VELAGAPUDI RAMAKRISHNA SIDDHARTHA ENGINEERING COLLEGE

(AUTONOMOUS **)**

AI TOOLS AND TECHNIQUES

HOME ASSIGNMENT - 1,2,3,4

Submitted To :

DR. Sangeetha Mam

Submitted By:

208W1A1299

208W1A12A0

208W1A12A1

Contents

HOME ASSIGNMENT – 1

HOME ASSIGNMENT - 2

HOME ASSIGNMENT - 3

HOME ASSIGNMENT - 4

HOME ASSIGNMENT - 1

Problem Statement :

Apply back jumping technique for n-queens problem

Abstract :

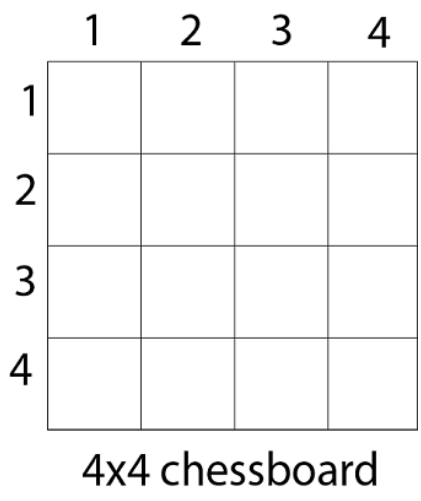
The N-queens problem is a popular classic puzzle where numbers of queen were to be placed on an n x n matrix such that no queen can attack any other queen. The Branching Factor grows in a roughly linear way, which is an important consideration for the researchers. However, many researchers have cited the issues with help of artificial intelligence search patterns say DFS, BFS and backtracking algorithms. The proposed algorithm is able to compute one unique solution in polynomial time when chess board size is greater than 7. This algorithm is based on 8 different series. For each series a different approach is taken to place the queen on a given chess board. General Terms: N-Queen, Knight move, NP-hard.

Introduction:

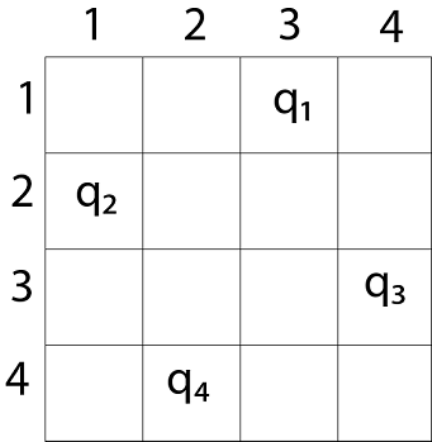
This problem is to find an arrangement of N queens on a chess board, such that no queen can attack any other queens on the board.

The chess queens can attack in any direction as horizontal, vertical, horizontal and diagonal way.

A binary matrix is used to display the positions of N Queens, where no queens can attack other queens.



