Building Information Classification Using Street View Images

[1]Bhargav Petla
[2]Sai Teja Gilukara
[3]Ajay Choudhury
EECS Department
BS-MS, IISER Bhopal

***Abstract*—** Land-use classification based on spaceborne or aerial remote sensing imagery has received much attention in recent decades. This type of classification is typically a patch-wise or pixel-wise labeling of the entire image. A categorization map based on individual buildings, on the other hand, is far more helpful for many applications, such as urban population density mapping or urban utility planning. However, such semantic categorization still presents some fundamental issues. For example, in India, usual datasets of individual buildings do not exist, and how to recall delicate borders of individual buildings?. In this project, we proposed a general framework for classifying the functionality of individual buildings. The proposed method is based on Convolutional Neural Networks (CNNs), which classify façade structures from street view images such as Google Street View and stock images which we collect physically and through image databases. We created a benchmark dataset that was used for training and evaluating CNNs. In addition to that, we will save our best-trained model and use it to predict building image types.

***Keywords*— building, instance, land-use, classification, land, land-usage, CNN, ReLu, Pooling, normalisation, batches, accuracy**

I. INTRODUCTION

Over the last few decades, land cover classification from Earth Observation (EO) images in complicated metropolitan contexts has focused on remote sensing. Furthermore, high-resolution spaceborne and aerial images are few information sources for monitoring large-scale urban development. However, in EO-data, the transition from land cover to land use is complex, relying mainly on the geometry and look of individual buildings and the patterns they group. The correlation of physical indications such as building volumes, density, and alignment has been used to infer building usage, such as commercial, residential. However, as we move to a smaller level of intrinsic urban scale, such pattern analysis cannot be transferred to the classification of individual buildings. As can be seen, classifying land use at the level of individual structures is not an easy operation. Typically, such a classification map is only available through city cadastral databases, which are inaccessible or, in some cases, non-existent. Updating such databases without the use of automated procedures can be time-consuming. As a result, achieving a building instance-level classification automatically is required and can be advantageous for applications related to urban planning. Towards the automatic classification of individual buildings, the challenges are twofold. Firstly, less data on street view images makes obtaining data of various classes complicated, especially for suburban areas in developing countries like India. Secondly, The available dataset might not be huge to train a neural network from scratch as the accuracy of neural networks in such classification cases tend to grow with the growing size of the dataset. In this project, we propose a general framework to tackle the challenges mentioned above; we collected our data from various sources, including taking live pictures and pre-available Indian data sets; we also used a pre-trained model and fine-tuned it produce desired results. Therefore, We built a benchmark dataset of building street view images to train Convolutional Neural Networks (CNNs) for the classification over large areas, as CNN has been demonstrated its powerful ability in tasks of this sort. We stored the best accuracy model to predict the building types.

II. CONTRIBUTION

With this idea of individual building instance classification into different categories. we plan to contribute to the evolved framework for land usage planning, urban restructuring, slum redevelopment and energy efficient area planning. Remote sensing images provide limited information and such information is efficient only on large scale development and restructuring. Individual building instance classification can be helpful and act as a base for many important projects.

Furthermore, we have also created a database of various categories of building instances, with this information we can give a head start to other projects that depend on works related to city, slum or any other area management. Geotagging of such dataset can open up new facets of various researches and can help create a scan of an area that can be passed on to various machine learning algorithms to analyse and study the restructure and development ideas over an area. With extension of our proposed framework, studies involving characteristics, traditions and climate of an area can be fuelled as designs of buildings and structures are heavily influenced by the area's climate, traditions and culture.

III. BACKGROUND

Land-use classification based on remote sensing images has been in studies over the past decade. Such classifications are usually done for better land planning, symmetry planning and land management. But for many applications like urban population density mapping or utility planning, a map utilising classification based on individual buildings is much more useful, informative and efficient. Classification of land usage patterns based solely on top-view data can produce subtle results as major geometrical information like height, volume etc. are missing in remote sensing images. Such studies are good for density planning of an area but for land usage pattern detection or classification it has some serious challenges. Here are some of the gaps in the previous studies:

- Land use classification can be done better with geometrical data with height information than just a top view of the structures.
- The correlation of structure data like volume, density, alignment, and height is high with the individual building-type classification and such data is usually missing in remote sensing images.
- Intricate analysis of individual building information can lead to a better understanding of the land usage pattern and area structure analysis.

