Information and Inference: A Journal of the IMA (2022) **00**, 1–10
https://doi.org/DOI HERE

Satellite Image Segmentation

Faraz Vossoughian
Corresponding author: fv2146@vt.edu

[]

Considering the growing number of imaging satellites orbiting around the earth, processing the transmitted data has become more essential. The transmitted image data that is mostly available to public, can be used to gain great insight on different areas of the earth. Manual detection and classification of such images can be quite cumbersome and time consuming. Therefore, an automation process that can accurately detect different classes of identifiable sites becomes very . In this project deep learning is used to perform the classification and segmentation.

1. Introduction

Though the goal of the project is to detect different aerial objects within a larger grid, the first stage of the project image classification has been performed to accurately classify different aerial images using deep learning (CNN). This step will serve as the encoder part of encoder-decoder combination that is necessary for an image segmentation algorithm. Different pre-trained networks that have proven themselves and were trained on millions of training data to extract features have been used. For example, VGG16 was obtained from the “tensorflow” module. Thereafter, the loss and accuracy of the classifier was tracked and hyperparameters such as learning rate and epochs were fine tuned to obtain the highest accuracy in classifying images. In the second stage of the project the encoder or ”back bone” is used in an U-net architecture that performs semantic image segmentation on sattelite imagery.

2. Dataset

In this project, the training, validation and test data are obtained from two sources. First, UC Merced Land Use Dataset. This collection has 21 classes of aerial objects and each image measures 256x256 pixels. These images were manually extracted from USGS national map. To use these images, it should be noted that each backbone requires the input in a specific format. For example, VGG16 requires 224x224 pixels as its input, so a set of transforms are defined to prepare training, test and validation datasets. Fig. 1 shows an example data point for the tennis and beach classes.The dataset is split into pools of training, test and validation. The second dataset consists of aerial imagery obtained by MBRSC satellites and annotated with pixel-wise semantic segmentation in different classes. The total volume of the dataset is 72 images grouped into 6 larger tiles. Each image has a corresponding mask (label) that signifies the different classes

© The Author(s) 2022.

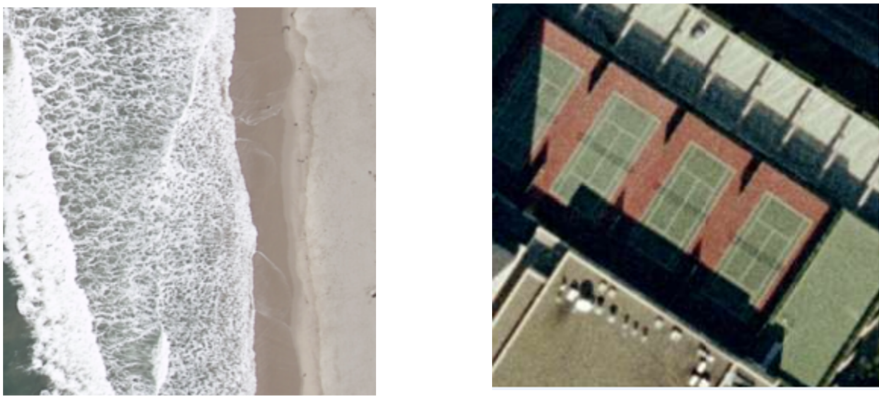

FIG. 1. Data point sample

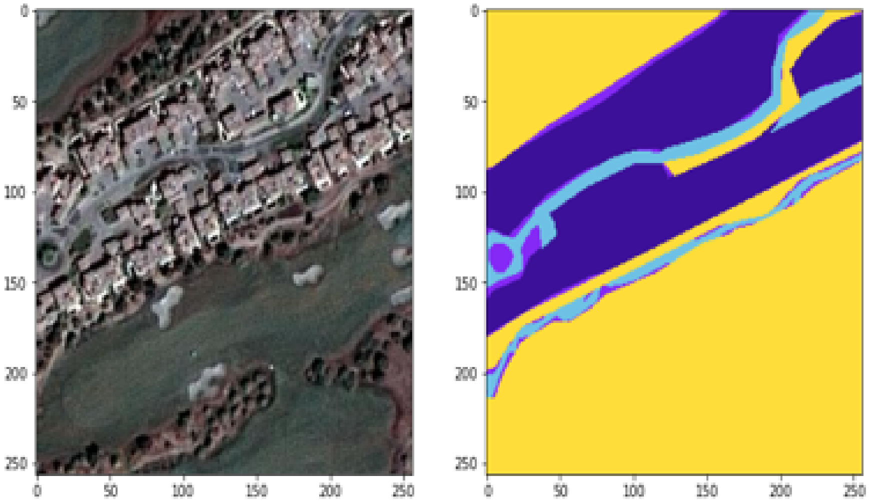

FIG. 2. Satellite image and its corresponding mask (label)