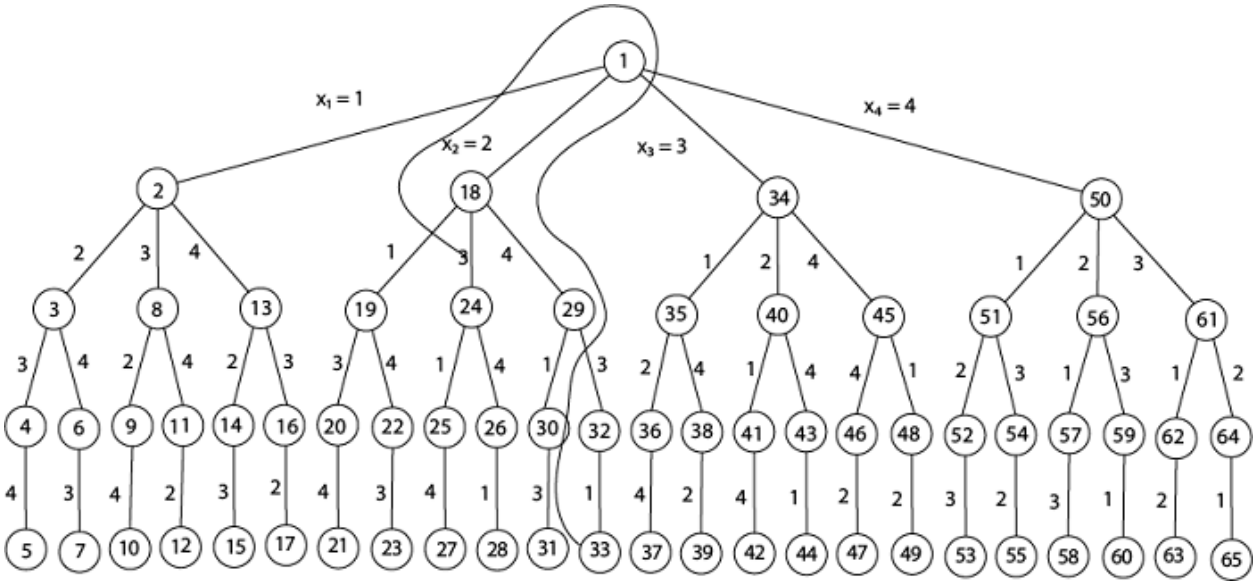
4x4 chessboard

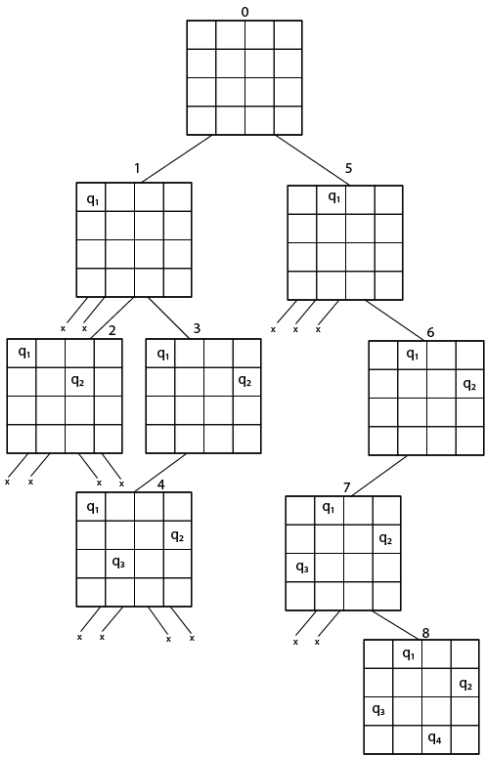


Input:

The size of a chess board. Generally, it is 8. as (8 x 8 is the size of a normal chess board.)

Architecture Diagram:





Algorithm :

Naive Algorithm

```
while there are untried configurations
{
  generate the next configuration
  if queens don't attack in this configuration then
  {
    print this configuration;
  }
}
```

Backtracking Algorithm

1) Start in the leftmost column

2) If all queens are placed
   return true

3) Try all rows in the current column.
  Do following for every tried row.
   a) If the queen can be placed safely in this row
      then mark this [row, column] as part of the
      solution and recursively check if placing
      queen here leads to a solution.
   b) If placing the queen in [row, column] leads to
      a solution then return true.
   c) If placing queen doesn't lead to a solution then
      unmark this [row, column] (Backtrack) and go to
      step (a) to try other rows.

4) If all rows have been tried and nothing worked,
  return false to trigger backtracking.

Output:

The matrix that represents in which row and column the N Queens can be placed.

If the solution does not exist, it will return false.

1 0 0 0 0 0 0 0

0 0 0 0 0 0 1 0

0 0 0 0 1 0 0 0

0 0 0 0 0 0 0 1

0 1 0 0 0 0 0 0

0 0 0 1 0 0 0 0

0 0 0 0 0 1 0 0

0 0 1 0 0 0 0 0

In this output, the value 1 indicates the correct place for the queens.

The 0 denotes the blank spaces on the chess board.

Conclusion:

In conclusion, there is no solution for the N-Queen problem when n=3. **Simply put, there's no possible way to fit 3 queens in a 3x3 chess board without creating any conflict.**

References:

- https://www.researchgate.net/publication/321192822_A_new_approach_to_solve_n-queens_problem_based_on_series#:~:text=Abstract,important%20consideration%20for%20the%20researchers.

- https://medium.com/@jtfeliciano/n-queens-problem-describing-the-problem-12dbd5c1508e#:~:text=In%20conclusion%2C%20there%20is%20no,board%20without%20creating%20any%20conflict.

- https://www.tutorialspoint.com/N-Queen-Problem#:~:text=Input%3A%20The%20chess%20board%2C%20row,is%20a%20valid%20or%20not.&text=Input%20%E2%88%92%20The%20chess%20board%2C%20the,matrix%20where%20queens%20are%20placed.

HOME ASSIGNMENT - 2

Problem Statement:

Apply any AI/ML algorithms Estimate number of households.