For example in the figure 1.1 classification of the area is done as a commercial area but the area within the classified area is religious, such errors can be alleviated using individual building instance classifications.

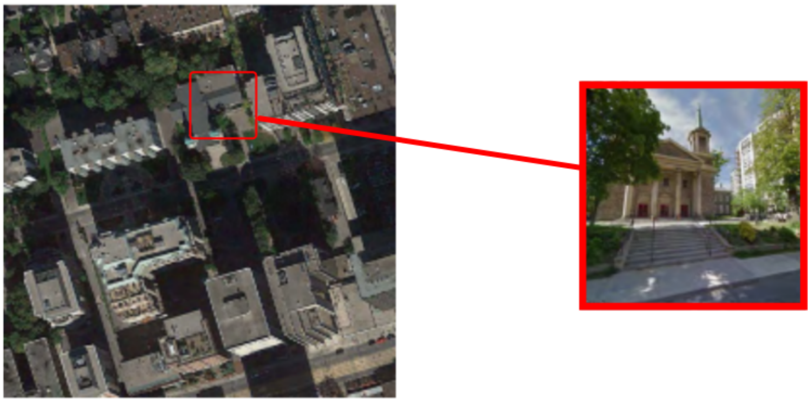


Fig.1. Cons of land area classification using remote sensing data alone.

To solve such issues using the analysis and classification of individual building instances using a convolutional neural network fed with street view images of individual building instances.

IV. MATERIALS AND METHODS

4.1) Material

To train a building instance classifier, we first build a corresponding street-view benchmark dataset, which contains a total of 620 images from Six classes, i.e. *apartment, Religious building, garage, Individual house, office building, shops, slums* and there are around 110 images for each building class as shown below.

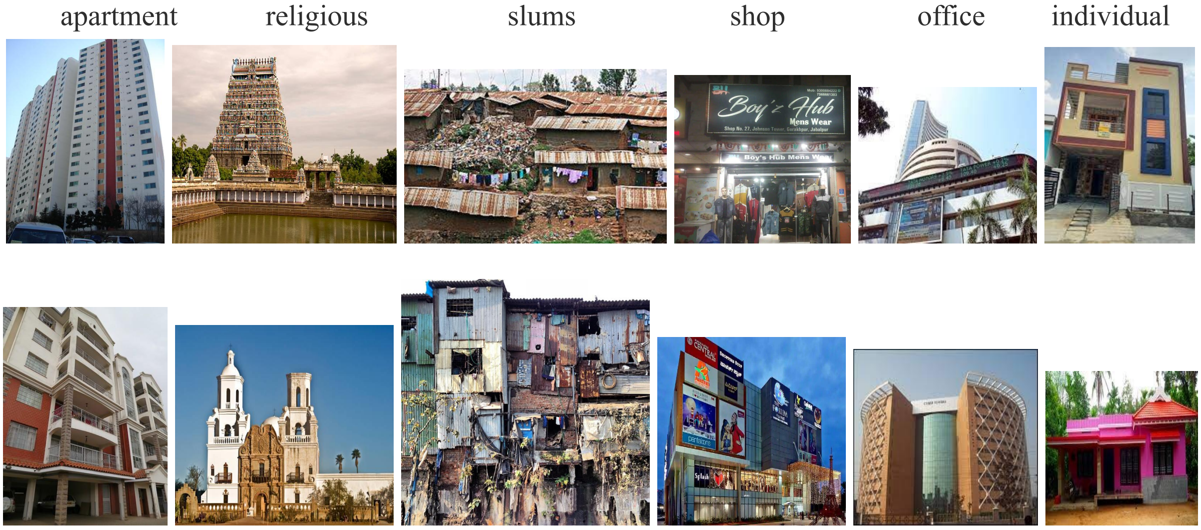


Fig 1: Examples of the benchmark dataset. It contains 620 street view images of buildings with six classes. The images are downloaded from Google StreetView and various stock images mixed with live images.

