

Identification of Illegal Construction using Image Processing

Sathvik Bellary Kiran
Bangalore, India

Rithvik Mohan V.
Bangalore, India

ABSTRACT

Every major city in the world faces the problem of illegal construction—owners building extra floors or people encroaching on reserved land. Not only does this cause problems for the people living in the house but can also cause major problems to the environment, public safety and are a huge fire hazard. As of today, most of the checking for illegal constructions happens manually through site visits and inspection. Many of these defaulters get away with their acts for decades before it is discovered that the house, they are living in is not approved/is illegally constructed. Also, it is a major loss of money for government authorities.

This paper describes a method, using drones to capture aerial images and store them in a secure database. Once that is done, an algorithm is described that can comb through the millions of images that are present in the database and flag all those images that can be possible or positive illegal construction with the help of image processing using computer vision. The algorithm would flag the images which may have possible faults and present them to the supervisor who can then manually approve the flagged sites for further physical inspection or can deem them as having no faults. This paper only presents a method to reduce the time spent by the person who is manually checking each of the sites' images and automatically presents him/her only ones that have a high risk of faulting them. This will not only significantly reduce the time spent on checking but can also save the authorities lots of money and help them to tackle more such defaulters. This paper describes a method, using drones to capture aerial images and store them in a secure database. Once that is done, an algorithm is described that can comb through the millions of images that are present in the database and flag all those images that can be possible or positive illegal construction with the help of image processing using computer vision. The algorithm would flag the images which may have possible faults and present them to the supervisor who can then manually approve the flagged sites for further physical inspection or can deem them as having no faults. This paper only presents a method to reduce the time spent by the person who is manually checking each of the sites' images and automatically presents him/her only ones that have a high risk of faulting them. This will not only significantly reduce the time spent on checking but can also save the authorities lots of money and help them to tackle more such defaulters.

General Terms

SSIM (Structural Similarity Index), OpenCV, Imutils, Greyscaling, Thresholding, Contours

Keywords

Illegal construction, Drones, Image Processing, Computer Vision

1. INTRODUCTION

India faces a menace of illegal construction which has developed as an economical and social problem. Illegal construction poses a threat to public safety and causes problems in city planning and utilization of resources.

Much research has already been done in the field of illegal-construction detection with the help of bird's eye view images or through SITS (Satellite Image Time Series). In this paper, a method for monitoring the illegal constructions with the help of aerial images shot with drones is proposed. This method will help in reducing the time taken in the tedious process of manual checking by identifying potential breaches.

2. BACKGROUND

2.1 What is Illegal Construction

Illegal construction (also known as illegal building or illegal housing) is a construction that has been built without a valid permit. Illegal-constructions are a potential technical hazard on uncontrolled sites or finished buildings. They can also be a major environmental threat when the works encroach upon preserve areas like nature reserves.

Illegal buildings are also consequences of urbanization, overpopulation, homelessness and poverty in many countries. It can be seen in expanding slums and shantytowns. In most cases, illegal building activity may be due to the profitable expansion of existing buildings for exploiting the property value. This is seen when owners of buildings expand existing structures by constructing additional floors or likewise.

There can be various effects on the environment, such as landslides and deforestation. These can occur when the construction encroaches on reserved land. Moreover, illegal constructions pose a high risk to the urban population. The risk of fire accidents is extremely high in such buildings. Also sinking of the buildings (due to additional weight added on the structure/ weak soil), collapsing of buildings have been observed in such cases [1].

2.2 Forms of Illegal Construction

There are five main types of illegal construction [2]:

1. Violation of plan, the plan approved by the government authority is not followed during construction.
2. Illegal land occupation: Vacant land is occupied and constructed by people without prior notice.
3. Improper construction: The building that has not obtained construction license, or constructed outside the license.
4. Construction without permission: Extra floor/room is constructed, which is not present in the approved plan.

- Temporary building that is not demolished beyond the approved period of time for their uses.

