SATELLITE IMAGINARY & AI FOR VEGETATION MONITORING

Team 1 EcoSpectra

AFSHA RATHER - ENG21CT0001

GAURANG GOYAL - ENG21CT0006

KHUSHI SHAH - ENG21CT0015

TANYA SINGH - ENG21CT0044

CONTENT

01 INTRODUCTION
02 LITERATURE SURVEY
03 METHODOLOGY
04 BLOCK DIAGRAM
05 IMPLEMENTATION
06 RESULT

INTRODUCTION

Satellite imaging combined with artificial intelligence has become a powerful approach for monitoring vegetation and assessing environmental health. Through remote sensing, satellite data can capture vast areas of vegetation, enabling regular monitoring over time and detecting even subtle changes in plant cover. This is especially important for tracking ecological shifts due to climate change, urban expansion, and land use practices, which impact biodiversity, ecosystem resilience, and climate stability.

AI algorithms enhance these analyses by processing large volumes of satellite data efficiently, enabling detailed assessments like identifying plant species, detecting vegetation stress, and quantifying biomass. By leveraging machine learning, researchers can automate tasks such as change detection, anomaly identification, and pattern recognition, which are crucial for large-scale environmental monitoring.

LITERATURE REVIEW

S No	Author's name/Paper title	Technology/Design	Results shared by the author	What you infer
1	**Automated Power Lines Vegetation Monitoring Using High-Resolution Satellite Imagery**	High-resolution satellite imagery and machine learning, specifically a semi-supervised approach combining supervised classifiers with deep learning for image segmentation.	84% accuracy was achieved in identifying vegetation risk areas and 92% accuracy in classifying no-risk areas.	A cost-effective and efficient method for vegetation management along power lines using satellite imagery and advanced machine learning, transitioning from periodic inspections to risk-based assessments.
2	**Monitoring and Mapping Vegetation Cover Changes in Arid and Semi-Arid Areas Using Remote Sensing Technology: A Review**	Remote sensing technologies, multispectral sensors (Landsat) and hyperspectral sensors (EO-1 Hyperion), alongside Geographic Information Systems (GIS) and various vegetation indices (NDVI, EVI) to analyse vegetation cover changes.	Trajectory-based change detection method achieved an overall accuracy of 0.95% in vegetation classification.	Remote sensing methods for monitoring vegetation cover and the necessity of high-resolution data and advanced analytical techniques for effective resource management and conservation in vulnerable ecosystems.
3	**The Use of Artificial Intelligence and Satellite Remote Sensing in Land Cover Change Detection: Review and Perspectives**	Artificial Intelligence (AI) techniques, particularly deep learning methods such as CNN, U-Net, and transformer models, alongside satellite remote sensing data for land cover change detection (LCCD).	Achieved high accuracy with supervised learning techniques when sufficient labeled data was available.	Recent advancements in AI and satellite remote sensing have significantly improved land cover change detection methods by addressing data labelling challenges and enhancing accuracy through various machine learning approaches.

S No	Author's name/Paper title	Technology/Design	Results shared by the author	What you infer
4	**Crop Monitoring Using Satellite/UAV Data Fusion and Machine Learning**	PLSR(Partial Least Squares Regression), RFR(Random rest Regression),SVR (Support Vector Regression),and ELR(Extreme Learning Regression) were implemented to estimate AGB, LAI, and N using satellite image-derived VIs and UAV image-based CH and CC as input features.	PLSR 85%, SVR 89%, RFR 91%, ELR 92%	Machine learning techniques to predict important crop characteristics like leaf area, biomass, and nitrogen content by analysing satellite images and UAV images.
5	**Wheat leaf traits monitoring based on machine learning algorithms and high-resolution satellite imagery**	Machine learning algorithms (SVM, ANN, and DNN) for predicting Leaf Area Index - LAI, Leaf Dry Weight, Specific Leaf Area, and Specific Leaf Weight.	Deep Neural Network (DNN) model performed the best in predicting the leaf parameters, with an overall precision of >72%.	sentinel-2 data within a DNN model could provide a comparatively precise and robust prediction of leaf parameters and yield valuable insights into crop management.
6	**Deep learning techniques for estimation of the yield and size of citrus fruits using a UAV**	Yield estimation for individual trees and for a whole commercial orchard using long short-term memory networks	R-CNN model	Transfer learning with fine-tuning techniques is applied to a pre-trained neural network for automatic citrus fruit counting by an UAV.