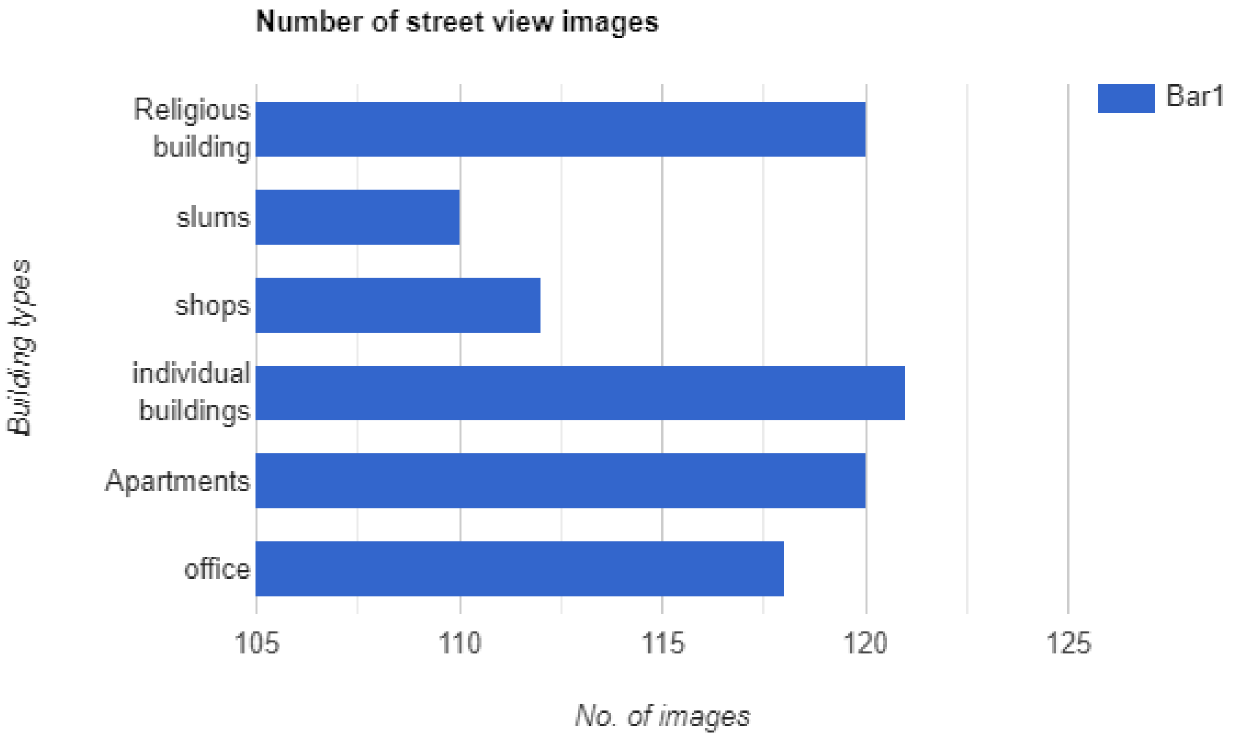


Fig 2: Number of street view images of each building class

Apartments building	A building consists of blocks of apartments
Office building	A building where non-specific commercial activities take place
Religious building	A building that was built for religious activities
shops	A building primarily used for selling goods that are sold to the public
slums	A building for poor people
Individual building	A dwelling unit inhabited by a single household

Table 1: Building class descriptions

Further, we divide our dataset into 3 files, seg_train which consists of training data, seg_test which consists of testing data and seg_pred which contains unclassified images which we use for prediction based on our best training model.