2.3 Current Monitoring method

Currently, satellite mapping is being used on various cities (170 cities) but, in other places, checking is done manually. [3]

Some issues with satellite mapping:

- Satellites are expensive to develop and deploy for specific purposes.
- The images taken by the satellite may not have great resolution when compared to a drone. The resolutions of the image cannot be changed.
- It may be difficult to take images of a particular area at a particular time of day since a lot of technicalities being involved in order to capture the image.

To overcome these challenges, a method is proposed using images captured by drones and analyzing these images using image recognition algorithms combined with computer vision techniques. Anomalies are detected in the captured images by referencing previously captured pictures of the same place and hence identifying the possible illegal construction spots.

3. METHOD

3.1 Capturing Images

In order to capture appropriate images of the property, the following criteria should be met:

- There should be favourable weather conditions (there should not be strong wind).
- There should be clear and bright conditions for the image to be processed clearly.
- The image must be captured at least 20 feet above the roof of the building.
- The image quality should be at least 8-10 megapixels in resolution.
- An experienced drone pilot must be appointed in order to fly the drones safely.

Each image must be marked with the property number and stored in a database with a structure similar to Table 1 mentioned ahead in Section 4.4 of this paper. The database size can be decided based on the number of images to be stored.

The flight path of the drone can be recorded in order to set an automatic flight path for the upcoming inspections (if possible, i.e. if hardware permits). The next set of images should be taken after a set time interval along with the same conditions as the previous image.

Appropriate permissions should be taken from the local law enforcement authorities if there are restrictions to fly drones.

4. ALGORITHM

4.1 SSIM (Structural Similarity Index)

SSIM or Structural Similarity Index is a metric designed to predict and analyze the quality of images mostly in television screens and other image sources. Another major use of SSIM has been to predict the similarity between two different images.

Here SSIM is used to predict the similarities and spot major differences between the images in the database.

The SSIM for an image is calculated on various windows/partitions of an image. The measure between two windows x and y of common size $N \times N$ is: [4]

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}$$

4.2 Software Tools

In order to compute the difference between the two captured images of the plot, SSIM is used as a metric. This is implemented using the scikit-image library for image processing. The algorithm makes use of python scikit-image, OpenCV and imutils.

- OpenCV: OpenCV is a library developed by Intel which is aimed at providing advanced and real time computer vision [5]
- Imutils: Imutils is a library of python that has various functions for basic manipulation of images (Example: rotation, gray-scaling, resizing, etc)
- Scikit-image: Scikit-image is an open-source image library of python that contains algorithms for various operations on images such as geometric transformations, morphology, segmentation and many more.[6]

4.3 Greyscaling the Images

RGB images contain a lot of data some of which is not required. The components, i.e. R, G, B, image holds different intensity labels. It is represented by 3 channels, each which consists of 8-bits [9]. Therefore, for a color image, there will be intensities for each scale. Hence, lots of data would be stored. In order to reduce the amount of data to be stored, the images are greyscaled, which consists of only 2 channels. The function `cv2.cvtColor(image, flag)` is used where the flag specifies the type of conversion. In order to greyscale the image, we set the flag to `cv2.COLOR_BGR2GRAY` [5]. Once the images are greyscaled they are stored in the database (table 2 specified in the next part).

4.4 Database Structure

There are two tables used, with the following structures:

Table 1. Index table (db1)

Index	Plot No	Address	Area	Owner Name

2015	2016	2017	2018	2019	2020	...

Table 1 consists of the following attributes:

- Plot No - plot number of the property captured
- Address – complete address of the property
- Area – area in which the property resides

4. Owner name – the name of the owners of the respective properties
5. 2015 – images captured in the year 2015
6. 2016 – images captured in the year 2016
7. And so on for the upcoming years.

Table 2. Current year’s table (db2)

Index	Latest Image	Old Image	SSIM (int)	Difference Image

Threshold	Contours	Final New Image	Final Old Image

Table 2 consists of the following attributes:

1. Latest image – most recent image captured of the respective property.
2. Old image – previously captured image of the respective property
3. SSIM – SSIM score calculated by the algorithm
4. Difference Image – image obtained after
5. Threshold – contains the binarized image.
6. Contours – contains the coordinates of the identified area.
7. Final New Image – Image with highlighted rectangles (if any)
8. Final Old Image – Image before processing

4.5 Computing Image Difference

4.5.1 Preprocessing

The first step is to find the SSIM between the two images. To accomplish this, a program is created to compute the SSIM and compare all images stored in the database side by side (with the image taken on the previous inspection).