3. Custom Classifier and Feature Extractor

As mentioned before, pre-trained models (transfer learning) were used for extracting features from input classes . For each model, the input needs to be in a certain format. For example for VGG encoder, the input to the model must be 224x224 pixel images. The convolutional layers (filters) are followed by a ReLU activation function to capture positive elements in the feature map that max pooling will be performed upon prior to fully connected layers (where classification occurs). It should be noted that VGG16 was used to extract features from images, the classification of those features were done using a custom classifier that is specifically defined and trained for the scope of this project.

After attaining the features using VGG16 or other pre trained models, a custom classifier is defined. Table. 1 tabulates specification of this classifier. The number of nodes on the hidden layer was selected to be 5000 as it seemed to yield the best predictions.

TABLE 1 *Classifier Specifications*

Number of hidden layer nodes	5000
Activation Function	ReLU
Number of output layer Nodes	21
Output type	Softmax

4. Classifier Training

The training was performed using loss function and Adam optimizer since this optimizer also includes momentum it performs better when looking for a local minimum when adjusting the weights and biases. The optimization was done with a learning rate of 0.001. The training was set up so that it shows validation loss and accuracy at each epoch (15 in total). Table. 2 shows the training loss, validation loss and validation accuracy at every 5 epochs.

TABLE 2 *Progress of validation loss and validation accuracy at every 5 epoch*

Epoch	Validation Loss	Validation Accuracy
1	5.633	0.505
5	0.334	0.823
15	0.511	0.839

It could be seen that as the training progresses, the validation loss decreases and validation accuracy increases.

5. Model Prediction

In order to showcase the capability of the model, the model was tested with images that the model was not trained on. It should be noted that the reason Softmax was used in the output layer is that the exponent of Softmax yields probability of the class being predicted.

Images that have never been seen by the model were used to see how model performs. Fig.3 shows an example image that gets classified by the model.

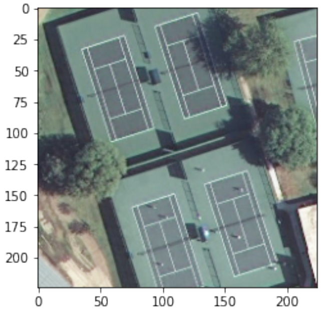

FIG. 3. Input image used for prediction

The top 5 predictions are shown below in Fig. 4. It can be seen that the top 5 classes share similar features as the input image. This is reassuring since it shows that convolution layers of the model work correctly in extracting features.

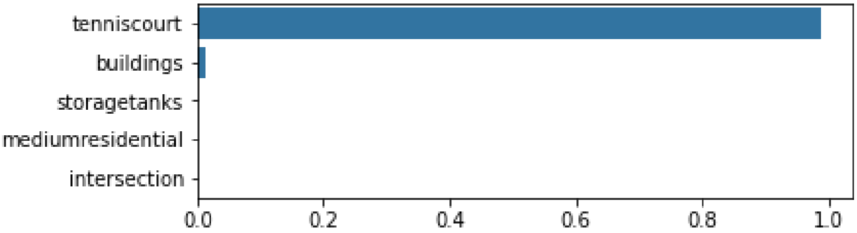

FIG. 4. Top 5 predictions by model

6. U-net model

Now that the approach on the classifier (encoder) is established, the U-net architecture is introduced. The U-net architecture consists of two major parts. First, the encoder or feature extractor applies the convolutions layers followed by max pooling to extract feature representation of input image at multiple different levels. The second part of the U-net architecture is upsampling. In this section of the architecture, the feature vectors get concatenated together to produce an output segmentation map.