Introduction:

Estimation of population distribution and the number of households in small areas over a region is frequently required for the analysis of spatial activities or regional planning, for example, urban land use zoning, disaster prevention in hazardous areas, environmental preservation along rivers, economic development in deprived neighbourhoods, local area marketing, and so forth. To analyze these activities, statistical data in small areas are increasingly demanded in recent years (Smith, 2002). In practice, however, small-area data are not .always easily available. Therefore, most studies use the data aggregated into municipal districts, typically, census data. The aggregated data are, however, often too coarse to analyze activities in small areas. The time resolution is also important but that of census data is too coarse, especially in developing countries, because census survey is carried out every 5-10 years in these countries. The coarse resolution makes timely analysis difficult. To overcome this limitation, a number of methods for producing small-area data is under development.

In recent years, the deep learning method attracts attention in satellite/aerial image analysis. One of the advantages of this method is that it can automatically extract and identify features using supervised data (Krizhevsky, 2012). In remote sensing, the deep learning method is applied to various analyses, such as population estimation (Robinson, 2017; Silvan-Cardenas, 2010), land cover classification (Martin, 2015; Marco, 2015), and change detection in land use (Mou, 2018; Lebedev, 2018). Although the deep learning method applied to satellite images analysis is powerful, it should be noted that satellite images are not sufficient enough to estimate the number of households, because they do not indicate the number of floors of high-rise buildings, building uses, and area characteristics that are indispensable for estimating the number of households. In this study.

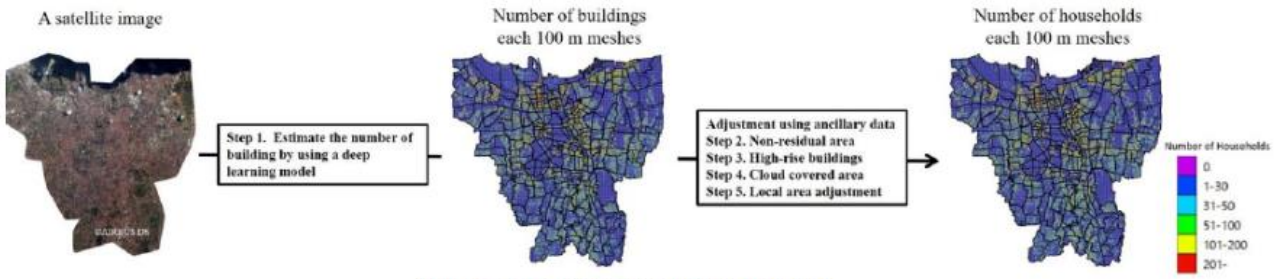


Figure 1. The outline of a proposed method.

Procedure:

1) Dataset:

The target area was Djakarta city. The satellite images used for our analysis were optical satellite images (SPOT) and synthetic-aperture radar (SAR) images (TerraSAR-X). The resolutions of SPOT and TerraSAR-X images are 1.5 meters and 0.5 meters, respectively. SAR images are particularly useful for analyzing Southeast Asian countries, because these countries have many cloudy days. However, the visual resolution of SAR is lower than that of optical images. Therefore, it is difficult to count the number of buildings, especially in densely built areas. In this study, optical images were used together with a residential map (land cover map) created by SAR images.

2) Estimation of the number of buildings by the deep learning method

Our proposed method consisted of five steps shown in Figure 1.
Section 2.2 introduces an application of the deep learning method to the estimation of the number of buildings in100 meter grid cells using satellite images. Techniques for detecting building shapes from high-resolution aerial photographs or satellite images have been proposed by Bischke (2019), Hamguchi (2018) and others. As is noticed from Figure 2, in the case of SPOT optical images, the resolution was 1.5 meters, which was not sufficient enough to detect building boundaries accurately. Therefore, we did not employ a method for detecting the shape of buildings. Instead, we directly estimated the numbers of buildings in 100 meter grid cells using supervised data obtained from OSM data. Stated explicitly, the supervised data were pairwise 100 m grid cells data consisting of patches of SPOT images, and the number of buildings obtained from OSM data. Data in several local areas with low OSM accuracy were corrected by census data or visual confirmation.

Figure 3 shows the architecture of our deep learning model for estimating the number of buildings. The input data were the patches of 100 m grid cells cut out from optical satellite images, and the output data were the estimated number of buildings in 100 m grid cells. The model consisted of six convolutional layers, three pooling layers and one fully connected layer. The convolutional layer and the pooling layer were designed to learn the attribute values that were possibly effective for estimating the numbers of buildings.

Figure 2. SPOT image and building polygons from OSM.

Estimation of the number of buildings by the deep learning method

Our proposed method consisted of five steps shown in Figure 1. Section 2.2 introduces an application of the deep learning method to the estimation of the number of buildings in100 meter grid cells using satellite images.Techniques for detecting building shapes from high-resolution aerial photographs or satellite images have been proposed by Bischke (2019), Hamguchi (2018) and others. As is noticed from Figure 2, in the case of SPOT optical images, the resolution was 1.5 meters, which was not sufficient enough to detect building boundaries accurately.