The two images to be compared are converted to grey-scale and loaded to the program. The program created takes two arguments (i.e. two images to be compared), and then the SSIM is computed.

This is done using Scikit-image's inbuilt function called "compare_ssim()". A score and a difference image are returned by this function. The score is the SSIM index between [-1,1] (where 1 is a perfect match) and the difference image contains the actual differences between the two images. It has a range [0,1]. To visualize this, the array is converted to 8-bit unsigned integers (range [0,255]) to further process using OpenCV. [7]

After this process, the compare_ssim() function returns two values, i.e. the difference image and the score.

4.5.2 Thresholding

The next step is to threshold our difference image. Basically, in digital image processing, thresholding is defined as a process that converts a grayscale digital image into a simple binary image. It is also one of the simplest methods of

segmenting images.

To threshold the image, OpenCV's function cv2.threshold() is used. This function is applied to the difference image with two options specifying the type of thresholding. Here, two thresholding settings are used - cv2.THRESH_BINARY_INV and cv2.THRESH_OTSU:

- cv2.THRESH_OTSU: Otsu's binarization method of image thresholding returns a single intensity threshold with two separate pixel classes, i.e. foreground and background. The algorithm iterates through all possible threshold values of the grey-scale image and calculates the measure of spread. It then balances out the weights by maximizing the variance between the foreground and background class pixels.[8]



Original image Otsu’s binarized image

(Image Courtesy: [Pikez33 - http://en.wikipedia.org/wiki/File:Image_processing_post_otsus_algorithm.jpg#file](http://en.wikipedia.org/wiki/File:Image_processing_post_otsus_algorithm.jpg#file))

- cv2.THRESH_BINARY_INV: This is inverse binary thresholding. Here, the algorithm works the opposite of normal binarization where the destination pixel is set to 0 if the corresponding pixel is greater than the threshold and maximum value. [5]

4.5.3 Finding Contours

The next step is to use the thresholded image and find the contours in it. For this Open.CV.findcontours() is used. Contour is basically a curve joining all the points along the boundary having the same intensity or colour. The function Open.CV.findcontours() stores the various contours in a list. Markers can now be drawn to highlight the difference points (illegal constructions).[6]

Bounding rectangles are added around the contour by using CV2.boundingRect(). The (x,y) coordinates returned by this is then passed to the drawing function CV2.rectangle() to draw a red rectangle around the contour.

4.6 Comparing the Images

The two images can now be visualized side by side. Red rectangles highlight the various differences (if any) in the comparing image to show the differences in the images. This whole process is applied to all the pairs of images stored in the database.

4.7 Identifying Potential Breachers

Once the database is loaded with the respective images (old and new), the software is made to run on all sets of images. Once run, it gives an SSIM score from which the various breaches/deviations are identified. The breaches/deviations are marked by a red rectangle where the software identifies the breaches itself. It can then selectively present high-risk plots/constructions where the threat seems much more than the allowed limit. The concerned authority can set the limit by changing the SSIM score. The respective agent/person who is

checking the images can designate the highlighted plots for further physical inspection after observing the results.