METHODOLOGY

1. Data Collection: Satellite images from Sentinel, Landsat, or MODIS are used to capture vegetation data.

2. Preprocessing: Images undergo augmentation, noise reduction, and normalization.

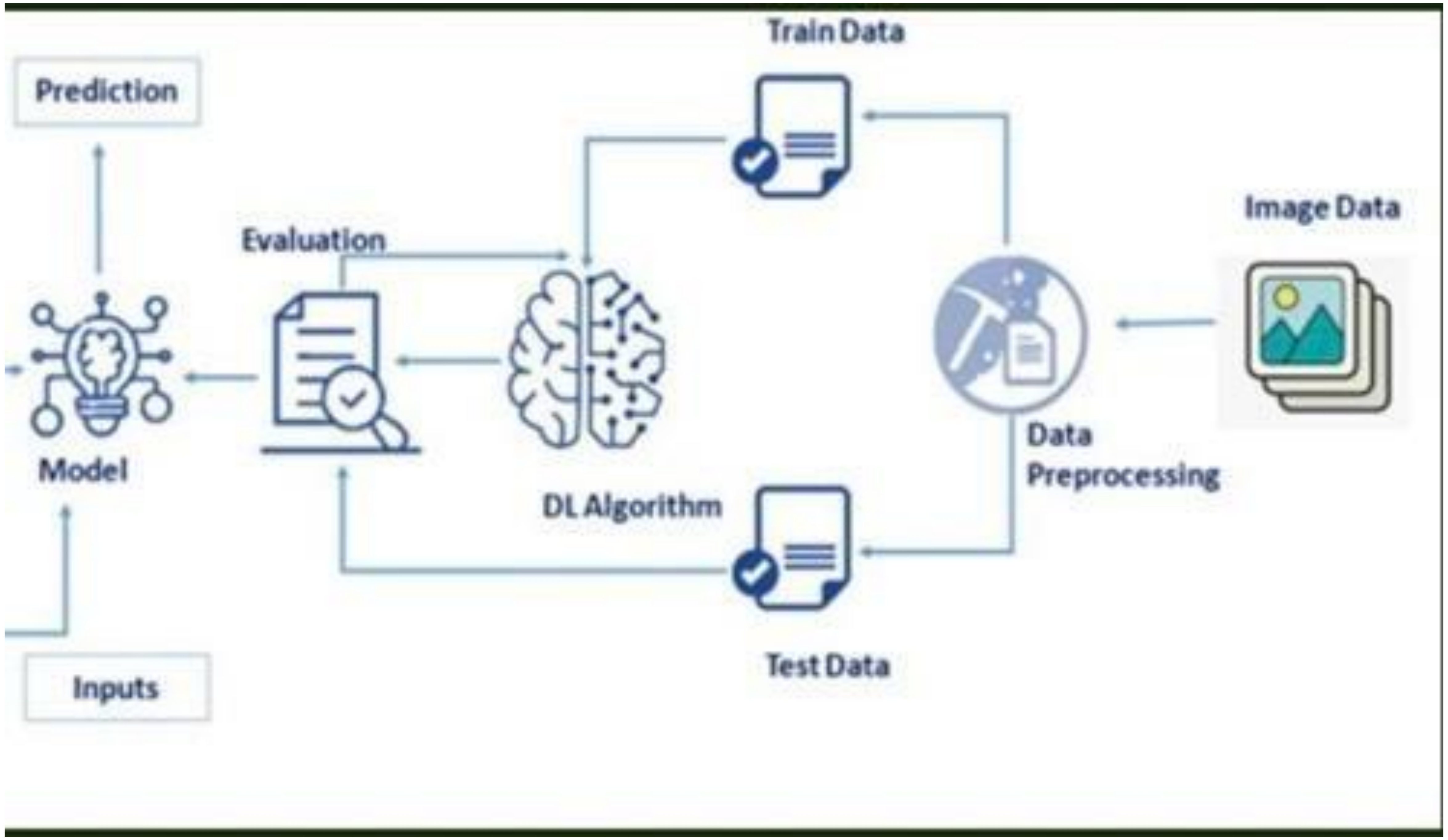
3. Model Design: A CNN is designed for vegetation analysis, with multi-class classification (e.g., healthy crops, deforestation).

