TOPOGRAPHICAL FEATURES EXTRACTION FROM SATELLITE IMAGES

Capstone-I project report submitted by the student of Hybrid UG program in Computer Science & Data Analytics

Sameer Kumar

RollNo.2312_res566

Group No. 103

INDIAN INSTITUTE OF TECHNOLOGY PATNA
BIHTA - 801106, INDIA

Date – 25-06-2024

Declaration

I hereby declare that this submission is my own work and that, to the best of our knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgment has been made in the text.

Student Name - Sameer Kumar

Roll No – 2312_res566

Group No- 103

Signature-

Summary of the Project

This project utilizes deep learning techniques to extract topographical features from satellite images by applying convolutional algorithms of neural networking and advanced image processing, the aim is to identify geographical terrains such as rivers, mountains, and forests, also the system will identify man-made structures like house, bridge, and road. We will also focus on determining the exact dimensions of the topography. This model aims to enhance geographical analysis, providing valuable insights for environmental studies, urban planning, and disaster management.

Throughout the development of the project, we divided our work. Each team member has taken a part and worked accordingly.

- Lokesh (2312RES358)- He has done normalizing images, processing masked images, rendering images on the image dataset, and deployed the deep learning model on hugging face.

- Vicky (2312RES732)- Processing labels, creating RGB2 label function, and splitting the dataset into training and test set.

- Devesh (2312RES247)- He has worked on website development, and has done deep learning network coding, made model matrix, and has trained the deep learning model.

- Aman (2312RES86)- Patchifying and processing of images on an image dataset and has worked to make ui of our deep learning model on gradio ui.

Contents

Chapter 1

Introduction

This project focuses on the extraction of topographical features from satellite images. Utilizing deep learning techniques and advanced image processing, we have developed a model that processes input images, such as satellite images from Google Maps, and outputs images with highlighted topographical features.

Throughout this project, we have applied various image preprocessing techniques, including pacifying, normalizing, rendering, and processing labels, as well as creating training and test sets. We implemented deep learning methodologies such as the Jaccard coefficient and the U-Net model. The model was trained, saved, and used for predictions.

Furthermore, we deployed the trained model on Hugging Face, with the user interface created using Gradio UI for ease of interaction. Comprehensive documentation of the code for image preprocessing, deep network implementation, prediction, and deployment is available at GitHub repository.

Chapter 2

Dataset

The dataset has been taken from the website humansintheloop. The dataset contains aerial imagery of Dubai obtained by MBRSC satellite and annotated with pixel-wise segmentation in 6 classes. It comprises 72 satellite images which are grouped in 6 larger files. The masking has been done in a way that the contents are building-#3C1098, land-#842986, road #6C1E4, vegetation-#FEDD3A, water-#E2A929, unlabeled- #9B9B9B.

The structured and detailed nature of this dataset ensures that it is well-suited for training deep learning models, enabling them to identify and segment various topographical features from satellite imagery. This dataset forms the foundation of our project's objective to enhance feature extraction capabilities through state-of-the-art deep learning techniques. By leveraging this rich dataset, we aim to develop robust models that can be applied to diverse geographical regions, furthering the capabilities of automated topographical analysis and environmental monitoring.

Chapter 3

Jaccard Index

The Jaccard index is a statistic used for checking the similarity and diversity of two sets. The method to find the Jaccard index is finding the intersection over the union that is: J(A,B) = |AnB|/|AuB|.

Where:

- A and B are sets (or lists) of elements.
- |AnB| means the number of elements common to A and B.
- |AuB| means the number of unique elements in A and B combined.

The value of the jaccard index ranges between 0 to 1 where 0 means no similarity and 1 means the two sets are similar. I have used the Jaccard index for the following:

1. Removing duplicate data from our dataset.
2. I have used it as a distance metric to measure the similarity between data points.

Chapter 4

U-Net Model Artitecture

The U -Net model architecture is a convolutional neural network used for the segmentation of images. I have used this architecture for land cover classification, building detection, and other remote sensing applications, also used for general image segmentation tasks like segmenting objects in natural scenes, and image editing. The U-Net model has an encoder-decoder structure, the algorithm is as follows- 1. Input: An input image, typically in the form of a grayscale or RGB image. 2. Encoding: The input image is passed through convolutional layers and max-pooling operations, reducing its size and extracting hierarchical features. 3. Decoding: The encoded features are then decoded using upsampling operations and skip connections to reconstruct the segmentation mask. 4. Output: The output is a binary mask where each pixel indicates the class or category to which it belongs

In the image below the left side is the encoder which takes image as input and breks it into small parts and the decoder which is on the right side do the masking of the input images.

