

Prediction of Development Ratio of a Particular Area Using Satellite Imagery

Prasad Bhamare¹, Aditya Jhuria², Dnyaneshwar Mhatre³, Rashtratej Sawai⁴

Abstract- *The government is unable to estimate Development status of a remote area and also they are unable to help them. Because government only has their satellite image as a record and they can only see that area through map but through this image they cannot get status about that area. So, considering this satellite image of an area, there is a profound need to detect status of the remote area. In this survey paper, we have three major tasks to design our project. First of all, we have to find datasets related to satellite image and then designing of user-friendly desktop application and another task is to authenticate the user and preprocess the input satellite image.*

Keywords- *PyQT, NumPy, OpenCV, Satellite Image, LandSat 7, Google Earth, MySQL*

I. INTRODUCTION

There are so many regions in the world where humans exist but they have no facilities for their livelihood. They don't even have basic necessities of life like water, food and so on. Some region has lack of only one factor and some regions have lack of all the factors. Some region has water but not electricity while another region has home but not any other necessities. For such type of regions, some organizations are ready to help them with the support of government of that country but due to lack of communication from that region, the organization knows only the location of that region. They don't even know what the basic necessities of that region are?

In that case, the organization can only have the satellite image of the region and they try to determine necessities by observing satellite image. But by only observing that region through satellite image we cannot estimate the presence of the factors on that region. To solve this kind of problem we are introducing an application to predict Development status of a region.

The system which we are designing has the ability to identify some major factors which are very basic necessities of a region and they are electricity, water supply, agricultural field, roads. One more factor we are using for estimate Development status of region is roof top of the house. Roof top is a very essential factor for our system. For prediction of Development status, our system takes satellite image and then this satellite image is compared

with our trained model which contains all these major factors present within it and after comparing these factors we get prediction of the status of that satellite image in the form of percentage of presence of factors in the image and by considering this percentages of factors we are predicting Development status.

To achieve required result, application is designed through python language and using its libraries. So, to design user friendly desktop application, PyQT library method is used in python language. To preprocess datasets of satellite images, we are using OpenCV library method and through preprocessing of image, we convert our input satellite image into grayscale image, contour image and smoothen image. To authenticate the user, we are using MySQL database connectivity.

The datasets of satellite image is collected from LandSat 7 and Google Earth website and along with this platform, there is another application (Magic Puzzle) on which by providing longitude and latitude we get satellite image of that area. Some times, satellite image is downloaded in the form multiple tiles and by combining all the tiles, we are getting a single whole image.

In this way, we are collecting datasets of satellite image.

The overview of survey paper is that it contains literature survey and then it contains System Architecture and then conclusion and references along with URL of datasets of satellite images.

II. LITERATURE SURVEY

Shailesh M. Pandey and Tushar Agarwal, propose a two-step approach for predicting poverty in rural regions of India from satellite imagery. First, we train a multi-task fully convolutional model to predict three developmental parameters – the main material of the roof, source of lighting and source of drinking water – from satellite imagery. Using only satellite imagery as input, according to paper, we are able to estimate income and poverty close to the true values collected on the ground by significant manual effort and monetary expense.

BARAK OSHRI and ANNIE HU, Data on infrastructure quality outcomes in developing countries is lacking, and this work explored the use of globally available remote sensing data for predicting such outcomes. Using Afrobarometer survey data, according to paper we

introduced a deep learning approach that demonstrates good predictive ability. For all these endeavors, data will form a core issue. The quality of a deep model heavily relies on adequate data available, and a large focus should be towards making better use of existing image and survey data, through strong cataloging and collating efforts. However, our results demonstrate the proof of concept that satellite imagery can be used to predict infrastructure quality.

Anthony Perez and Christopher Yeh, Our results show that the current state-of-the-art in satellite-based poverty prediction lends itself to predicting relative wealth within a single country where some ground truth data is available, but may struggle with extrapolating across country borders. Using some combination of nightlights and predictions from the proposed models may yield further improvements.