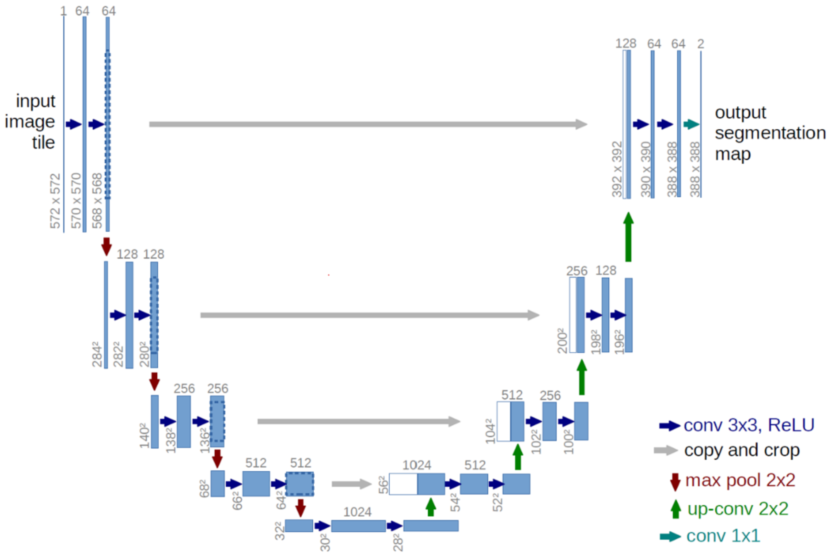

FIG. 5. U-net Architecture

7. U-net model training

For training the U-net model, there's a need to consider a loss function and a metric to monitor how well the model is performing. In this project IOU is used as a metric for how well the segmentation is performed by the model. The IOU represents the area of overlap between predicted and label tile over the entire area. Therefore, one would want to maximize IOU while training the model. Maximizing IOU is equivalent to minimizing -IOU which can be done efficiently using Adam optimizer. Figure below depicts the underlying concept behind IOU. This metric is compared for different backbones and stand alone U-net model to explore whether a pretrained encoder performs better for the purpose of this project.

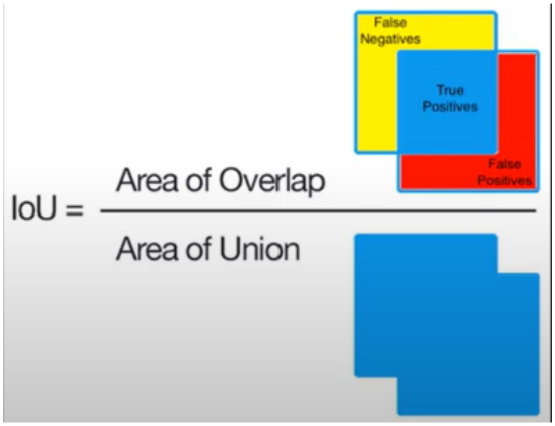

FIG. 6. IOU depiction

8. U-net Predictions

After training the model using training images and labels, a set of predictions are made by the model. In this section some of those results are showcased. The model seemed to have reached a stable state (no further improvements in loss and IOU) after around 100 epochs. The results represent the segmented image compared to the provided label for each satellite image. It can be seen that considering the limited number of training data, the model has done a great job segmenting the test images. Classes that showed up less frequently in the training data are more prone to miss classification, for example, the plantation sites have been miss classified more frequently than the roads or bodies of water that appeared more frequently in the training data set.

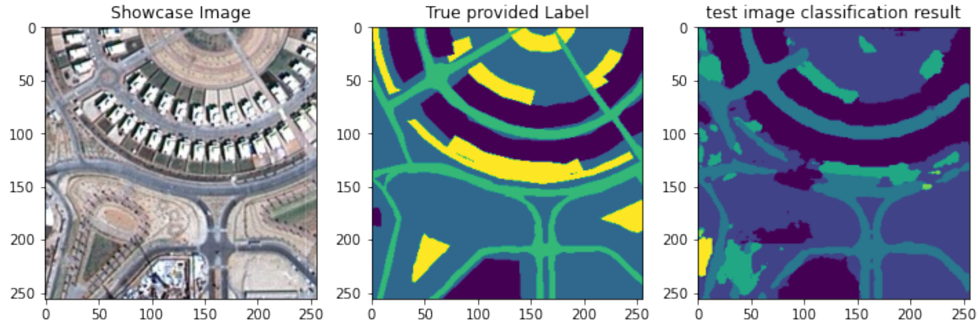

FIG. 7. Prediction Showcase 1

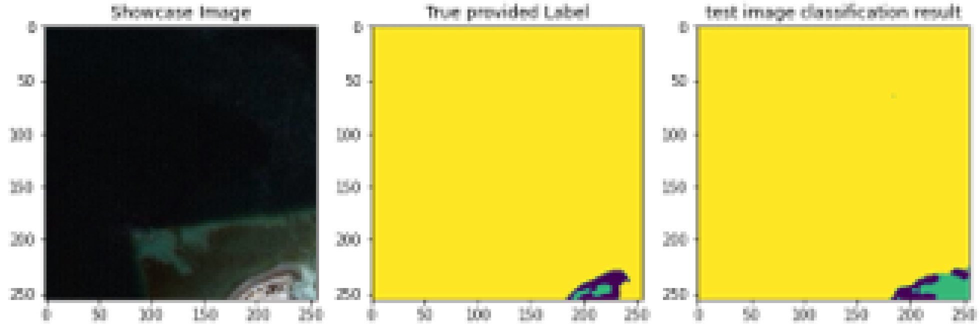

FIG. 8. Prediction Showcase 2

9. Observations

Some key observations have been made in this study. The mean IOU that was earlier defined as an accuracy metric arrives at a steady state after around 100 epochs and its climb stops. Conversely, the loss value continues to decline and at 100 epochs it stabilises. The Figured below show the progression of the model after each epoch. Additionally, different backbones were used as encoders of the U-net model and mean IOU metric was tracked. The Table below shows how they stack up against each other. The standard U-net and U-net with VGG backbone obtained the highest mean IOU compared to other backbones.

