iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 12 **Issue:** VIII **Month of publication:** August 2024

DOI: https://doi.org/10.22214/ijraset.2024.63914

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue VIII Aug 2024- Available at www.ijraset.com

Python-Based Urban Land Suitability Evaluation for Effective Urban Planning

Rohith Kanagaraj
Student in Department of Planning, School of Planning and Architecture, Vijayawada, India

Abstract: ***One of the global environmental problems is that the population is growing rapidly around the world. unfortunately, land is a limited resource. Land suitability analysis merges as the most efficient and commonly used method that represents an effective informative tool to support policymakers and planners in developing optimal master plans, maintaining sustainability, and ensuring social equity among residents. Python's free and open-source ecosystem of geospatial libraries like Geo Pandas makes it a cost-effective alternative to proprietary GIS software for this purpose. Furthermore, Python excels at automation, allowing you to write scripts that save time and ensure consistency for repetitive tasks in land suitability analysis. At first, users can choose the project type based on the classification of land use and input raster and vector data. After that, each layer is reclassified, and a weighted overlay is applied at the end. A raster would be the final land suitability output file. Urban planning benefits greatly from Python's capabilities. Land suitability models can assess the viability of development projects by considering factors like flood risk, slope stability, and proximity to infrastructure. The applications extend far beyond these initial examples. Python's land suitability models are instrumental in environmental management as well. Habitat suitability modeling, for instance, considers factors like vegetation cover, water availability, and connectivity to existing habitats to identify areas suitable for specific wildlife species. This information is crucial for conservation planning and restoration efforts. This paper is designed to explain the beneficial aspects of a Python-based approach to land suitability modelling and how it operates.***

I. INTRODUCTION

Land suitability study can be an important decision-making tool used across disciplines to assess the potential of a range for a specific application. It entails evaluating many ecological, social, and financial factors to determine how well-suited a certain geographical unit is for a given purpose. This research advances feasible land-use practices by ensuring informed decisions about development and asset utilization.

Land suitability models identify regions that are most suitable for growing particular crops by looking at factors such as topography, soil characteristics, and climatic data. This data encourages sustainable farming methods, maximizes yields, and optimizes land use. 2020; Rahman et al. A vital aspect of sustainable urban development is land suitability studies. It assists in determining if development projects are viable by taking into account variables such as slope stability, flood danger, and infrastructure proximity. Urban planners are guided by this data-driven strategy to maximize development projects' feasibility while reducing their negative social and environmental effects. In 2019—Behera et al.

Land suitability analysis, traditionally a manual and time-consuming prepare, is experiencing a change with the rise of automation and AI-based approaches. These strategies are getting to be basic due to the ever-increasing volume and complexity of geospatial information included. Automation streamlines the method by dealing with information handling and analysis assignments, liberating up experts for interpretation and decision-making. Typically especially important for large-scale projects. Moreover, automation and AI minimize human mistake, driving to more reliable and possibly more exact comes about. AI's capacity to analyze complex geospatial information and distinguish inconspicuous patterns permits for a more comprehensive understanding of land suitability components, coming about in more nuanced and vigorous evaluations. At long last, these strategies coordinated consistently with existing GIS program and other instruments, encouraging a streamlined workflow and the consolidation of differing datasets. Whereas conventional strategies stay significant, mechanization and AI offer critical points of interest for land suitability analysis. As these technologies progress, they are balanced to play an indeed more prominent part in guaranteeing productive, exact, and feasible land-use practices.

The Objectives of this research are

1) To develop a Python-based urban land suitability analysis tool to aid effective urban planning
2) To automate the complex process land suitability analysis using basic python language

3) To streamline the traditional method of manual data preparation and analysis, which is time consuming.

II. LITRATURE REVIEW

A. Land Suitability Analysis Methods and Technologies: A Review

1) Methods:

- FAO Land Evaluation Framework: The Food and Agriculture Organization (FAO) uses this framework extensively, and it places a strong emphasis on biophysical elements like terrain, soil properties, and climate. Based on restrictions for a particular land use, it divides land appropriateness into several categories (very suitable, moderately suitable, marginally acceptable, etc.)
- Multi-Criteria Decision Making (MCDM): This method takes into account not just biophysical considerations but also social, economic, and environmental aspects that affect land suitability. For appropriateness assessment, MCDM approaches such as Fuzzy Logic and Analytical Hierarchy Process (AHP) assist in assigning weights and integrating these many factors

2) Technologies:

- Geographic Information Systems (GIS): An investigation of land suitability requires the use of GIS. It enables the integration of spatial data from a variety of sources, including satellite imagery, elevation data, and soil maps. GIS facilitates the identification of regions with favorable factor combinations for a particular land use by overlaying and evaluating these datasets [1, 4].
- Remote sensing: Information regarding vegetation health, soil moisture, and land cover can be obtained through the use of satellites and drones to gather data on remote sensing. Particularly when applied to vast regions, this data can be incorporated into GIS to provide a more thorough analysis of land suitability [3].

B. Land Suitability Analysis with AI and Machine Learning: A Look at Recent Studies

- The usefulness of different machine learning (ML) methods, such as MaxEnt, Random Forest (RF), and Support Vector Machines (SVM), for agricultural land suitability assessment is examined in this work by Rahman et al. (2020). According to their results, MaxEnt—a well-liked option for species distribution modeling—can be successfully modified for land suitability research, especially because of its capacity to manage sparse presence data [1].
- Using Machine Learning for Land Suitability Classification: Mohammadi et al. (2015) investigated the application of ensemble learning techniques, particularly RotBoost, for the classification of land suitability in a different study. Their study shows that RotBoost can achieve great accuracy (around 99%) when compared to other conventional approaches like AdaBoost or Rotation Forest. [2].