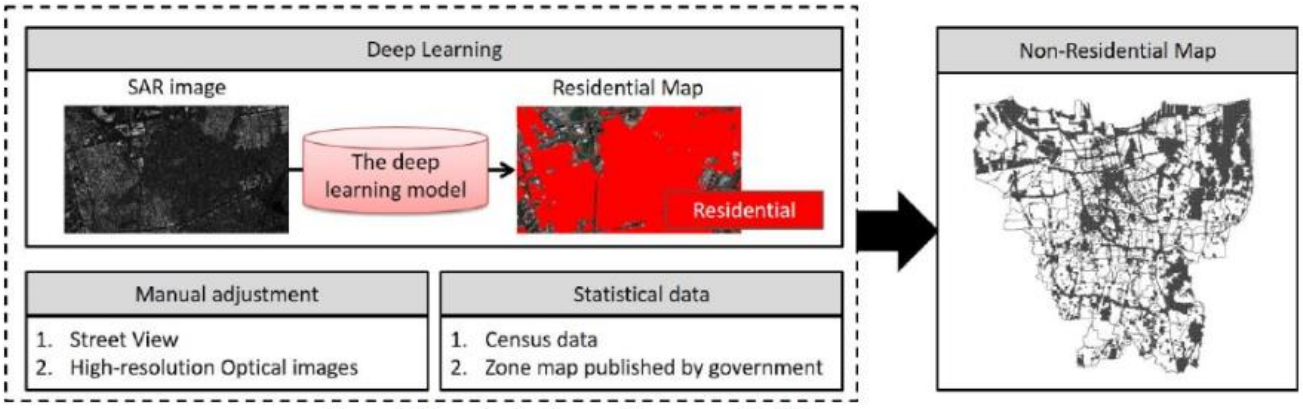


Figure 4. Flow chart of generating non-residential area map

Some other steps:

Adjustment for non-residential area.

Adjustment for high-rise apartment.

Adjustment for cloud covered area.

Adjustment for large deviation rates.

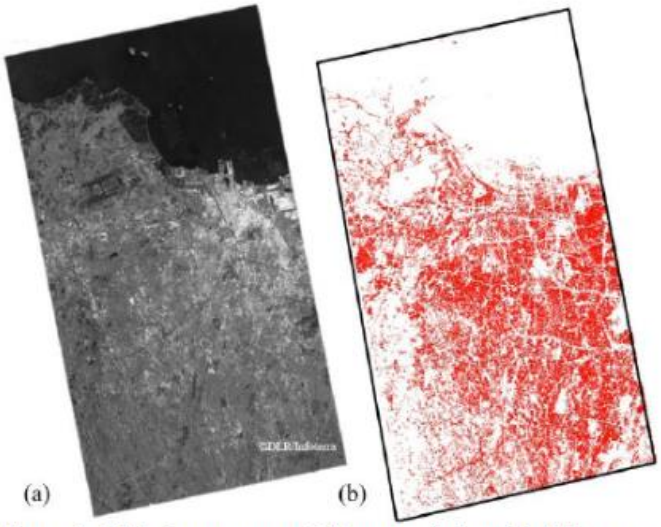


Figure 5. SAR image map and the generated residential map. (a) Input SAR image and (b) residential map generated by the deep learning model (the red points are residential)

Results:

The number of households in 100m grid cells estimated by our deep learning method with several adjustments mentioned in Section 2. Note that in Figure 8, the grid cells where buildings existed but the number of households was zero imply the cells where buildings were public, industrial, and commercial facilities. Our results also showed that the number of households was large in high building density areas (which were known from SPOT images), and small in low density areas. This result suggests that the estimation of the number of households by our method is effective to estimate not only the number of households but also that of buildings. In numerical terms, the number of households estimated by our method was 2,164,945, and that of the census was 2,404,745. The total deviation rate was around 10%, which looks fairly good. This deviation might have resulted from the fact that the distribution of buildings and their surrounding environments in Indonesia have a special kind of provinciality, which was difficult to adjust in our method. The deviation might also resulted from the fact that the updated year of the

OSM, that of satellite images used as supervised data and that of the ancillary data were different fromthat of the census by several years. Therefore, the actual deviation rate may be slightly different.

