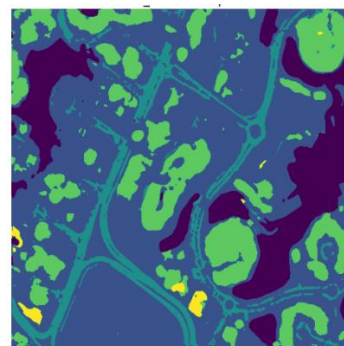
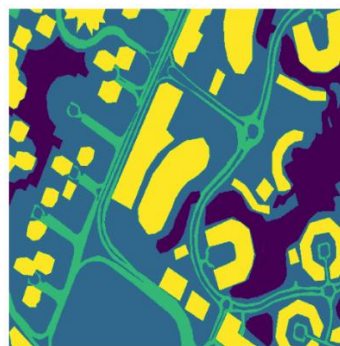


Figure 6.
Output segmented images.

Table 1.
Evaluation metrics.

	Precision	Recall	F1 Score
Image 1	0.835	0.830	0.832
Image 2	0.902	0.902	0.902
Image 3	0.776	0.787	0.782
Image 4	0.747	0.752	0.749

Table 1 gives the precision, recall, and F1 score for four images as well to have an all in one understanding of how good was a model on these metrics. Precision is the number of actual positives from all predict positive instances, whereas recall is how many true positives out there. The F1 metric is the harmonic mean of precision and recall giving a performance balance between various evaluation parameters. Precision, recall and F1 score for Image 1 are close to each other which indicate the model has detected relevant instances consistently. Image 2 displays best scores in each metric, meaning that the model predicts this image accurately. Images 3 and 4 have slightly lower values, in particular image 3 with a recall higher than its precision indicating the model may be better at finding true positives but is less successful avoiding false positives.

Table 2.
Comparative analysis.

Method	Accuracy
Unet++ with efficientnet-b0	0.83
Unet++ with mobilenet_s0	0.79
Unet++ with mobilenet_v2	0.80
Unet++ with VGG19	0.84

Accuracy of Unet++ Architectures coupled with different backbone models are summarized in Table 2. Result shows that the combination Unet++ and VGG19 gives an accuracy of 0.84 here which is better than other comparison made above Unet++ with Efficientnet-b0 came in next at an accuracy of 83% suggesting similarly effective performance across the dataset. On the other hand, Unet++ combined with Mobilenet_v2 yielded a slightly lower accuracy of 0.80 and finally is Mobilenet_s0 showing even less accuracy i.e., 0.79. This comparison examine that, the segmentation performance for VGG19 is best among other tested backbone network and choice of backbone network impacts segmentation performance.

5. Conclusion

The experimental results demonstrated the effectively of the proposed UNet++ with VGG19 for segmenting satellite images. When applied to a large Kaggle dataset with varying images, the provided model exhibited high Precision, Recall, and F1 metrics consistently across different images. This corroborates that the designed system is indeed durable enough to differentiate and partition land cover patterns. The model has specifically given very good results when compared to other configurations such as Unet++ with Efficientnet-b0, Mobilenet_s0 and Mobilenet_v2 having highest accuracy of 0.84. The increased accuracy might be notch due to the higher design of VGG19 which had a very deep and complicated architecture hence an accurate model may have been build in comparison with other sequences investigated.

Copyright:

© 2024 by the authors. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

References

- [1] Sertel, Elif, BurakEkim, PariaEttehadOsgouei, and M. ErdemKabadayi. "Land use and land cover mapping using deep learning based segmentation approaches and vhr worldview-3 images." *Remote Sensing* 14, no. 18 (2022): 4558.
- [2] Kotaridis, Ioannis, and Maria Lazaridou. "Remote sensing image segmentation advances: A meta-analysis." *ISPRS Journal of Photogrammetry and Remote Sensing* 173 (2021): 309-322.
- [3] Ayala, Christian, Rubén Sesma, Carlos Aranda, and Mikel Galar. "A deep learning approach to an enhanced building footprint and road detection in high-resolution satellite imagery." *Remote Sensing* 13, no. 16 (2021): 3135.
- [4] Sheykhmousa, Mohammadreza, Norman Kerle, Monika Kuffer, and SamanGhaffarian. "Post-disaster recovery assessment with machine learning-derived land cover and land use information." *Remote sensing* 11, no. 10 (2019): 1174.
- [5] Wagner, Fabien H., Ricardo Dalagnol, YuliyaTarabalka, Tassiana YF Segantine, RogérioThomé, and Mayumi CM Hirye. "U-net-id, an instance segmentation model for building extraction from satellite images—case study in the joanópolis city, brazil." *Remote Sensing* 12, no. 10 (2020): 1544.
- [6] Li, Zhiwei, Huanfeng Shen, QihaoWeng, Yuzhuo Zhang, Peng Dou, and Liangpei Zhang. "Cloud and cloud shadow detection for optical satellite imagery: Features, algorithms, validation, and prospects." *ISPRS Journal of Photogrammetry and Remote Sensing* 188 (2022): 89-108.
- [7] Rahaman, Jarjish, and Mihir Sing. "An efficient multilevel thresholding based satellite image segmentation approach using a new adaptive cuckoo search algorithm." *Expert Systems with Applications* 174 (2021): 114633.
- [8] Sisodiya, Neha, NitantDube, and Priyank Thakkar. "Next-generation artificial intelligence techniques for satellite data processing." *Artificial Intelligence Techniques for Satellite Image Analysis* (2020): 235-254.
- [9] Yuan, Kunhao, Xu Zhuang, Gerald Schaefer, Jianxin Feng, Lin Guan, and Hui Fang. "Deep-learning-based multispectral satellite image segmentation for water body detection." *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 14 (2021): 7422-7434.
- [10] Pan, Zhuokun, Jiashu Xu, YubinGuo, Yueming Hu, and Guangxing Wang. "Deep learning segmentation and classification for urban village using a worldview satellite image based on U-Net." *Remote Sensing* 12, no. 10 (2020): 1574.
- [11] Raghavan, Ramesh, Dinesh ChanderVerma, Digvijay Pandey, RohitAnand, Binay Kumar Pandey, and Harinder Singh. "Optimized building extraction from high-resolution satellite imagery using deep learning." *Multimedia Tools and Applications* 81, no. 29 (2022): 42309-42323.
- [12] Nalepa, Jakub, Michal Myller, Marcin Cwiek, Lukasz Zak, Tomasz Lakota, Lukasz Tulczyjew, and Michal Kawulok. "Towards on-board hyperspectral satellite image segmentation: Understanding robustness of deep learning through simulating acquisition conditions." *Remote sensing* 13, no. 8 (2021): 1532.
- [13] Nguyen, Thanh Tam, ThanhDat Hoang, Minh Tam Pham, Tuyet Trinh Vu, Thanh Hung Nguyen, Quyet-Thang Huynh, and Jun Jo. "Monitoring agriculture areas with satellite images and deep learning." *Applied Soft Computing* 95 (2020): 106565.
- [14] Khan, Sultan Daud, LouaiAlarabi, and Saleh Basalamah. "Deep hybrid network for land cover semantic segmentation in high-spatial resolution satellite images." *Information* 12, no. 6 (2021): 230.
- [15] Neupane, Bipul, TeerayutHoranont, and JagannathAryal. "Deep learning-based semantic segmentation of urban features in satellite images: A review and meta-analysis." *Remote Sensing* 13, no. 4 (2021): 808.
- [16] Tahir, Arsalan, Hafiz SulimanMunawar, JunaidAkram, Muhammad Adil, Shehryar Ali, Abbas Z. Kouzani, and MA Parvez Mahmud. "Automatic target detection from satellite imagery using machine learning." *Sensors* 22, no. 3 (2022): 1147.
- [17] Rashkovetsky, Dmitry, Florian Mauracher, Martin Langer, and Michael Schmitt. "Wildfire detection from multisensor satellite imagery using deep semantic segmentation." *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 14 (2021): 7001-7016.
- [18] Saralioglu, Ekrem, and OguzGungor. "Semantic segmentation of land cover from high resolution multispectral satellite images by spectral-spatial convolutional neural network." *Geocarto International* 37, no. 2 (2022): 657-677.