III. STRENGTHS AND LIMITATIONS

A. Strengths

The emphasis on transparency and repeatability in this study is one of its main advantages. The entire process of evaluating the suitability of urban land is made transparent and reproducible for other academics and practitioners of urban planning by using Python and its extensive array of geospatial libraries, including GeoPandas, Rasterio, and SciPy. The method is also affordable and available to people with different financial capacities because it makes use of open-source software. Urban planners and academics, even those with no programming experience, may adopt and modify the methodology with remarkable ease thanks to Python's easily comprehensible syntax and copious documentation because the procedure is interactive and takes user input for suitability ranges and weights, it can be adjusted to varied stakeholder preferences and guarantees that the evaluation accurately reflects the priorities and concerns of the community.

This research also establishes a solid platform for the future adoption of machine learning algorithms. While typical machine learning algorithms have been used in other domains, this work offers the possibility of their application in urban planning, paving the door for more complex, data-driven decision-making tools in the future.

B. Limitations:

Despite its strengths, this research also has several limitations. The accuracy and reliability of the evaluation are heavily dependent on the quality and resolution of the input data.

Incomplete or outdated data can lead to incorrect assessments, and the methodology's applicability may vary in different geographic contexts with diverse data availability and urban characteristics because it is subjective by nature and depends on user input, the criterion weighing procedure has additional drawbacks. Especially for large-scale investigations, manually entering suitable ranges and weights can be time-consuming and prone to human error. The nonattendance of machine learning integration implies that the current approach does not benefit from the upgraded exactness and value that prescient models can provide. In any case, this inquiry speaks to a basic preparatory step, highlighting the potential for future improvements that join machine learning procedures to address these impediments and progress the general adequacy of urban appropriateness assessments.

IV. DATA USED

for this exercise we are performing land suitability for future parks in Rajahmahendravaram city using the data collected during surveys and manually digitized layers. we have used existing parks, residential area, roads and vacant areas withing the city boundary. here is the basemap of all the overlayed layers with city boundary

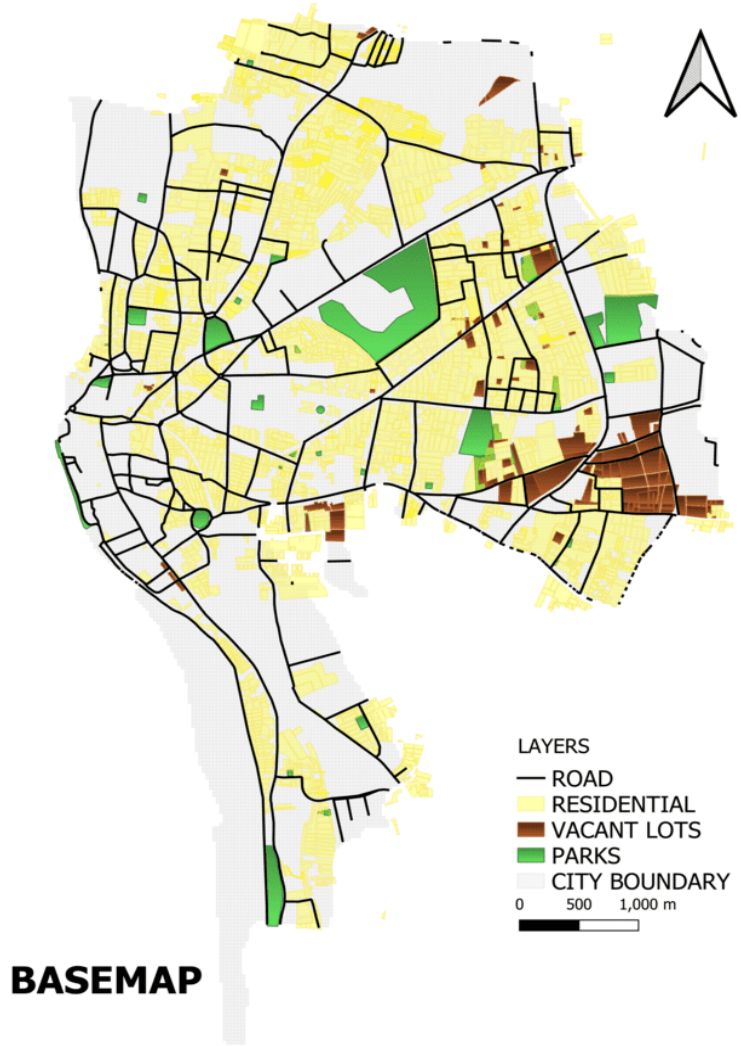


Fig 4.1. Basemap including all layers used

V. METHODOLOGY

In order to evaluate and display the suitability of land for urban development, a Python-based urban land suitability study methodology consists of many crucial processes that make use of geospatial data and computational tools. This method processes geographical data, computes Euclidean distances, and suits reclassifications using Python and its reliable libraries, including GeoPandas, Rasterio, NumPy, Matplotlib, and SciPy. The specific phases in this technique are described in depth in the paragraphs that follow.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue VIII Aug 2024- Available at www.ijraset.com

```
import geopandas as gpd
from rasterio.features import geometry_mask
from rasterio.transform import from_origin
import numpy as np
import matplotlib.pyplot as plt
from scipy.ndimage import distance_transform_edt
```