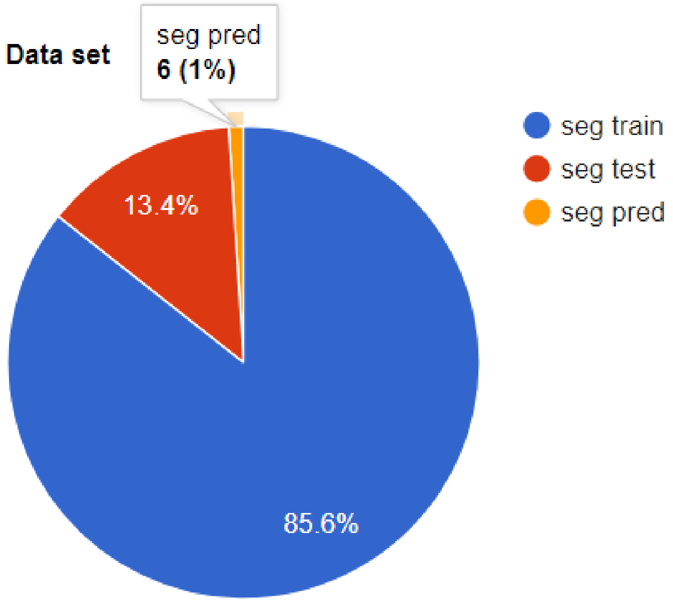


Fig 3: The folder seg_train consists of 512 images, seg_test contains 80 images and seg_pred contains 6 images.

4.2) Methodology

1. Data reading and pre-processing
2. Cnn network using different layers
3. Training Cnn with our dataset and saving the best model based on our accuracy and predicting

4.21. Pre-processing

We collected data from various sources such as stock images and live images and then we used PyTorch library instead of tensor flow because of its pythonic nature and ease of extendability.

a) Resize

We resize all images to 150 height and 150 widths just to be sure that we are using all images of the same size.

b) Random horizontal flip

We used documentation Technic, a random horizontal flip that has a default probability of 0.5 which means for each epoch probability the image is original or the horizontal flip is 0.5. We did this to add a variation to our data set.

c) Transform.to tensor

We used transform.to tensor to change the pixel range; it also changes the data type of our image from NumPy to tensor. We did that because PyTorch takes tensors as input.

d) Normalise

We changed the range from 0-1 to -1 -1, in the input 2*3 matrix, the column represents the RGB channel and the row represents the mean and standard deviation and the new pixel will be calculated using the formula of x-mean/std. where x is the old pixel value.

e) Dataloader

In PyTorch we will feed the data in the form of a data loader, data loader basically helps in reading the data and feeding it to the model for training in batches.

V. WORKFLOW

Introduction to CNN: Image classification is the process in which the input is an image and the output is class or the probability that the image falls in that class. Humans brains are built to do this classification easily but computers need to be taught. We use the Convolution Neural Networks (CNN).