4. Training: Models are trained with labeled vegetation datasets and fine-tuned with hyperparameter optimization.

5. Validation & Testing: Accuracy is assessed using metrics like precision, recall, and IoU.

6. Deployment: Real-time satellite data is integrated for continuous monitoring and automated alerts for vegetation changes.

BLOCK DIAGRAM



IMPLEMENTATION

In [2]:

```
import os
import cv2
from PIL import Image
import numpy as np
from patchify import patchify
from sklearn.preprocessing import MinMaxScaler, StandardScaler

from matplotlib import pyplot as plt
import random
```

In [3]:

```
minmaxscaler = MinMaxScaler()
```

In [4]:

```
dataset_root_folder = '/content/drive/MyDrive/Colab Notebooks/datasets/satellite/'
```

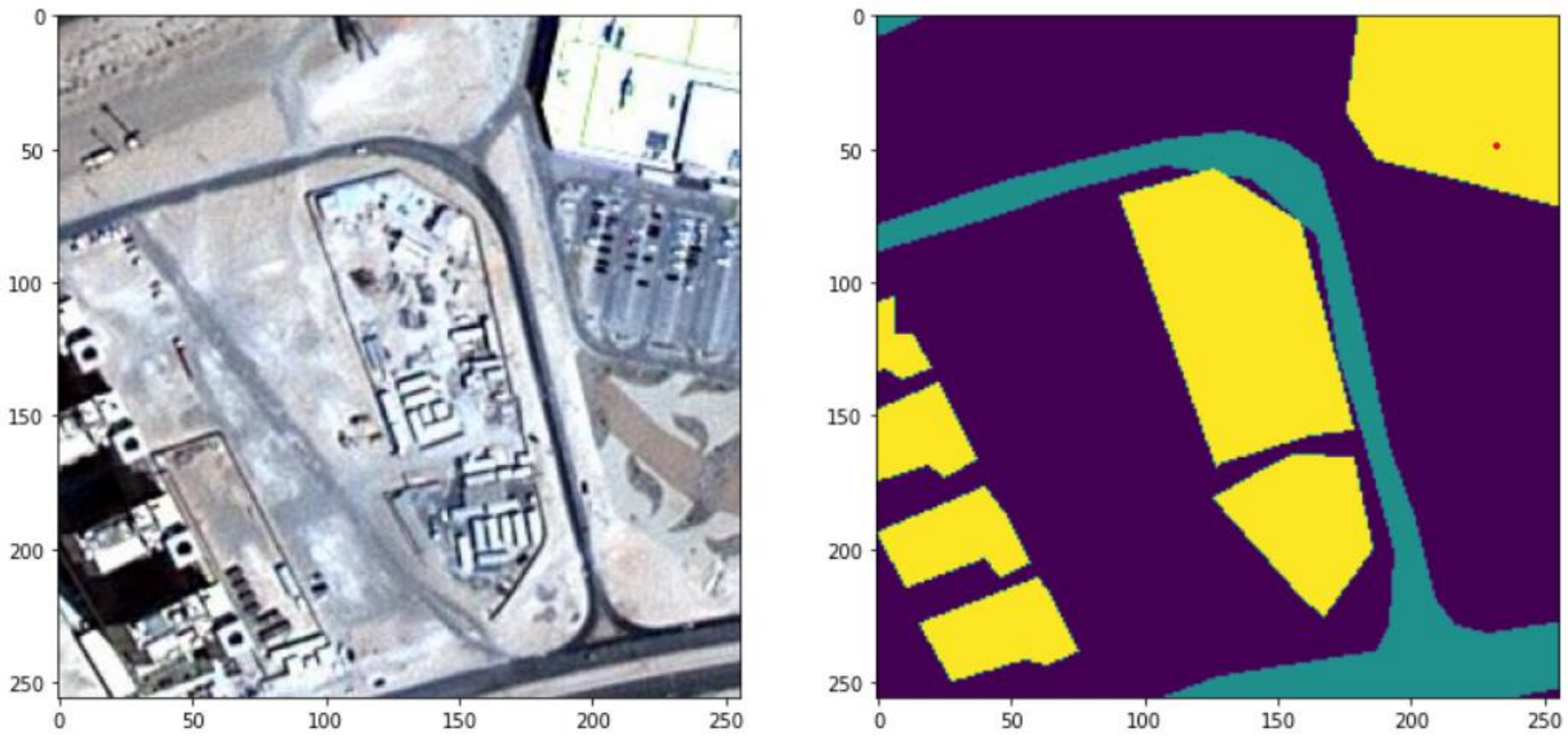
IMPLEMENTATION

In [18]:

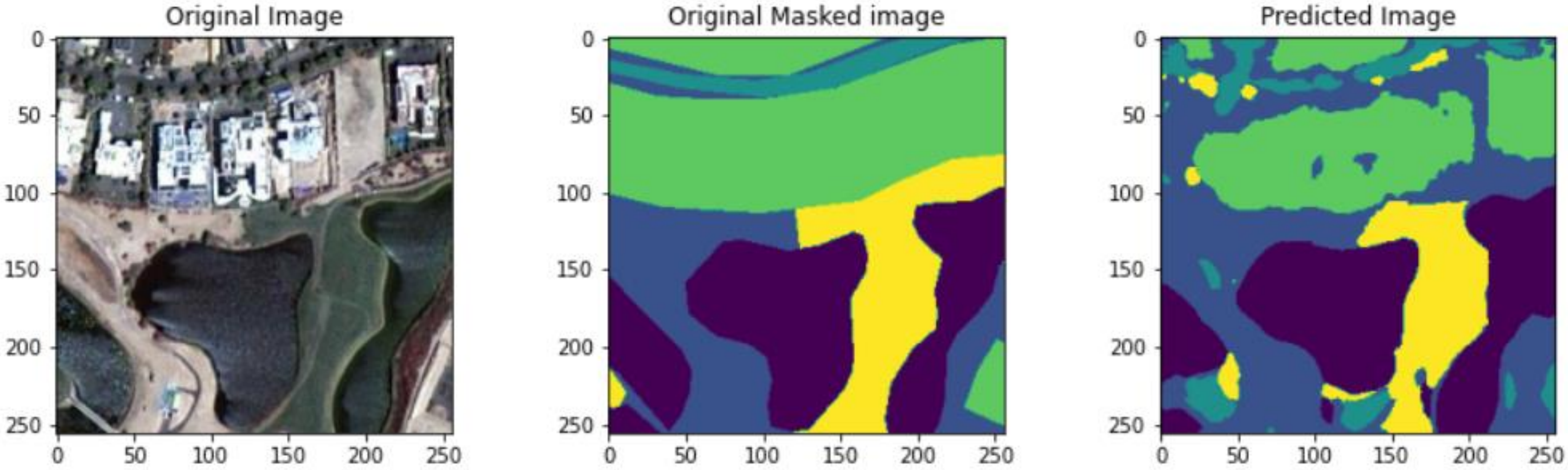
```
random_image_id = random.randint(0, len(image_dataset))

plt.figure(figsize=(14,8))
plt.subplot(121)
plt.imshow(image_dataset[random_image_id])
plt.subplot(122)
#plt.imshow(mask_dataset[random_image_id])
plt.imshow(labels[random_image_id][:,:,0])
```

Out[18]: <matplotlib.image.AxesImage at 0x7fbab9b99750>



RESULTS



THANK YOU