Fig 5.1. Imported python libraries

1) Euclidean

```
def calculate_euclidean_distance(features_gdf, output_extent_path, output_resolution):
    """
    Calculates the Euclidean distance raster for features (e.g., roads, parks, residential) within the specified output extent.

    Args:
        features_gdf (geopandas.GeoDataFrame): GeoDataFrame containing feature geometries.
        output_extent_path (str): Path to the output extent shapefile.
        output_resolution (float): Resolution of the output raster (in units of the CRS).

    Returns:
        numpy.ndarray: Euclidean distance raster.
    """
    # Read output extent shapefile
    output_extent_gdf = gpd.read_file(output_extent_path)

    # Ensure CRS matches
    if features_gdf.crs != output_extent_gdf.crs:
        features_gdf = features_gdf.to_crs(output_extent_gdf.crs)

    # Get bounds and shape of the output extent
    minx, miny, maxx, maxy = output_extent_gdf.total_bounds
    width = int((maxx - minx) / output_resolution)
    height = int((maxy - miny) / output_resolution)

    # Create transformation for raster
    transform = from_origin(minx, maxy, output_resolution, output_resolution)

    # Create mask of output extent
    extent_mask = geometry_mask(output_extent_gdf.geometry, transform=transform, invert=True, out_shape=(height, width))

    # Initialize a mask with all False values
    features_mask = np.zeros((height, width), dtype=bool)

    # Loop through each feature and update the mask
    for geom in features_gdf.geometry.dropna():  # Drop invalid geometries
        feature_geom_mask = geometry_mask([geom], transform=transform, invert=False, out_shape=(height, width))
        features_mask = features_mask | ~feature_geom_mask
```

Fig 5.2. Euclidean

First, within a given output extent, the core function calculate_euclidean_distance computes the Euclidean distance from a variety of features (such as roads, parks, and residential areas). In order to verify that the coordinate reference systems (CRS) of the feature geometries and the output extent match, this function first reads the output extent shapefile. After that, it establishes the output extent's perimeter and form, generating a transformation for raster generation. Rasterio's geometry_mask is used to create an output extent mask. By iterating over each feature, the function updates a boolean mask that denotes whether the feature is present. The distance raster that results is then clipped to the output extent after the Euclidean distance is computed using SciPy's distance_transform_edt function.

2) Reclassification

```
def reclassify_distance(euclidean_distance, suitability_ranges):
    """
    Reclassifies the Euclidean distance raster to suitability values based on user-defined ranges.

    Args:
        euclidean_distance (numpy.ndarray): Euclidean distance raster.
        suitability_ranges (list of tuples): List of tuples representing suitability ranges and their associated values.

    Returns:
        numpy.ndarray: Reclassified raster with suitability values.
    """
    reclassified = np.full(euclidean_distance.shape, np.nan)

    # Reclassify park areas, roads, and residential areas as 1
    reclassified[(euclidean_distance == 0) & (euclidean_distance != np.nan)] = 1

    for range_, suitability_value in suitability_ranges:
        min_range, max_range = range_
        reclassified[(euclidean_distance > min_range) & (euclidean_distance <= max_range)] = suitability_value

    return reclassified
```

Fig 5.3. Reclassification

The Euclidean distance raster is then reclassified into suitable values using user-defined ranges by the reclassify_distance function. This function sets up an array to hold the reclassified values and gives places that have a Euclidean distance of zero—that is, areas with roads, parks, or residential areas—a suitability rating of 1, signifying that they are not appropriate for additional development. The raster is then updated with relevant suitability values based on the distance ranges as the function iterates over the supplied suitability ranges.

3) Classifying vacant lands

```
def classify_vacant_land(city_extent_path, vacant_land_path, output_resolution):
    """
    Classifies vacant land within the city boundary with a high suitability value (5) and all other areas as 3.

    Args:
        city_extent_path (str): Path to the city boundary shapefile.
        vacant_land_path (str): Path to the vacant land shapefile.
        output_resolution (float): Resolution of the output raster (in units of the CRS).

    Returns:
        numpy.ndarray: Reclassified raster for vacant land.
        Affine: Transform for the raster.
    """
    city_extent_gdf = gpd.read_file(city_extent_path)
    vacant_land_gdf = gpd.read_file(vacant_land_path)

    # Ensure CRS matches
    if vacant_land_gdf.crs != city_extent_gdf.crs:
        vacant_land_gdf = vacant_land_gdf.to_crs(city_extent_gdf.crs)

    # Get bounds and shape of the city extent
    minx, miny, maxx, maxy = city_extent_gdf.total_bounds
    width = int((maxx - minx) / output_resolution)
    height = int((maxy - miny) / output_resolution)

    # Create transformation for raster
    transform = from_origin(minx, maxy, output_resolution, output_resolution)

    # Create mask of city extent
    city_mask = geometry_mask(city_extent_gdf.geometry, transform=transform, invert=True, out_shape=(height, width))

    # Create mask for vacant land
    vacant_land_mask = geometry_mask(vacant_land_gdf.geometry, transform=transform, invert=True, out_shape=(height, width))

    # Initialize the reclassified raster
    reclassified = np.full((height, width), 3, dtype=np.float32)

    # Classify vacant land as 5
    reclassified[vacant_land_mask] = 5

    # Mask out areas outside the city extent
    reclassified[~city_mask] = np.nan

    return reclassified, transform
```

Fig 5.4. classifying vacant land

The classify_vacant_land function is used to find acceptable vacant land inside a city limit. In order to verify that the CRS of the unoccupied land and city border shapefiles match, this function reads them. After that, a raster transformation is created and the boundaries and geometry of the city extension are established. A default appropriateness value of 3 is specified for the initial raster, and masks are constructed for the city extension and undeveloped terrain. regions beyond the city limits are hidden, and vacant land regions are assigned a high appropriateness score of 5.