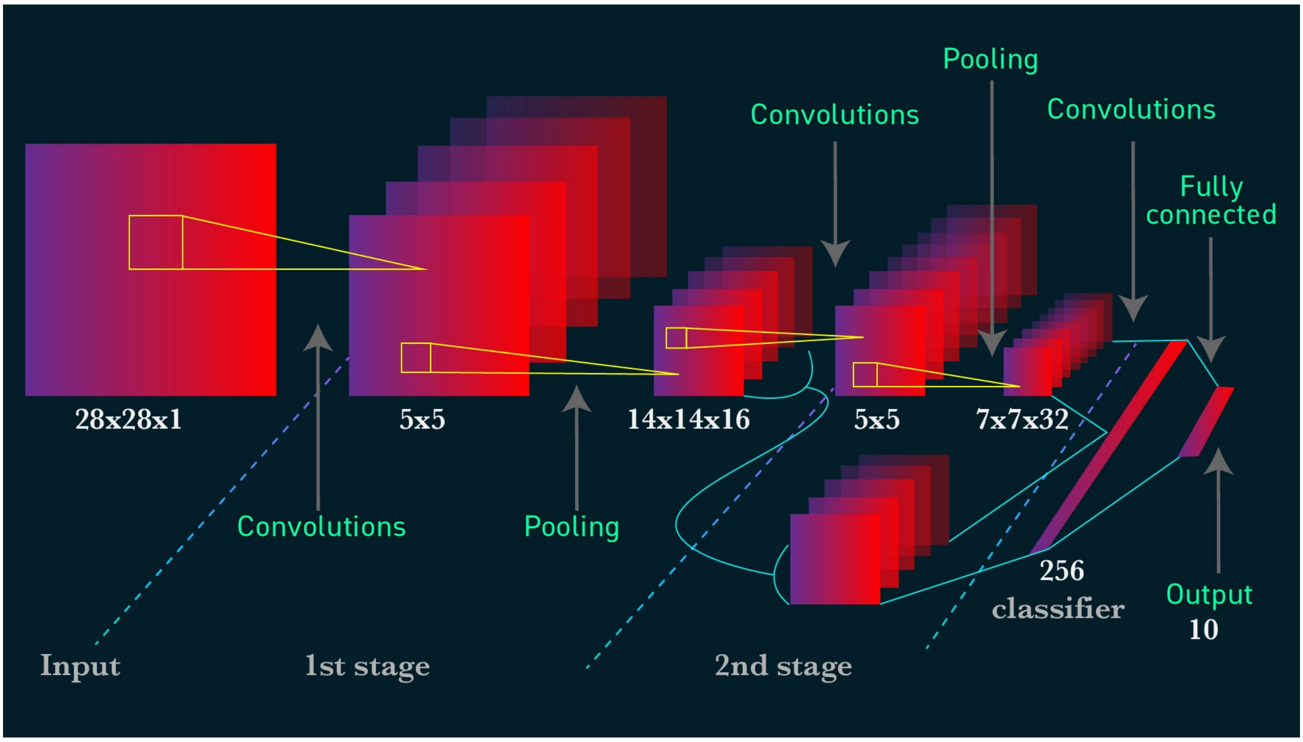


Fig 4: Basic architecture of a CNN

A CNN structure has

- Convolutional layers
- ReLU layers

- Pooling layers
- a Fully connected layer

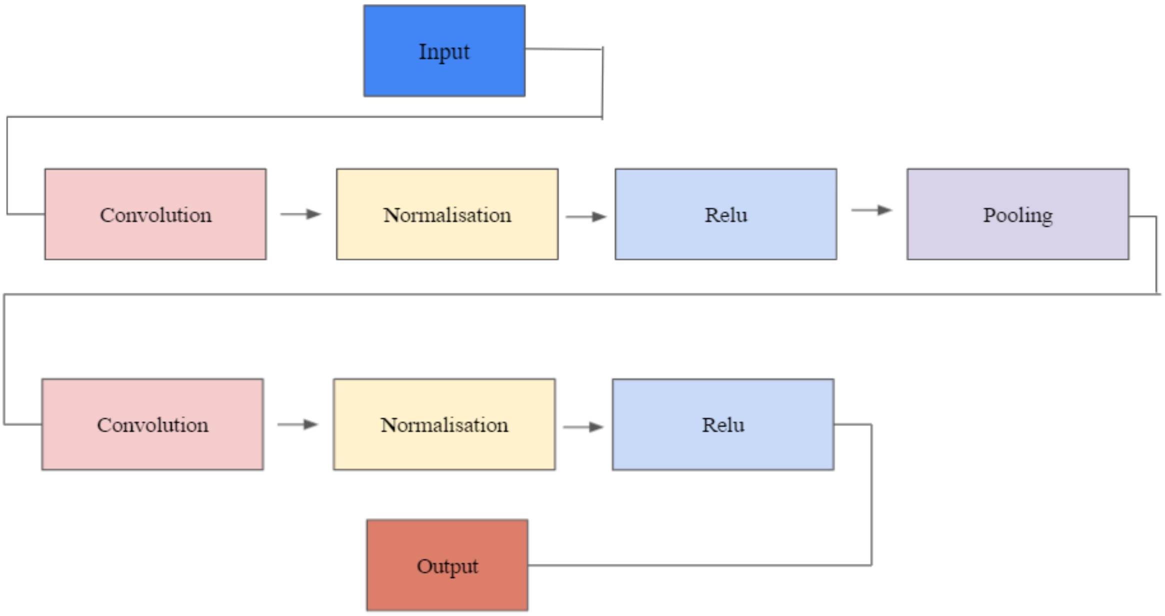


Fig 5: The architecture of the CNN used in the classification of buildings.