Boris Babenko and Jonathan Hersh, Presents the CNN predictions for urban areas using imagery for either Digital Globe or Planet, using the 10% withheld validation sample. We present R2 estimates that show the correlation between predicted poverty and benchmark poverty as measured in the 2015 Intercensus. R2 is estimated at 0.61 using the Digital Globe imagery, and 0.54 using Planet imagery. Recall we can only compare urban areas due to lack of coverage of rural areas for Digital Globe. The drop in performance is modest but not severe, especially considering that Planet imagery offers daily revisit rates of the earth's landmass. Poverty estimates for urban areas in Mexico are mapped.

III. PROBLEM STATEMENT

- Identifying Development Status of a Particular area.
- Helping Governments and NGOs recognize Development prone areas
- Taking Satellite Images and using Image Processing to identify Structures
- Using Machine learning to recognize a pattern

IV. PROPOSED WORK

It is the time to articulate the research work with ideas gathered in above step by adopting all the below approaches:

A. Open Computer Vision (OpenCV)

A prior knowledge on Python and Numpy is required before starting OpenCV. Python is a general purpose programming language which became very popular in short time mainly because of its simplicity and code readability. It enables the programmer to express his ideas in fewer lines of code without reducing any readability. Numpy is a highly

optimized library for numerical operations. It gives a MATLAB-style syntax. All the OpenCV array structures are converted to-and-from Numpy arrays. So whatever operations you can do in Numpy, you can combine it with OpenCV, which increases number of weapons in your arsenal. OpenCV is a cross-platform library using which we can develop real-time computer vision applications. It mainly focuses on image processing; video capture and analysis including features like face detection and object detection. To install OpenCV in python in linux use command:- `pip install opencv`.

B. Convolutional Neural Network (CNN)

CNN uses special convolution and pooling operations and performs parameter sharing. To apply CNN method on the application, there is a need of TensorFlow and Keras library for the feature extraction. In our application, we are using three layers of CNN.

1. TensorFlow

TensorFlow is an open-source machine learning library for research and production. TensorFlow offers APIs for beginners and experts to develop for desktop, mobile, web, and cloud. TensorFlow is a framework created by Google for creating Deep Learning models. TensorFlow is an open source machine learning framework for all developers. It is used for implementing machine learning and deep learning applications. To develop and research on fascinating ideas on artificial intelligence, Google team created TensorFlow. TensorFlow is designed in Python programming language, hence it is considered an easy to understand framework. TensorFlow is also called a "Google" product. It includes a variety of machine learning and deep learning algorithms. TensorFlow can train and run deep neural networks for handwritten digit classification, image recognition, word embedding and creation of various sequence models. The steps, which require the execution and proper dimension of the entire network:

- Include the necessary modules for TensorFlow and the data set modules, which are needed to compute the CNN model.
- Declare a function called `run_cnn()`, which includes various parameters and optimization variables with declaration of data placeholders. These optimization variables will declare the training pattern.
- In this step, we will declare the training data placeholders with input parameters. We can reshape the tensor according to our requirements.
- Now it's time to create convolutional layers.
- Let us flatten the output ready for the fully connected output stage - after two layers of stride 2 pooling with the dimensions of 28 x 28, to dimension of 14 x 14 or minimum 7 x 7 x,y co-ordinates, but with 64 output channels. To create

the fully connected with "dense" layer, the new shape needs to be $[-1, 7 \times 7 \times 64]$.

- We should set up recording variables. This adds up a summary to store the accuracy of data.

2. Keras

Models in Keras can come in two forms – Sequential and via the Functional API. For most deep learning networks that you build, the Sequential model is likely what you will use. It allows you to easily stack sequential layers (and even recurrent layers) of the network in order from input to output. The functional API allows you to build more complicated architectures, and it won't be covered in this tutorial. Next, we add a 2D convolutional layer to process the 2D MNIST input images. The first argument passed to the Conv2D() layer function is the number of output channels – in this case we have 32 output channels (as per the architecture shown at the beginning). The next input is the kernel_size, which in this case we have chosen to be a 5x5 moving window, followed by the strides in the x and y directions (1, 1). Next, the activation function is a rectified linear unit and finally we have to supply the model with the size of the input to the layer (which is declared in another part of the code – see here). Declaring the input shape is only required of the first layer – Keras is good enough to work out the size of the tensors flowing through the model from there. Next we add a 2D max pooling layer. The definition of the layer is dead easy. We simply specify the size of the pooling in the x and y directions – (2, 2) in this case, and the strides. That's it. Next we add another convolutional + max pooling layer, with 64 output channels. The default strides argument in the Conv2D() function is (1, 1) in Keras, so we can leave it out. The default strides argument in Keras is to make it equal to the pool size, so again, we can leave it out. The input tensor for this layer is (batch_size, 28, 28, 32) – the 28 x 28 is the size of the image, and the 32 is the number of output channels from the previous layer. However, notice we don't have to explicitly detail what the shape of the input is – Keras will work it out for us.