Conclusion:

The application of the proposed method to Djakarta shows that the method is practically useful for estimating the number of households in 100 m grid cells. This method would become more practical if the following limitations are overcome. First, this method required manual adjustments. To minimize this work, the deep learning model for identifying buildings should be improved. The accuracy of the identification hinges on the resolution of optical satellite images. As is noticed from Figure 2, the boundaries of buildings in SPOT images (the resolution of 1.5m) were blurred in high density areas. Toestimate more accurately, finer resolution of satellite images should be used to detect the boundaries of buildings distinctively. Second, the ancillary data used in our method are not always easily available in developing countries. Therefore, an adjustment method with easily available satellite images and open data should be developed

References:

- Abdikan, S., Sanli, F. B., Ustuner, M., Calò, F., 2016: Land cover mapping using sentinel-1 SAR data. The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, 41, 757.

- Bast, H., Storandt, S., Weidner, S., 2015: Fine-grained population estimation. In Proceedings of the 23rd SIGSPATIAL International Conference on Advances in Geographic Information Systems, pp. 1-10.

- Bischke, B., Helber, P., Folz, J., Borth, D., Dengel, A. 2019: Multi-task learning for segmentation of building footprints with deep neural networks. 2019 IEEE International Conference on Image Processing. pp. 1480-1484.

- Hamaguchi R., Hikosaka S., 2018: Building detection from satellite imagery using ensemble of size-specific detectors. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops.

HOME ASSIGNMENT - 3

Abstract :

The wide unfold of mobiles as hand-held devices end up in varied innovative applications that create use of their ever-increasing presence in our existence. One such application is location chase and monitoring; chase exploitation Geographical Positioning System (GPS) and world System for Mobile Communication (GSM) technology. Land information systems (LIS) provide a technological foundation for decision-making in a wide range of natural resource applications, including scientific, environmental, engineering, and public policy. The LIS provides a 'framework to mix land surface models, relevant knowledge and computing tools and resources' (Kumar et al., 2006). Typically, LIS represents a processed info repository for holding geospatial elements, comprising 'mapping unit' pure mathematics, and connected georeferenced materials like satellite imaging, earth science observations, and predictions, and scanned heritage mapping, that's wherever the thought of analysing the land via the assistance of satellite imaging enters. The system displays the item moving path on the monitor and also the same info may also be communicated to the user's cellular phone, thus we'd like an associate degree application to form one such mil model that offers North American nation automatic classification of land exploitation mil options. With a marker feature with draggable practicality on Google maps, survey officers will regulate coordinate positions additional exactly with real-world things on digital maps.

Introduction:

Recent advances in computing technology, cloud computing, and superior computing area unit paralleled with those in advanced computer science (AI) algorithms and important investment within the European Copernicus Earth Observation program and its watch satellite missions. AI allows automatic detection of spatial patterns in environmental knowledge like satellite pictures supported coaching knowledge. The paradigm of searching for spatial patterns rather than the historic target spectral data in the satellite representational process permits the detection of the latest sorts of land cowl and land use. The free and open convenience of world coverage Earth observation knowledge consistently collected and archived by house agencies. Remotely perceived satellite pictures and knowledge embrace spectral, spatial, and

temporal Resolutions. Spectral statistics involves components of remotely perceived image classification. The most facet that influences the accuracy of a ground object is spatial resolution. Land cowl maps for environmental designing, land use amendment detection, and transportation designing are often done mistreatment temporal resolution. knowledge integration and analysis of urban areas mistreatment mediums resolution remote sensing representational process chiefly focuses on the documentation of engineered up areas or is employed for differentiating between residential, business and industrial zones. to collect numerous geographic knowledge in numerous ways and techniques area unit obtainable.

LITERATURE REVIEW: Over the previous couple of years, deep learning (DL) algorithms have exploded in quality for remote-sensing image process. the most decilitre ideas applicable to remote sensing area unit given during this report, and quite two hundred publications during this field area unit reviewed and examined, the bulk of that were revealed among the last 2 years. A meta-analysis was initial conducted to assess the state of remote sensing decilitre analysis in terms of study targets, decilitre model(s) used, image special resolution(s), study space sort, and classification accuracy achieved. Following that, a comprehensive study of however decilitre has been used for remote sensing image analysis tasks like Volume 6, Issue 3, March – 2021 International Journal of Innovative Science and Research Technology ISSN No:-2456-2165 IJISRT21MAR430 www.ijisrt.com 636 image fusion, image registration, scene classification, object detection, land use and land cowl (LULC) classification, segmentation, and object-based image analysis is performed (OBIA). From pre-processing to mapping, this study covers nearly each application and technology within the field of remote sensing.