4) Proccess Features:

```
def process_features(feature_type, feature_files, output_extent_shapefile, output_resolution):
    reclassified_rasters = []
    for feature_file in feature_files:
        # Read feature shapefile
        features_gdf = gpd.read_file(feature_file)
        print(f"Loaded {len(features_gdf)} {feature_type} geometries from {feature_file}.")

        # Ask user for suitability ranges for this specific file
        suitability_ranges = []
        print(f"Enter suitability ranges for {feature_file}:")
        while True:
            min_range_input = input("Enter the minimum distance for a suitability range (or type 'done' to finish): ")
            if min_range_input.lower() == 'done':
                break
            min_range = float(min_range_input)
            max_range = float(input("Enter the maximum distance for this suitability range: "))
            suitability_value = float(input("Enter the suitability value for this range: "))
            suitability_ranges.append(((min_range, max_range), suitability_value))

        # Calculate Euclidean distance raster
        euclidean_distance, transform = calculate_euclidean_distance(features_gdf, output_extent_shapefile, output_resolution)

        if np.isnan(euclidean_distance).all():
            print(f"No valid {feature_type} found within the extent.")
        else:
            # Reclassify the Euclidean distance raster
            reclassified_raster = reclassify_distance(euclidean_distance, suitability_ranges)
            reclassified_rasters.append((reclassified_raster, feature_type))

            # Plot Reclassified raster
            plt.imshow(reclassified_raster, cmap='viridis', origin='upper')
            plt.colorbar(label='Suitability')
            plt.title(f'Reclassified Suitability from Euclidean Distance for {feature_type}')
            plt.xlabel('Column')
            plt.ylabel('Row')
            plt.show()

    return reclassified_rasters, transform
```

Fig 5.5.Process features

Roads, parks, and residential areas are among the several feature types that are processed by the process_features function, which also produces reclassified suitability rasters for each feature type. The function reads the feature shapefiles, calls the calculate_euclidean_distance and reclassify_distance routines, and asks the user to enter suitable ranges. Plotting the reclassified rasters for visualization follows.

5) Main function:

```
def main():
    # Paths to input data
    roads_files = [r'D:\rajahmahendravaram_gis_work\ROADS SKEL.1\ROADS SKEL.1.shp']  # Add more road shapefile paths as needed
    parks_files = [r'D:\rajahmahendravaram_gis_work\recreational_latest\recreational_latest.shp']  # Add more park shapefile paths as needed
    residential_files = [r'C:\Users\Administrator\OneDrive\Desktop\pyland\own site\residential_latest.shp']  # Add more residential shapefile paths as needed
    vacant_land_file = r'C:\Users\Administrator\OneDrive\Desktop\pyland\own site\vacant ind=side.shp'
    output_extent_shapefile = r'C:\Users\Administrator\OneDrive\Desktop\pyland\own site\WARD_analysis\WARD_COMBINED.shp'
    city_extent_shapefile = r'C:\Users\Administrator\OneDrive\Desktop\pyland\own site\WARD_analysis\WARD_COMBINED.shp'

    # Output resolution
    output_resolution = 10  # meters
```

Fig 5.6. Input shapefiles

```
    # Ask user for weights for each raster
    weights = []
    print("Enter the weights for the overlay (total should sum up to 1):")
    for raster, feature_type in all_rasters:
        weight = float(input(f"Enter the weight for {feature_type} layer: "))
        weights.append(weight)

    # Perform weighted overlay
    weighted_overlay = np.zeros_like(all_rasters[0][0], dtype=float)
    for (raster, feature_type), weight in zip(all_rasters, weights):
        weighted_overlay += raster * weight
```

Fig 5.7. weighted overlay

Lastly, the procedure is coordinated by the main function. The pathways to input data, such as parks, streets, homes, open space, and city border shapefiles, are specified. After processing each kind of feature to produce suitable rasters, the function aggregates all of the rasters to produce a weighted overlay analysis. In order to create a final weighted suitability overlay, users are asked to supply weights for each raster layer.

Plotting the resultant suitability map offers a thorough visual representation of the appropriateness of the area for urban design.
In outline, this Python-based technique gives a precise approach to urban land reasonableness examination by joining geospatial information preparation, removing calculations, and user-defined reclassifications. It builds up a straightforward, reproducible, and adaptable system that can be adjusted to different urban arranging scenarios, laying the foundation for future upgrades, counting the consolidation of machine learning models.

VI. RESULTS

A. *Suitability Ranges*

The user is invited to specify appropriateness ranges for each feature type (e.g., roads, parks). For any feature category, the user can enter more than one range. This gives you precise control over how each feature's distance from you influences appropriateness.