In convolution, a convolution operation is performed to the input and will be output to the next layer. Neurons in this layer receive input from only the previous layer. In the process, the models form a feature map or an activation map. It extracts different features from the image.

As in the flowchart, The model has 2 Convolution layers. We used only 2 layers because of the limited size of the dataset. we also added padding of 1 to narrowly convolute.

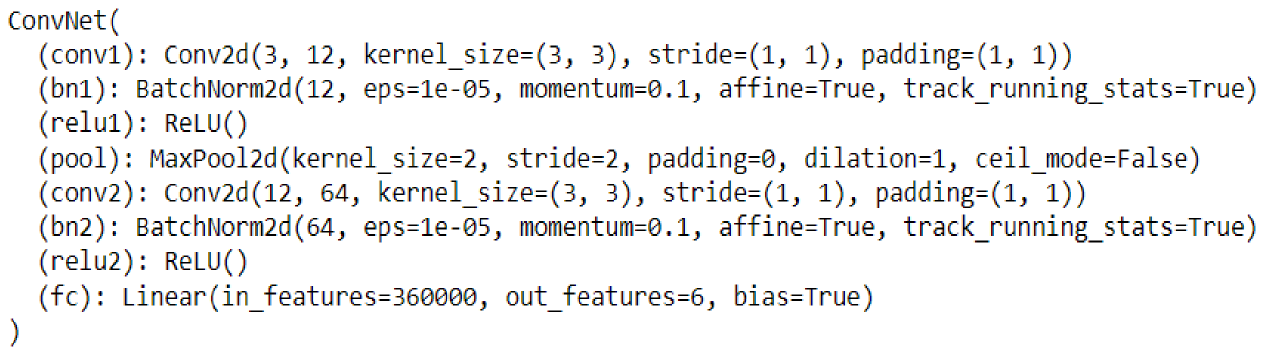
After the Convolution, we normalized the output of the convoluted output. This is done to standardize the output. Using batch normalization will enhance the learning and avoid overfitting of the model. In the model, we normalized the number of features of the previous layer.

The ReLU (rectified linear unit) layer is the next step done after normalisation.We applied an **activation function** onto your feature maps to increase non-linearity in the network. This is because images themselves are highly non-linear. This process will remove the negative values in the feature map by replacing them with zero.

The last step in this layer is pooling. Pooling will change the orientation of the image. This is done to increase the spatial variance. It also prevents the model from overfitting. In our model, images are just horizontally flipped because there will not be an inverted building.

After various trails, with increasing the layers and hypertuning the parameters, the best architecture we found was with 2 hidden layers. On increasing the hidden layers, the data is overfitting and the prediction is bad and on decreasing layers, the model is underfitting.

Now, after this step, the model classifies the image into one of the 6 classes based. It will check the accuracy on train and test data. And the process repeats for 15 times. The epoch count is set to 15 after so many tries. As the number increases, the model is starting to get overfit. The best model among the 15, i.e, the epoch with highest accuracy will be saved and will be used while prediction.



```
ConvNet(
  (conv1): Conv2d(3, 12, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
  (bn1): BatchNorm2d(12, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (relu1): ReLU()
  (pool): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
  (conv2): Conv2d(12, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
  (bn2): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (relu2): ReLU()
  (fc): Linear(in_features=360000, out_features=6, bias=True)
)
```

Fig 6: Example of the best model

VI. RESULTS AND DISCUSSION:

In our training experiments, we have come across different predictions based on different values of parameters. Few of the key observations are:

- The model training accuracy of the best model is <98% everytime and with comparison to testing accuracy, it's always half of the training accuracy, i.e, >50% .
- Increasing the count of images in a category is improving the accuracy of prediction.
- The classes in the model are so broad. Example: Religious buildings consist of Temples, churches and mosques. The model always classifies the temples as religious buildings but makes a wrong prediction when the image is a church or mosque. This is because of fewer images of those kinds. This holds true to other classes too.
- 99% of times, the slums are predicted correctly because it is the most unique class compared to the others
- The model is bad in classifying apartments, offices and shops because all three classes have similar images with glass on the frame of the building.

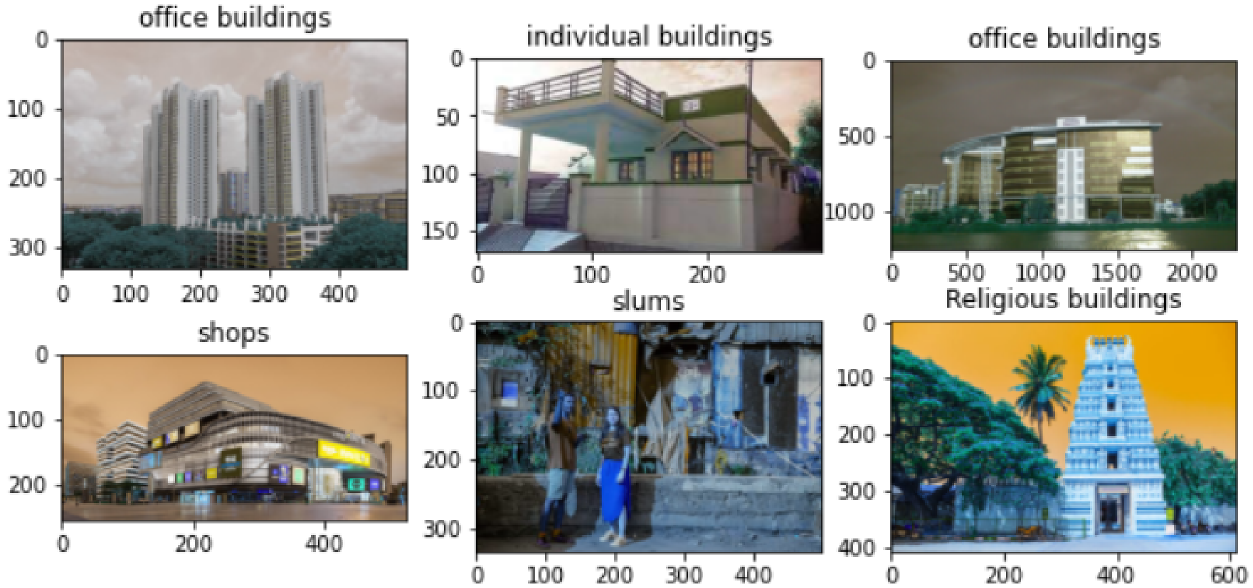


Fig 7: Few sample outputs

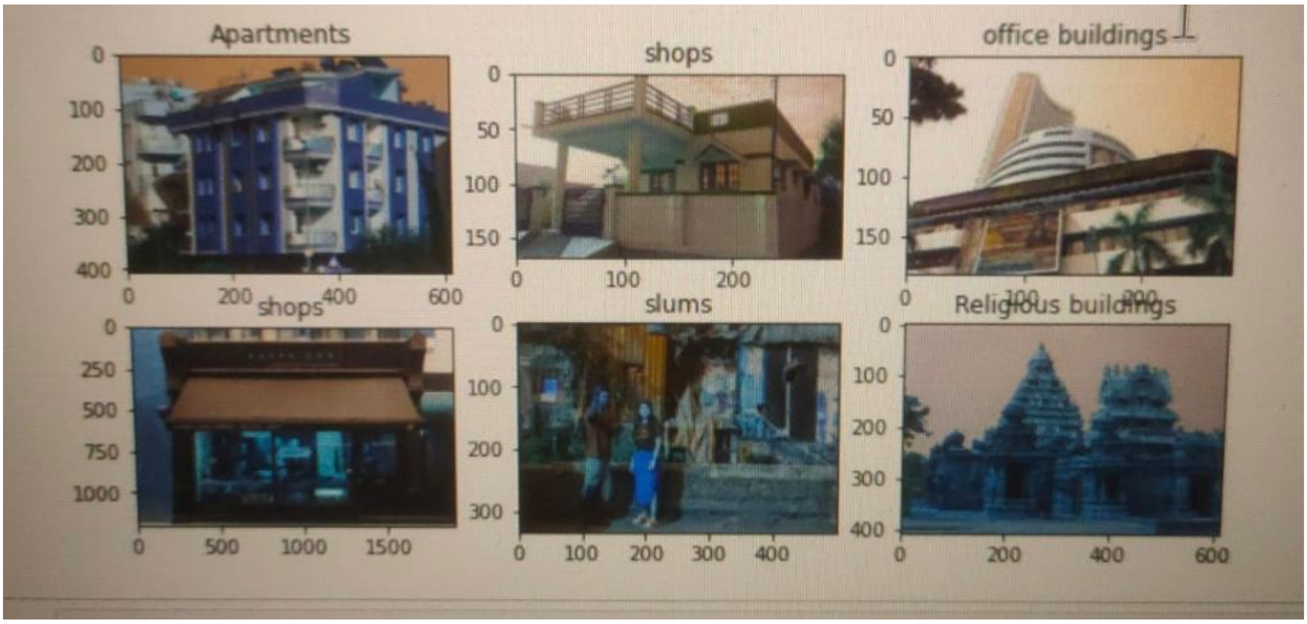


Fig 8: Results of the trained model

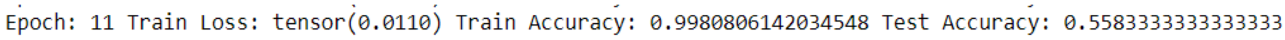

```
Epoch: 11 Train Loss: tensor(0.0110) Train Accuracy: 0.9980806142034548 Test Accuracy: 0.5583333333333333
```

Fig 9: Accuracy of best model

The proposed model is really doing well in classifying on an Indian dataset. On increasing the dataset and tuning the parameters, this model can be used various other sectors for development

VII. CONCLUSION

In this project, we have presented an ideology and a framework for building instance classification into six different categories. With this approach, land use classification can get a new dimension with which we can attain relatively higher accuracy in land usage planning and land usage efficiency planning.