V. SYSTEM ARCHITECTURE

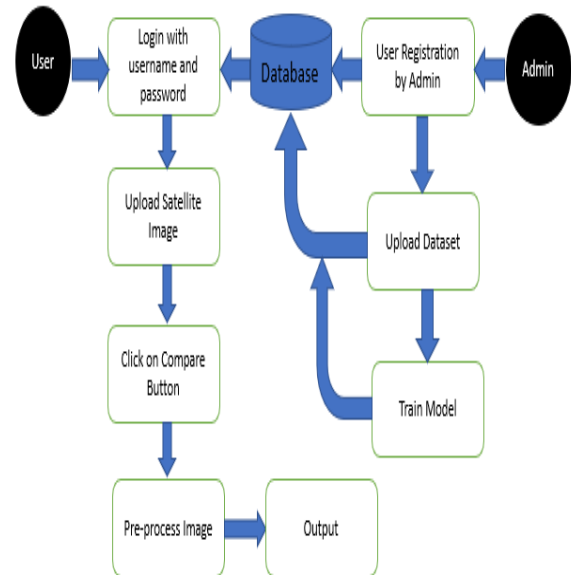


Figure No. 3.1 System Architecture

In the diagram, there is flow of our project.

1. The whole architecture is made by PyQt library used in python language. PyQt library gives all the necessary stuff related to GUI design. PyQt provides us display screen, buttons and so on. So, In this way PyQt helps us in design GUI.
2. After designing of GUI, another task is to authenticate valid user for operating application. To deal with this task, we are using MySQL database to store data of username and password and through this, user can authenticate easily.
3. Another task is to preprocess the input image which can be done by OpenCV library of python. By using this library, image is converted into grayscale image, contour image and smoothen image.
4. The major task of this survey paper is to collect datasets of satellite image and to achieve this result, we are working on google Earth images, LandSat 7 images and also take help of magic puzzle application on which, we are providing latitude and longitude of a particular area and as a result, we are getting satellite image of that area.

In this way, we achieve our all the tasks to achieve our project goal.

VI. CONCLUSION

Predict Development Ratio Status of an Area from Satellite Image Using Deep Learning. In this type of application, we have first collect datasets of satellite image and after that make a desktop application so that user can able to predict Development status. To predict status of a satellite image, we have use preprocessing of a input image so that features can be easily detected from input mage and to achieve this we are using Opencv library. In this way, we are successfully implementing all the tasks of the survey paper.

VII. FUTURE WORK

In future work, we apply CNN methods for training and testing model through collected datasets of satellite images and try to predict Development status of input satellite image. Along with CNN methods, we can use decision tree algorithm or naïve baiyes to predict whether that area is Highly Developed or less Developed in terms of occurrence of factors in that input image.

VIII. REFERENCES

- [1] Pandey, S. M., Agarwal, T., & Krishnan, N. C. (2014). *Multi-Task Deep Learning for Predicting Poverty from Satellite Images*. Ropar: The Thirtieth AAAI Conference on Innovative Applications of Artificial Intelligence (IAAI-18).
- [2] OSHRI, B., HU, A., ADELSON, P., & LOBELL, D. (2018). *Infrastructure Qulity Assessment in Africa using Satellite Imagery and Deep Learning*. Stanford: Association for Computing Machinery.
- [3] Perez, A., Yeh, C., Azzari, G., Burke, M., Lobell, D., & Ermon, S. (2017). *Poverty Prediction with Public Landsat 7 Satellite Imagery and Machine Learning*. California: 31st Conference on Neural Information Processing Systems.
- [4] Babenko, B., Hersh, J., Newhouse, D., Ramakrishnan, A., & Swartz, T. (2017). *Poverty Mapping Using Convolutional Neural Networks Trained on High and Medium Resolution Satellite Images, With an Application in Mexico*.

California: 31st Conference on Neural Information Processing Systems.