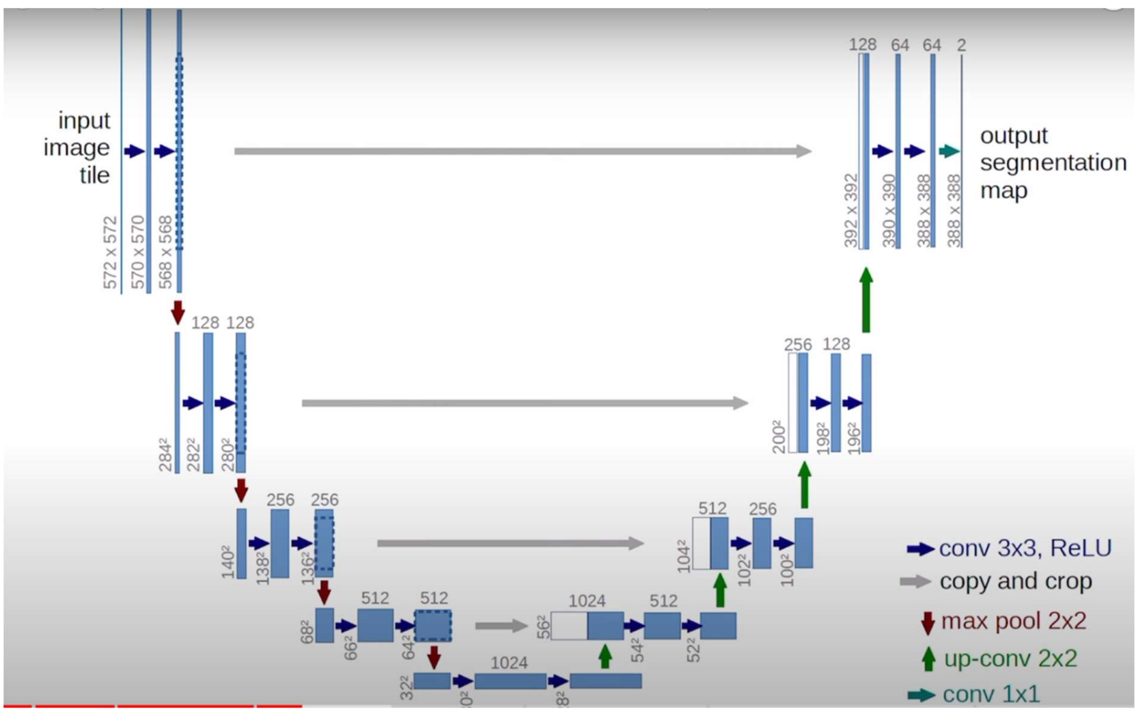

Figure 1 -U-Net model artitecture

(https://lmb.informatik.uni-freiburg.de/people/ronneber/u-net/)

Chapter 5

Deep learning network coding

I have utilized the powerful Python libraries Keras and TensorFlow to implement and compute various loss functions, including Dice Loss, Focal Loss, and Total Loss. These libraries provide robust tools and functionalities for deep learning and model development. Specifically, Total Loss is calculated as a combination of Dice Loss and Focal Loss, formulated as:

Total Loss=Dice Loss+(1×Focal Loss)\text{Total Loss} = \text{Dice Loss} + (1 \times \text{Focal Loss})Total Loss=Dice Loss+(1×Focal Loss)

Focal Loss is particularly effective in providing a more robust training signal, especially beneficial for addressing class imbalances in segmentation tasks. It focuses on hard-to-classify examples, ensuring that the model learns to handle difficult cases more effectively.

To prepare the model for training, model compilation is performed. This involves configuring the model with the Adam optimizer, which is known for its efficiency and adaptability, the specified loss function, and relevant metrics to evaluate model performance during training and testing. These steps ensure that the model is well-optimized and ready for effective training.

The following picture(Figure 2&3) illustrates the code used for calculating Total Loss and compiling the model, highlighting the integration of custom loss functions and the compilation process.

```
dl = sm.losses.DiceLoss(class_weights = w)
fl = sm.losses.CategoricalFocalLoss()
tl = dl + (1 * fl)
```

Figure 2: Code for calculating total loss

```
import tensorflow as tf
tf.keras.backend.clear_session()
models.compile(optimizer="adam", loss=tl, metrics=metrics)
models.summary()
```

Figure 3: Compilation

Chapter 6

Model saving and Prediction of image

Model saving

I have successfully saved the model using the `load_model` function from `keras.models`. This function allows for seamless saving and reloading of complex models. By specifying the appropriate file path, I have ensured that the model is correctly loaded from the designated file. Additionally, I have defined custom objects, including a custom loss function and metric, which are essential for the model's specific requirements. These custom objects are accurately recognized and integrated during the model's operations, ensuring that the model functions as intended with the same level of performance and accuracy as when it was originally trained. This process facilitates efficient model management and reproducibility, crucial for ongoing development and deployment in real-world applications.