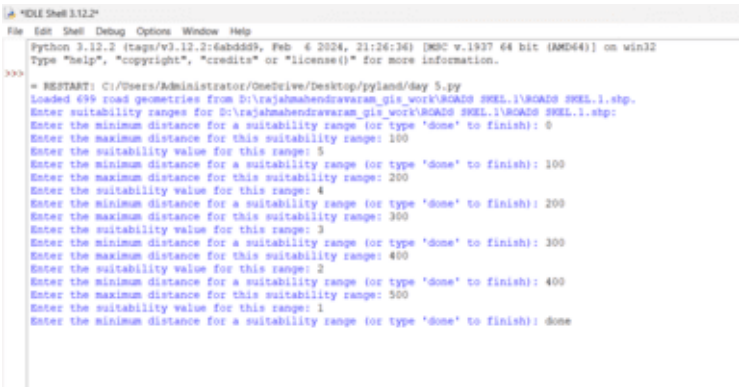


Fig 6.1. reclassification range (user input)

B. *Weights for Weighted Overlay*

Weighted overlaying requires the user to give weights to each reclassified raster layer. These weights should add up to one. This is an important stage because it shows how important each component is in evaluating the overall appropriateness of the property.

Fig 6.2. weights for weighted overlay (user input)

C. Reclassification Results:

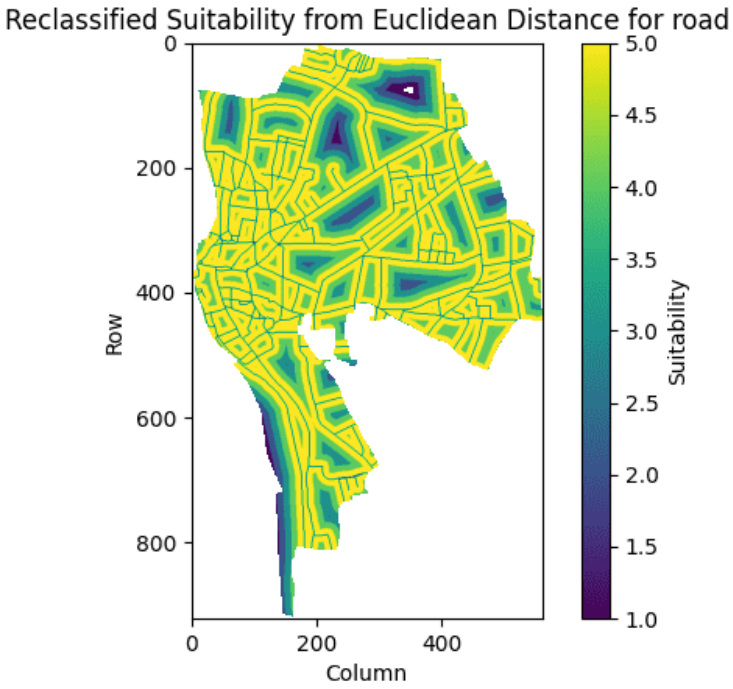


Fig 6.3.Roads reclassification output

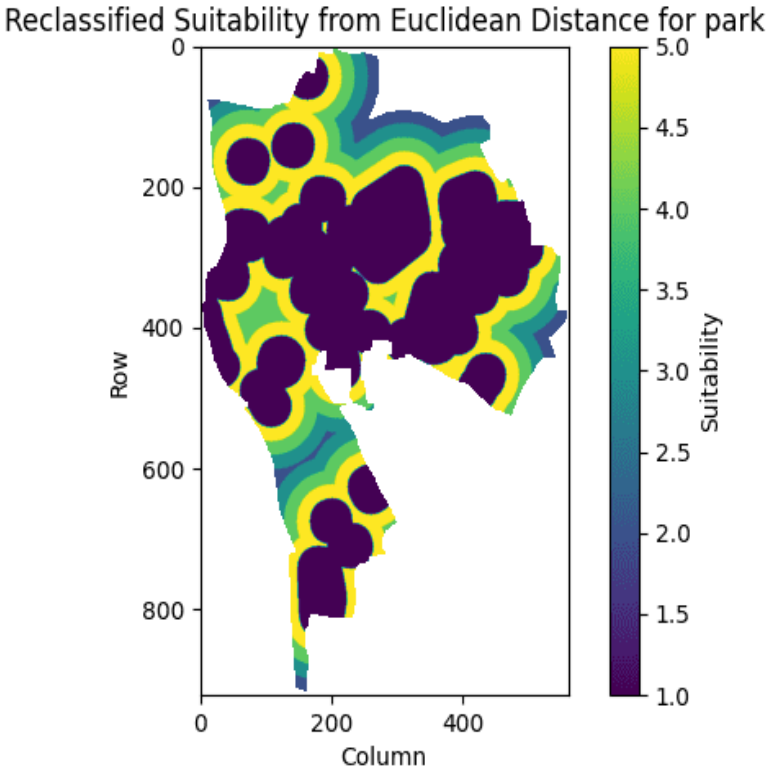


Fig 6.4.Parks reclassification output

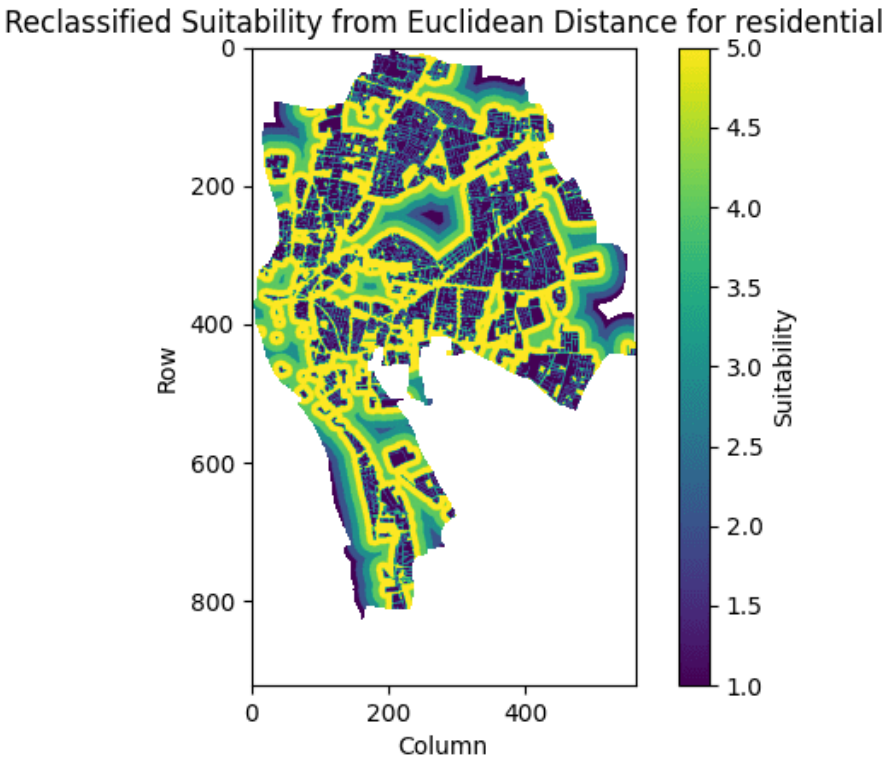
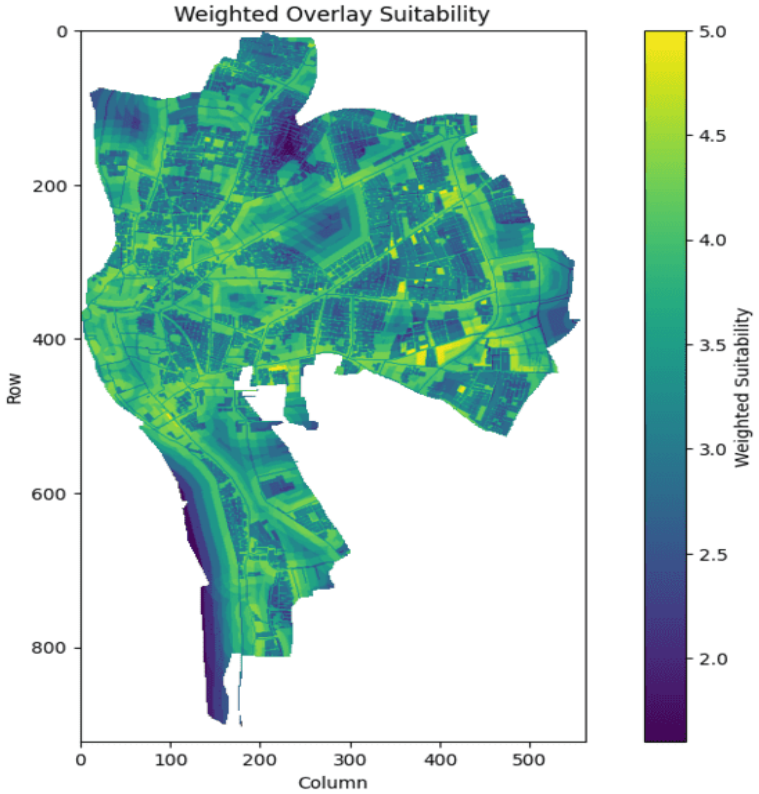