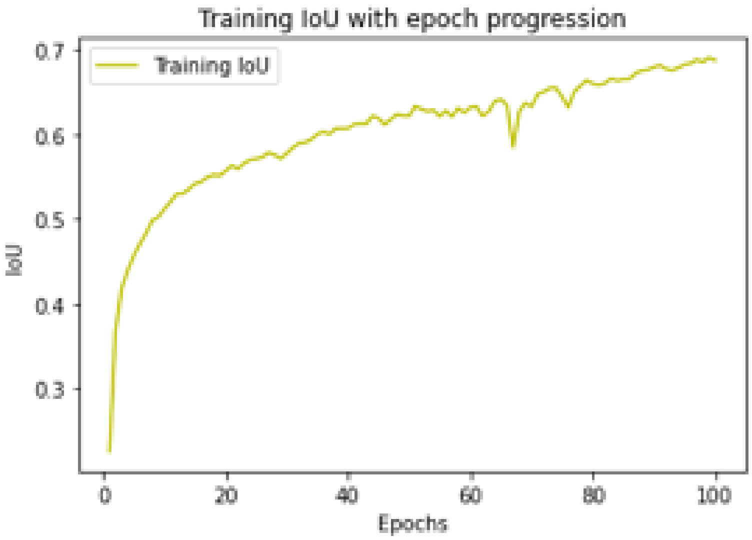

FIG. 9. Training IOU progression

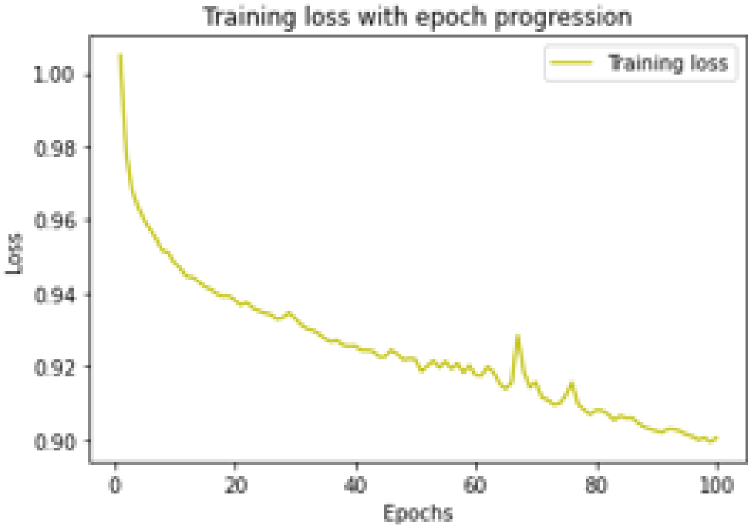

FIG. 10. Training loss progression

TABLE 3 *IOU Comparisons between different encoders*

Encoder	Mean IOU
Resnet34	0.52
Standarad U-net encoder	0.62
VGG16	0.68

10. Summary

To summarise, satellite imagery processing can be quite useful for differnet purposes if proper techniques and training data are used. In this project U-net architecture was used to classify multi class sattelite tiles. After training the model it became evident that the model provides decent segmented maps, however due to lack of the training data, The IOU between predicted and test labels were not very high. This could be due to the fact that each sattelite image (tile) presents a lot of novel features. for example, plantation appeared in only a small subset of the images, and therefore one cant expect the model to classify those rare features with high accuracy. Furthermore, the effect of different pre-trained backbones (encoders) on the accuracy of the model was studied. VGG 16 that has obtained its weights from training on millions of images performed the best in the U-net structure.

REFERENCES

1. Mark Pritt, and Gary Chern, Satellite Image Classification with Deep Learning
2. Shawn D. Newsam. UC Merced Land Use Dataset ageing
3. Dave Mendora Sattelite images along with their masks
4. Eo-JinHwanga, SangheeKimbJoon-YongJung , Fully automated segmentation of lumbar bone marrow in sagittal, high-resolution T1-weighted magnetic resonance images using 2D U-NET