METHODOLOGY ArcGIS professional provides information preparation tools for deep learning workflows, conjointly for improved support for deploying qualified models for feature extraction and classification. Similar capabilities are on the market in ArcGIS Image Server within the ArcGIS Enterprise ten.7 update that permits users to deploy deep learning models at scale mistreatment distributed computing. Learn modules within the ArcGIS API for Python to coach deep learning models employing a easy, intuitive API. ArcGIS Notebooks offers a ready-to-use atmosphere for deep learning model coaching. For object detection and classification workflows mistreatment CNTK, Keras, PyTorch, fast.ai, and TensorFlow, ArcGIS provides

intrinsically Python formation functions. You'll be able to conjointly produce your own Python formation feature that employs your most popular deep learning library or a specific deep learning model/architecture.

DEEP LEARNING MODULE Deep learning rekindled the pursuit of artificial intelligence in the direction of a general-purpose computer capable of automating any human-related operation. This is primarily due to a burst of interest in deep machine learning, which uses hierarchical feature representations rather than human-designed features or rules to model high-level abstractions, demonstrating great promise in recognizing and characterizing LC and LU patterns from VFSR imagery.

ArcGIS PRO: Every part of the information science progress may be motor-assisted by ArcGIS tools, as well as information preparation and preliminary information analysis; model training; special analysis; and at last, scattering results through internet layers and maps and driving field operation. ArcGIS professional additionally provides information preparation tools for deep learning workflows, additionally as improved support for deploying qualified models for feature extraction and classification.

ANDROID STUDIO: Linking of the deep learning trained modules from ArcGIS Pro to Android Studio to work as an front-end app giving the satellite imagery, data and also providing the user the benefit of choosing their specific location through the ArcGIS satellite map.

WORKING:

At the very initial stages, we will be training the deep learning data sets using satellite imagery for ArcGIS pro. In which the further segmentation and Labelling of the images will take place. To export training data, there is a labelled imagery layer with the class label for each position, as well as a raster input with all the original

pixels and band information. This land cover classification situation will use a subset of the one-meter resolution Kent County, Delaware dataset as the named imagery layer and World Imagery: Color Infrared as the raster input. With the feature class and raster layer in place, the module is ready to use the export training data() method in the ArcGIS. To export

training data, use the Learn module. In addition to feature class, raster layer, and output folder, we must also specify tile size (image chip size), strid size (distance to transfer each time when creating the next image chip), chip format (TIFF, PNG, or JPEG), and metadata format (how we are going to store those training labels). Depending on the size of the data, tile and stride size, and computing resources, this operation took 15 minutes to 2 hours in our experiment. After the modules have been trained and published as a deep learning package, the deep learning package will export and connect the maps to Android Studio for user access.

Fig. 1: A subset of the labelled data for Kent County, Delaware.

CONCLUSION:

Once the maps are exported and synced with the Android Studio, the users can easily drag the marker on the map to their selected locations. National Agriculture

Imagery Program (NAIP) images are categorised into six major land cover classes by the qualified model: 1) structures, 2) roads or parking lots, 3) water, 4) cultivated, open land or bare land, 5) woodland, and 6) planted or dark cropland, among others. The use of ArcGIS Pro with the ArcGIS API for Python and combine it with deep learning tools (e.g., Keras) to make data planning and modelling easier and integrating it with Android studio for the user output .Using NAIP bands to identify NAIP, the U-Net model performs very well.

References:

1) Abdi, A.M., 2020. Land cover and land use classification performance of machine learning algorithms in a boreal landscape using Sentinel-2 data. GIScience & Remote Sensing, 57(1), pp.1–20.
2) Anon, 2018. Study Data from University of Putra Malaysia Update Knowledge of Data Mining (Integrative image segmentation optimization and machine learning approach for high quality land-use and land-cover mapping using multisource remote sensing data). Information Technology Newsweekly, p.324.
3) Carranza-García, M., García-Gutiérrez, J. & Riquelme, J., 2019. A Framework for Evaluating Land Use and Land Cover Classification Using Convolutional Neural Networks. Remote Sensing, 11(3), pp.Remote Sensing, Jan 2019, Vol.11(3).
4) Díaz-Alcaide, S. & Martínez-Santos, P., 2019. Review: Advances in groundwater potential mapping. Hydrogeology Journal, 27(7), pp.2307–2324.