Prediction of image

The prediction of image by model is done by the following process-

- The image is resized in length and width of 256, resizing of the image was necessary because we had resized the training dataset in the same format.
- The image is sliced from RGBA to RGB format
- To prepare the image for input in our model I have expanded the channels from shape (height, width, channels) to (1, height, width, channels).

Now the image is ready for prediction and the prediction is shown in Figure 4.

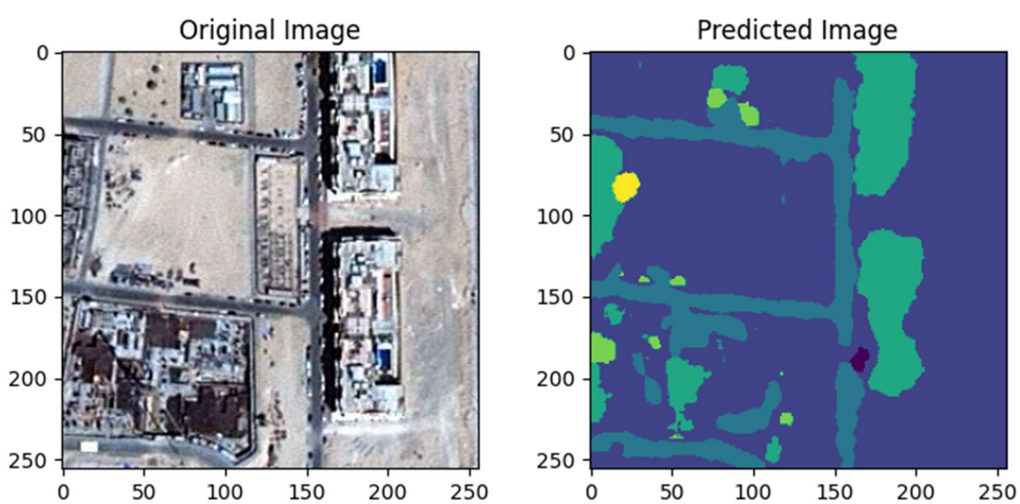

Figure 4 : Prediction

Chapter 7

Conclusion and Future Work

We have developed a highly effective model for accurately masking topographical features. In this project, we have successfully developed a robust model for the extraction of topographical features from satellite images, utilizing advanced deep learning techniques and comprehensive image processing methodologies. The dataset, sourced from Humans in the Loop, provided high-resolution aerial imagery of Dubai, meticulously annotated with pixel-wise segmentation across six distinct classes. This detailed dataset formed the cornerstone of our project, enabling precise training and evaluation of our deep learning model.

Throughout the project, we applied various preprocessing techniques, including patchifying, normalizing, rendering images, and processing labels. We created training and test sets to ensure a rigorous evaluation of our model's performance. The implementation of the U-Net model, combined with custom loss functions such as Dice Loss and Focal Loss, facilitated accurate segmentation and analysis of topographical features.

The comprehensive documentation of our code and methodologies is available in our GitHub repository, ensuring transparency and reproducibility. This project lays a solid foundation for future advancements in satellite image processing, contributing to fields such as urban planning, environmental monitoring, and resource management.

This model demonstrates exceptional performance in identifying and outlining various natural and man-made structures. Moving forward, we aim to enhance this model's capabilities to accurately delineate building footprints and provide precise measurements of various features. This includes determining the area covered by forests, the length of roads, and other relevant dimensions. By achieving this, our model will offer comprehensive insights into the spatial characteristics of different topographical elements, supporting more informed decision-making in fields such as urban planning, environmental monitoring, and resource management.

References

Dataset - Taken from humansintheloop named "Dubai Dataset", source https://www.kaggle.com/datasets/humansintheloop/semantic segmentation-of-aerial-imagery)

Data preprocessing - Learned from 65AI-Labs YouTube channel. Source- https://www.youtube.com/watch?v=UBzMgr6yfpw&t=633s

Machine Learning and Deep Learning- Udemy Source https://www.udemy.com/share/101Wci3@1IgiBN NQpQTTlAdWIxvmsim9JwBuhs8zeris-HB8EXmsUKtPkiR4VXvlTe_ux2VQ==/

Figure1- https://lmb.informatik.uni-freiburg.de/people/ronneber/u-net/

Figure2- https://colab.research.google.com/drive/1FdhPj9AJGgXnfdq7dz0tvHItoJC9w7Lu

Figure3- https://colab.research.google.com/drive/1FdhPj9AJGgXnfdq7dz0tvHItoJC9w7Lu

Figure4- https://colab.research.google.com/drive/1FdhPj9AJGgXnfdq7dz0tvHItoJC9w7Lu