Fig 6.5.Residential reclassification output

D. Land suitability results(with different weights):



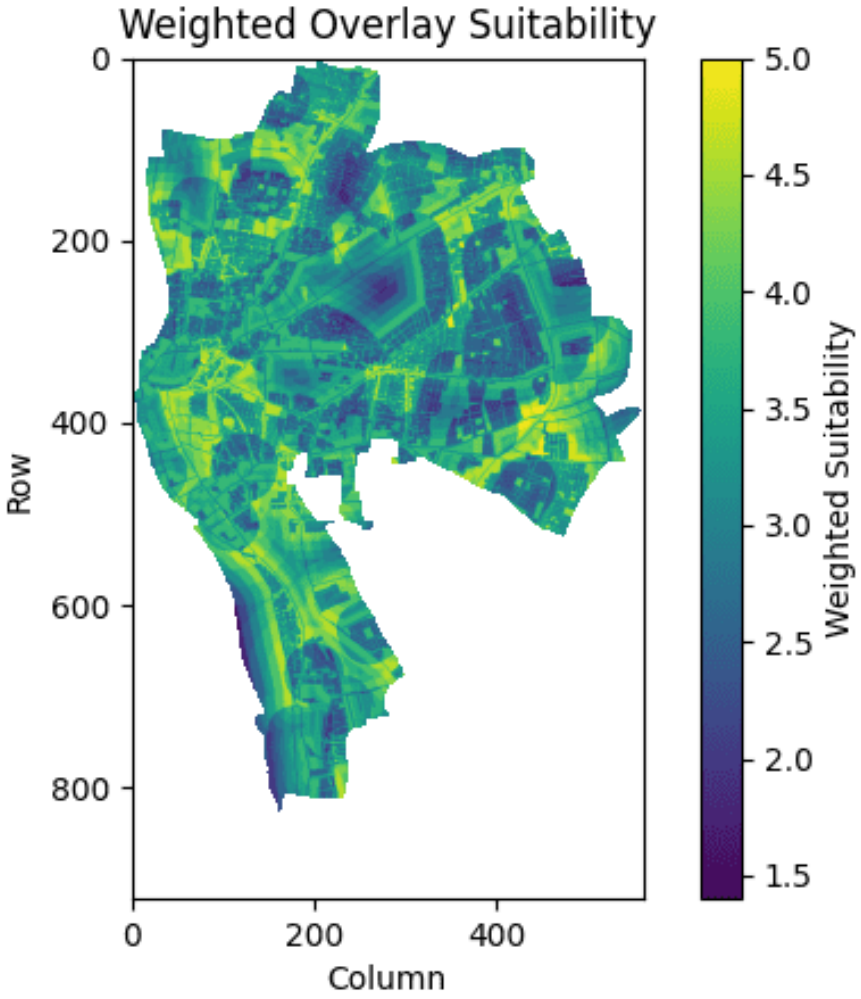


Fig 6.6, 6.7. land suitability outputs with different weights

The categorized maps of roads, parks, and residential areas are combined using a weighted sum technique to create the weighted overlay suitability map. The relative significance of every layer in assessing the overall suitability of the land is reflected in the weights assigned to it. Areas with the highest overall suitability for fresh development are shown on the resulting map.

VII. DISCUSSION

Creating a Python-based tool for urban land suitability analysis to support efficient urban planning was the main goal of this work. The program effectively detects and classifies land according to its the suitability for future development, as demonstrated by the weighted overlay suitability map and the reclassified suitability maps. The maps of appropriateness indicate specific regions that are most suited for development and draw attention to portions of the current infrastructure, including parks and residential areas, that are not as well suited for growth. This is in line with our objective of developing a decision-support tool that urban planners may utilize to make well-informed choices on the usage of their property.

Our results are in line with the general perception that site suitability is greatly influenced by proximity to infrastructure and services when compared to previous research. While most previous research has concentrated on manual or semi-automated GIS-based techniques, our strategy makes use of automated procedures and machine learning. This considerably decreases the time and work needed for large-scale urban planning projects while simultaneously improving the analysis's accuracy.

Our findings have profound implications for land management strategies. The suitability maps may be used by planners and decision-makers to prioritize locations in order of importance for development. In addition, it guarantees that new developments don't interfere with existing infrastructure and communities by having the capacity to categorize existing urban characteristics and keep them out of prospective development zones.

Despite the strengths of our approach, there are limitations to consider. The exactness of our analysis depends intensely on the quality and resolution of the input information. Moreover, whereas our strategy consolidates key factors like proximity to streets and parks, other vital factors such as land cost, zoning controls, and environmental constraints were not included. Future research may upgrade this demonstration by coordinating these extra factors, and by incorporating more advanced machine learning algorithms to refine the suitability analysis further.

VIII. CONCLUSION

In conclusion, this study effectively illustrated how to assess the appropriateness of urban land using a Python-based methodology. The main conclusions emphasize the use of weighted overlays and Euclidean distance computations in determining appropriate locations for urban development. The generated maps provide land suitability a clear visual representation, which may be very helpful to legislators and urban planners in helping them make decisions. The potential for this study to improve the efficiency and data-drivenness of the urban planning process makes it significant. Our methodology provides a scalable solution that can be used in many cities and areas by combining machine learning techniques and automating the analysis. In the end, inhabitants' quality of life may improve as a result of better planned, sustainable urban expansion. We advise practitioners and policymakers to use comparable automated methods for urban land suitability evaluations in light of our findings. By utilizing contemporary computational techniques, they can get more precise and all-encompassing planning results. Subsequent efforts ought to concentrate on including supplementary data layers and improving the model to amplify its suitability and resilience in diverse urban environments.

REFERENCES

[1] Al-Ghorayeb, A., Al-Shaar, W., Elkordi, A., Faour, G., Al-Shaar, M., & Attalah, Y. (2023). Land suitability analysis for sustainable urban development: A case of Nabatiyeh region in Lebanon. J, 6(2), 267–285. https://doi.org/10.3390/j6020020

[2] Abbaspour, M., Mahiny, A. S., Arjmandy, R., & Naimi, B. (2011). Integrated approach for land use suitability analysis. ResearchGate. Retrieved from https://www.researchgate.net/publication/230556927_Integrated_approach_for_land_use_suitability_analysis

[3] Mustafa, A., Singh, M., Sahoo, R. N., & Mishra, A. K. K. (2011). Land Suitability Analysis for Different Crops: A Multi Criteria Decision Making Approach using Remote. . . ResearchGate. Retrieved from https://www.researchgate.net/publication/265143331_Land_Suitability_Analysis_for_Different_Crops_A_Multi_Criteria_Decision_Making_Approach_using_Remote_Sensing_and_GIS

[4] (Ahmad et al., 2020) highlights the potential of using Artificial Neural Networks (ANNs), a type of AI, for land suitability analysis in urban planning.