HOME ASSIGNMENT - 4

Abstract :

The rapid pace of developments in Artificial Intelligence (AI) is providing unprecedented opportunities to enhance the performance of different industries and businesses, including the transport sector. The innovations introduced by AI include highly advanced computational methods that mimic the way the human brain works. The application of AI in the transport field is aimed at overcoming the challenges of an increasing travel demand, CO2 emissions, safety concerns, and environmental degradation. In light of the availability of a huge amount of quantitative and qualitative data and AI in this digital age, addressing these concerns in a more efficient and effective fashion has become more plausible. Examples of AI methods that are finding their way to the transport field include Artificial Neural Networks (ANN), Genetic algorithms (GA), Simulated Annealing (SA), Artificial Immune system (AIS), Ant Colony Optimiser (ACO) and Bee Colony Optimization (BCO) and Fuzzy Logic Model (FLM) The successful application of AI requires a good understanding of the relationships between AI and data on one hand, and transportation system characteristics and variables on the other hand.

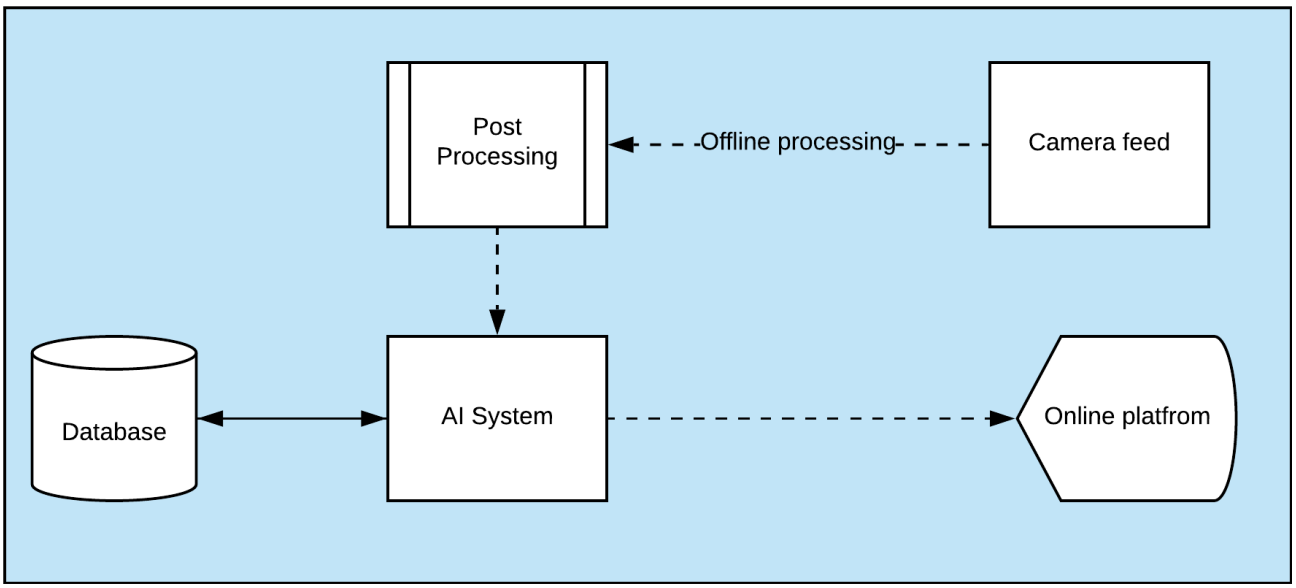
Introduction :

To create a system to monitor and generate report about active commuters in running buses at any point of time To make accurate estimates about probable timetable on the basis of previously occupied running time data To compensate for unforeseen changes in schedule eg. due to festival, national holidays or civil protests.

DataSets:

The Datasets are available in the Github Repositeries and these datasets will act as a input to the your Ai algorithms in order to manage the transportation system. The link to see the dataset in in the below and it is in the for of csv file
https://raw.githubusercontent.com/SmartPracticeschool/SBSPS-Challenge-1029-AI-powered-Public-Transport-Management-System/master/Dataset/Metro_Interstate_Traffic_Volume.csv

Architecture Diagram :



Procedure :