We have also built a dataset of street view images with a total of 6 kinds of buildings, investigation, modification, and expansion of this dataset can lead to other important studies like a study of building design patterns, location identification based on building designs, climate estimation using building information and many more.

We have put a convolutional neural network to use, with which we have attained a prediction accuracy of around 60% in predicting the type of building using the structure and height data. Such information when tagged with geo-data can have the potential to boost innovation in urban planning, development of slums, city economic restructuring, and more such studies.

For future work, to improve the classification efficiency, other information can be incorporated along with the images like location data, surroundings, text descriptions, brand names, etc. Also to get density estimation geo-tagged data can be combined with remote sensing images of the area. We feel our study can be a base to support many other projects and studies upon it.

References:
[1]https://towardsdatascience.com/wtf-is-image-classification-8e78a8235acb
https://openaccess.thecvf.com/content_cvpr_2016/papers/Wang_CNN-RNN_A_Unified_CVPR_2016_paper.pdf

[2]https://reader.elsevier.com/reader/sd/pii/S0924271618300352?token=32424DC7DD3EDAE5085734712F950057D51590C256FCDB739B9A21007E0F238F927C1F7B3CBBBD12145D2E1D3D9E9E94&originRegion=eu-west-1&originCreation=20211122172l0

[3]Building Instance Classification Using Street View Images – arXiv Vanity (arxiv-vanity.com)