4.8 Pseudo Code

The below pseudo-code describes the working of the algorithm. Two tables are used in the database- Table1 and Table2. The first one with each image of ever year (i.e. Table1) and one for working this year's anomalies, also called working table (i.e. Table2).

```

get the current_year and previous_year

for both current_year and previous_year in db1:

    db2.NewImage= db1.current_year

    db2.OldImage= db1.previous_year

for each NewImage and OldImage in db2:

    OldImage=cv2.cvtColor(OldImage,bgr2gray)

    NewImage=cv2.cvtColor(NewImage,bgr2gray)

    (ssimvalue,differenceimage)=compare_ssim(new
    image, old image, full=true)

    db2.ssimscore=ssimvalue

    db2.differenceimage= differenceimage

for each differenceimage in db2.differenceimage:

    differenceimage=(differenceimage*255).astype("uin
    t8") //convert each differenceimage into 8 bit
    unsigned integers,range[0-255]

for each differenceimage in db2.differenceimage:

    thresholdimage=cv2.threshold(differenceimage,0,25
    5,cv2.THRESH_BINARY_INV|

    cv2.THRESH_OTSU)[1]//Binarizing the difference
    image

    db2.thresholdimage=thresholdimage

for each thresholdimage in db2.thresholdimage:

    contours=
    imutils.grab_contours(cv2.findContours(thresholdi
    mage.copy(),cv2.RETR_EXTERNAL,cv2.CHAIN_
    APPROX_SIMPLE))

    db2.contours=contours

//bounding rectangles

for each contour in db2.contours:

    for c in contour:

        x,y,width,height= cv2.boundingRect(c)

        db2.FinalNewImage=cv2.rectangle(db2.NewImage,
        (x,y),(x+width,y+height),(0,0,255),2)

        db2.FinalOldImage=cv2.rectangle(db2.OldImage,(x

```

```

,y),(x+width,y+height),(0,0,255),2)

```

```

get ssim_limit from user

```

```

for each ssimvalue in db2.ssimscore:

```

```

    ssimvalue >= ssim_limit:

```

```

        show(db2.FinalNewImage)

```

```

        show(db2.FinalOldimage)





```

Now the supervisor can approve the plots for further physical inspection if he/she deems it necessary after viewing the old final image and the new final image presented.

5. EXPERIMENTAL DATA

After running the algorithm on a few images. The following data was obtained:

Table 3. Experimental Results

Original Image	Final Image	SSIM
		0.997
		0.945

Both the images in each row show the algorithm detecting additional construction, which has been highlighted by the red, rectangular box. This feature will help the supervisor in selecting potential defaulters in an easy and faster way.

6. CONCLUSION

In conclusion, the algorithm presented above can be used to identify illegal construction using image processing where the algorithm will skim through the images present in the database and will flag the images which appear to have inconsistencies. Further, it is the decision of the supervisor on whether to inspect the plot of the flagged image or not.

There are a lot of places where the accuracy of the algorithm can be improved by helping it identify specific things in the image. This may help remove the possibility of having false - positives for flagging. If this process is employed, it can limit the amount of time spent by government officials and also help save a lot of money.

7. REFERENCES

- [1] D. Di Martire , M. De Rosa , V. Pesce , M. A. Santangelo , and D. Calcaterra, "Landslide hazard and land management in high-density urban areas of Campania region, Italy", Nat. Hazards Earth Syst. Sci.,

- 12, 905–926, 2012
- [2] SU Guohua, “Research on legal system of handling illegal building in China,” Southwest University of Political Science and Law, 2008.
- [3] “Satellite mapping is being used to monitor illegal construction in 170 cities: Govt to HC ” Mumbai Mirror , October 5 2018, <https://mumbaimirror.indiatimes.com/mumbai/other/satellite-mapping-is-being-used-to-monitor-illegal-construction-in-170-cities-govt-to-hc/articleshow/66078824.cms>
- [4] Zhou Wang,” Image Quality Assessment: From Error Visibility to Structural Similarity”, IEEE TRANSACTIONS ON IMAGE PROCESSING, VOL. 13, NO. 4, APRIL 2004
- [5] “docs.opencv.org”- OpenCV Online Official documentation(OpenCV 2.x API, Version 4.5.0)
- [6] “pypi.org/projects/imutils”- imutils version-0.5.3 Documentation- Author- Adrian Rosebrock
- [7] “scikit-image.org”- Sci-kit image Online Official documentation (Version 0.17.2)
- [8] Nobuyuki Otsu (1979). "A threshold selection method from gray-level histograms". IEEE Trans. Sys. Man. Cyber. **9** (1): 62–66. doi:10.1109/TSMC.1979.4310076
- [9] Wilhelm Burger, Mark J. Burge (2010). Principles of Digital Image Processing Core Algorithms. Springer Science & Business Media. pp. 110–111. ISBN 978-1-84800-195-4.