Installing Modules
from tensorflow.keras.models import load_model
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import pickle
import os
!pip3 install --user lazy
!pip3 install --user jaydebeapi
!pip3 uninstall --yes ibmdbpy!pip3 install ibmdbpy --user --ignore-installed --no-deps
!wget -O $HOME/.local/lib/python3.8/site packages/ibmdbpy/db2jcc4.jar https://ibm.box.com/shared/static/lmhzyeslp1rqns04ue8dnhz2x7fb6nkc.zip
bus_limit=200
static_model=load_model("model24hrspredict.h5")
dynamic_model=load_model("model2hrspredict(6hrsinput).h5")
static_scaler=pickle.load(open("scalernew.pkl",'rb'))
dynamic_scaler=pickle.load(open("scaler6hrswala.pkl",'rb'))

```
def schedule(prediction):
sch=[]
for i,j in enumerate(prediction):
print(f"{int(round(j[0]/bus_limit))} AT {str(i).zfill(2)}:00")
sch.append([int(round(j[0]/bus_limit)),i])
return sch
def special_events(prediction): start=prediction['START']
stop=prediction['STOP']


magnitude=prediction['MAGNITUDE']
prediction['TIMESERIES'][start:stop]=prediction['TIMESERIES'][
start:stop]*(1.0+magnitude/100)
return prediction
def static_time_table(data,special_timeseries=None):
y_pred=static_model.predict(data)[0]
if special_timeseries!=None:
special_timeseries['TIMESERIES']=y_pred[0]
y_pred=special_events(special_timeseries)['TIMESERIES']
y_pred=static_scaler.inverse_transform(y_pred.reshape(-
1,1))
number=schedule(y_pred)
return [number,y_pred]
from db2 import Db2
db2 = Db2(table_name='INTERSTATE_TRAFFIC')
db2.last24hours()db2.last6hours()
db2.ida_traffic_pd.head()
```

Data cleaning

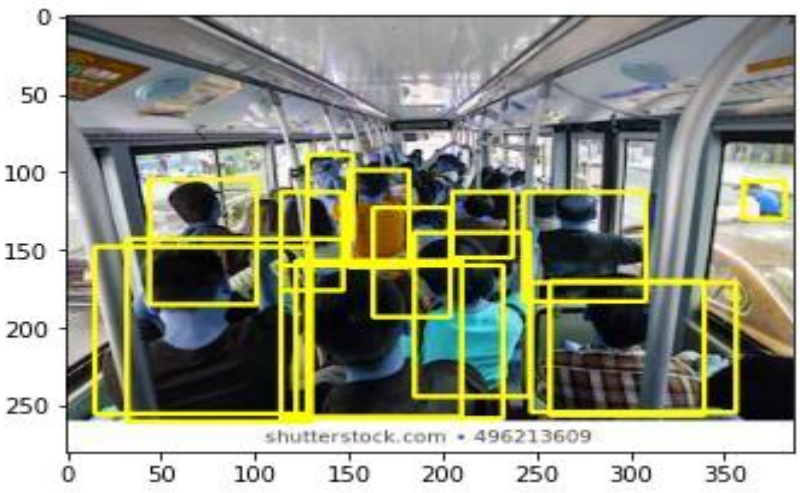
```
db2.last6hours()
db2.ida_traffic_pd.head()
data=pd.read_csv("../Dataset/Metro_Interstate_Traffic_Volume.csv")
data['date_time']=pd.to_datetime(data['date_time'],infer_datetime_format=True)
data=data.iloc[:,-2:]
data=data.set_index('date_time')
data['traffic_volume']=static_scaler.fit_transform(data['traffic
_volume'].values.reshape(-1,1))
num,pred=static_time_table(data['traffic_volume'][- 24:].values.reshape(1,-1,1))
```

```
plt.plot(pred,label='Prediction')
plt.xlabel("Hours")
plt.ylabel("Expected Crowd")
plt.legend()
```

Checking Dynamic Model :

```
data['traffic_volume']=dynamic_scaler.fit_transform(data['traffic_volume'].values.re
shape(-1,1))
y_pred=dynamic_model.predict(data['traffic_volume'][- 6:].values.reshape(1,-1,1))
y_pred=dynamic_scaler.inverse_transform(y_pred)
l=list(dynamic_scaler.inverse_transform(data['traffic_volume'] [-
6:].values.reshape(1,1)))+list(y_pred[0])
plt.plot(l,color='black',label="prediction")
plt.axvline(5,color='r',label="start")
plt.axvline(7,color='blue',label="stop")
plt.xlabel("Hours")
plt.ylabel("Expected Crowd")plt.legend()
def dynamic_changes(data,sch,start_time):
pred=dynamic_model.predict(data)[0]
pred=schedule(dynamic_scaler.inverse_transform(pred.resha
pe(-1,1)))
#temp=sch[start_time:start_time+2]
sch[start_time:start_time+2]=list(map(lambda x:x[0],pred))
return sch
dynamic_changes(data['traffic_volume'][- 6:].values.reshape(1,-
1,1),list(map(lambda x:x[0],num)),0)
```

Results :



Conclusion :

From the above experiment and procedure and algorithms We can finally conclude that the model has an 98.098% accuracy that means the transportation problem has been solved almost perfectly.

References :

https://github.com/SmartPracticeschool/SBSPS-Challenge-1029-AI-powered-Public-Transport-Management-System

https://github.com/Intelligent-Vehicle-Perception/Intelligent-Vehicle-Perception-Based-on-Inertial-Sensing-and-Artificial-Intelligence