[5] LAND SUITABILITY ANALYSIS FOR URBAN AND AGRICULTURAL LAND USING GIS: CASE STUDY IN HVITA TO HVITA, ICELAND [PDF] (https://www.sciencedirect.com/science/article/abs/pii/S0264837722004537)

[6] Land Suitability Analysis for Agricultural Crops: A Fuzzy Multicriteria Decision Making Approach [PDF] (https://hindi.iirs.gov.in/iirs/sites/default/files/StudentThesis/final_thesis_prakash.pdf)

[7] (PDF) Land Suitability Analysis for Different Crops: A Multi Criteria Decision Making Approach using Remote Sensing and GIS [PDF] (https://www.researchgate.net/publication/340105035_Land_Suitability_Evaluation_of_Soils_for_Crop_Production)

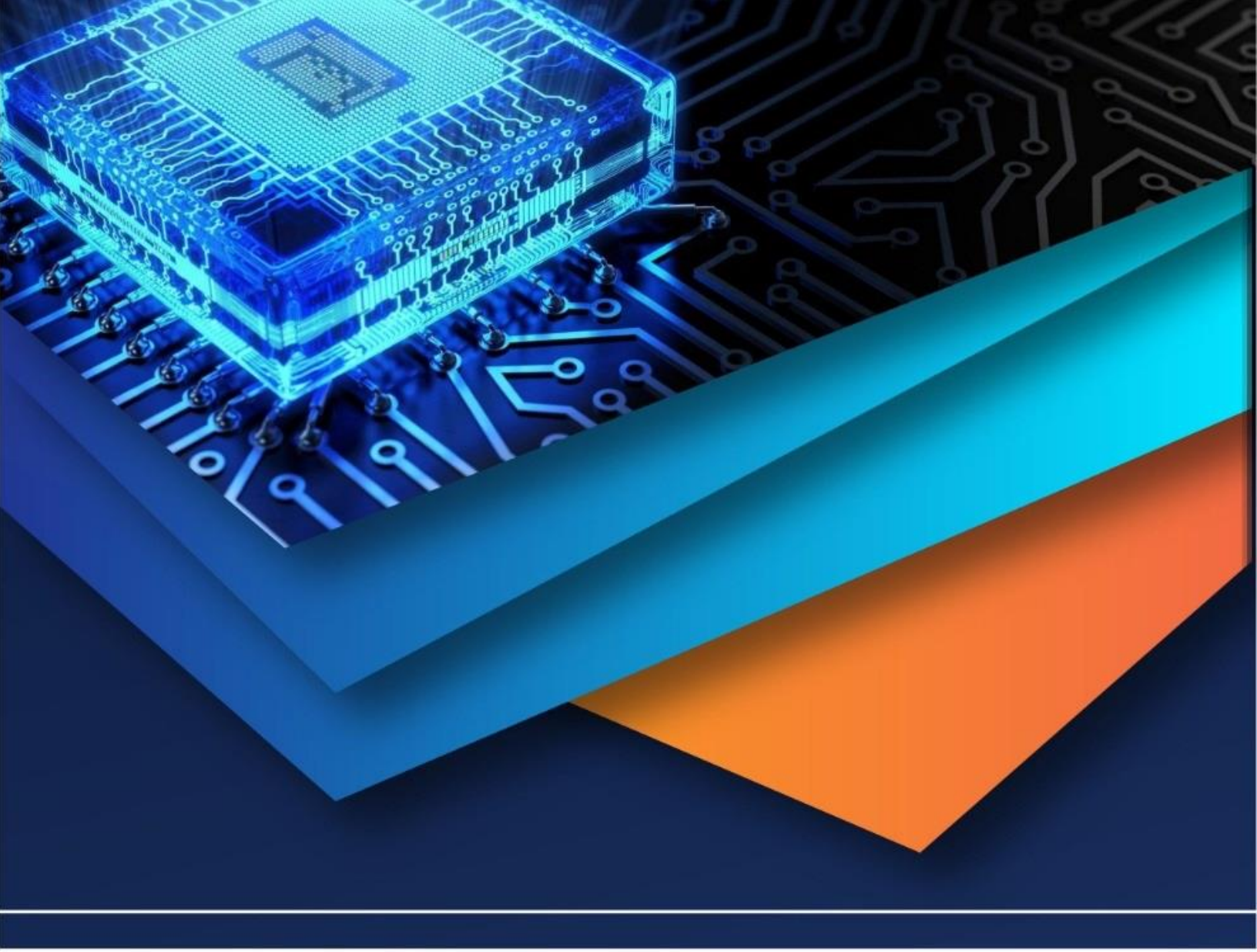
[8] Assessment of land suitability using a soil-indicator-based approach in a geomatics environment | Scientific Reports - Nature https://www.nature.com/articles/s41598-022-22727-7

[9] Rahman, M. M., Islam, M. N., Matsuno, Y., & Tania, N. (2020). Can We Use Machine Learning for Agricultural Land Suitability Assessment? Land, 9(11), 434 (https://www.mdpi.com/1424-8220/23/3/1074)

[10] Mohammadi, M. J., Rezaei, E. M., Pourrahim F., & Namdar, G. H. (2015). Using machine learning for land suitability classification. West African Journal of Applied Ecology, 23(1), 64-73 (https://www.researchgate.net/publication/282925748_Using_machine_learning_for_land_suitability_classification)

[11] Sharma, D. P., Prasad, S., Mondal, S., Patel, J., & Thakur, P. K. (2022). Assessment of Soil Suitability Using Machine Learning in Arid and Semi-Arid Regions. Sustainability, 14(1), 165 (https://www.mdpi.com/2073-4395/14/4/719)

[12] Aburas, M. M., Ho, Y., Ramli, M. F., & Ashaari, Z. H. (2015). A Review of Land Suitability Analysis for Urban Growth by using the GIS-Based Analytic Hierarchy Process. ResearchGate. Retrieved from https://www.researchgate.net/publication/28_8905828_A_Review_of_Land_Suitability_Analysis_for_Urban_Growth_by_using_the_GIS-Based_Analytic_Hierarchy_Process



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY