**PROJECT TITLE**

**DETECTING BUILDING DEFECTS USING CONVOLUTION NEURAL NETWORKS**

Developed by:

KATKURI SRIJA;

GOUDA KAVYA;

CHEEKATI KRISHNACHAITHANYA;

BURRA ARUNKUMAR,

# INTRODUCTION

## OVERVIEW:

This project deals with detecting the defects and spalls in buildings using Convolutional neural networks. With the help of this project the defects in buildings can be detected conveniently. This focuses on efficient building maintainence with reduced time and labour.

### PURPOSE:

The major purpose of this project is set to investigate the novel application of deep learning method of convolutional neural networks (CNN) in automating the condition assessment of buildings. The focus is to automated detection and localisation of key defects arising from dampness in buildings from images. However, as the first attempt to tackle the problem, this project applies a number of limitations.

Firstly, multiple types of the defects are not considered at once. This means that the images considered by the model belong to only one category. Secondly, only the images with visible defects are considered. Thirdly, consideration of the extreme lighting and orientation, e.g., low lighting, too bright images are not included in this project.

### PROPOSED SOLUTION

This era requires smart buildings. This proposed system is based on pre-trained

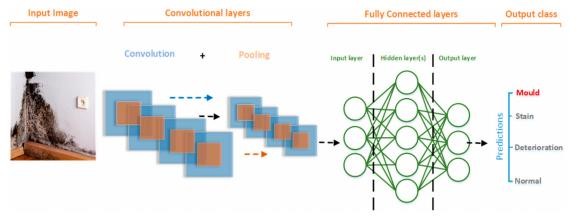
CNN classiﬁer of VGG-16, with class activation mapping (CAM) for object localisation.

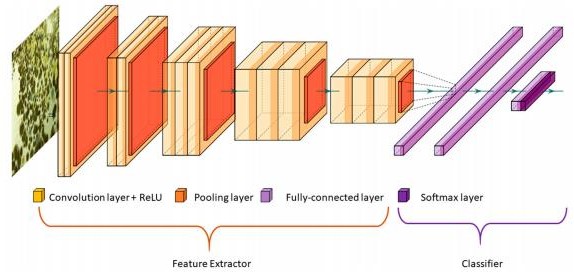
The challenges and limitations of the model in real-life applications have been

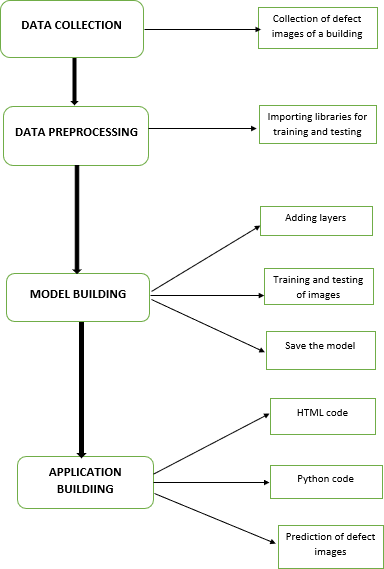
identiﬁed. The proposed system has proven to be robust and able to accurately detect and localise building defects. This approach is being developed with the potential to scale-up and further advance to support automated detection of defects and deterioration of buildings in real-time using mobile devices and drones. Where technology takes over manual work of building maintenance. This projects builds an application where it detects cracks and spalls in the buildings and helps in building maintenance.

# THEORETICAL ANALYSIS:

### BLOCK DIAGRAM:







### HARWARE/SOFTWARE DESIGN:

#### Hardware Design:

* + - Hard Disk
    - Laptop
    - RAM-4GB

#### Sofware Design:

* + - Python 3.6
    - Jupyternotebook
    - Spyder ide

# EXPERIMENTAL INVESTIGATIONS:

## DATA COLLECTION

We have taken images of diﬀerent resolutions and sizes which were obtained from the internet. The data was labelled into three main categories. The total number of images used as training data was 257. The remaining 133 images





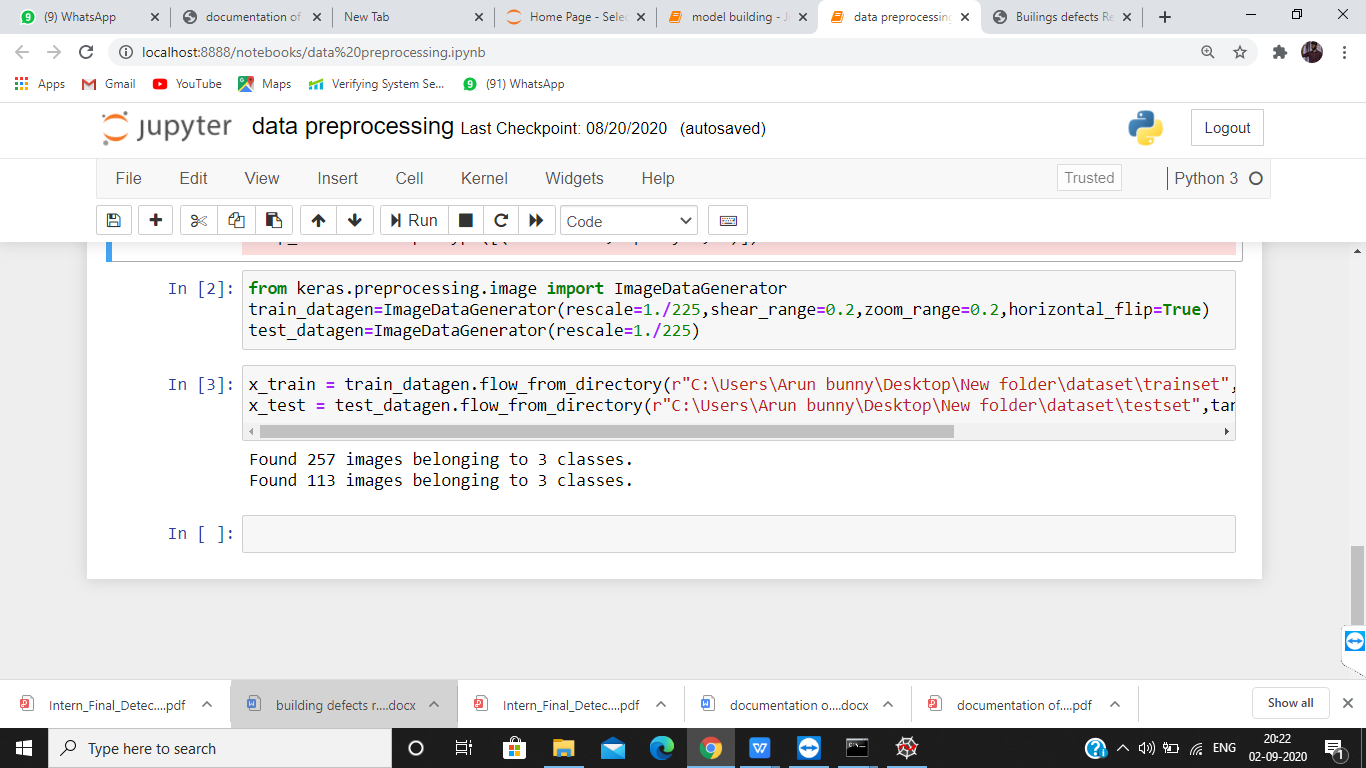
### DATA PREPROCESSING:

from keras.preprocessing.image import ImageDataGenerator

train\_datagen=ImageDataGenerator(rescale=1./225,shear\_range=0.2,zoom\_range=0.2,horizontal\_flip=True)

test\_datagen=ImageDataGenerator(rescale=1./225)

Here we are importing the ImageData generator to train and test the images dataset.



### MODEL BUILDING:from keras.models import Sequential from keras.layers import Dense

from keras.models import Sequential

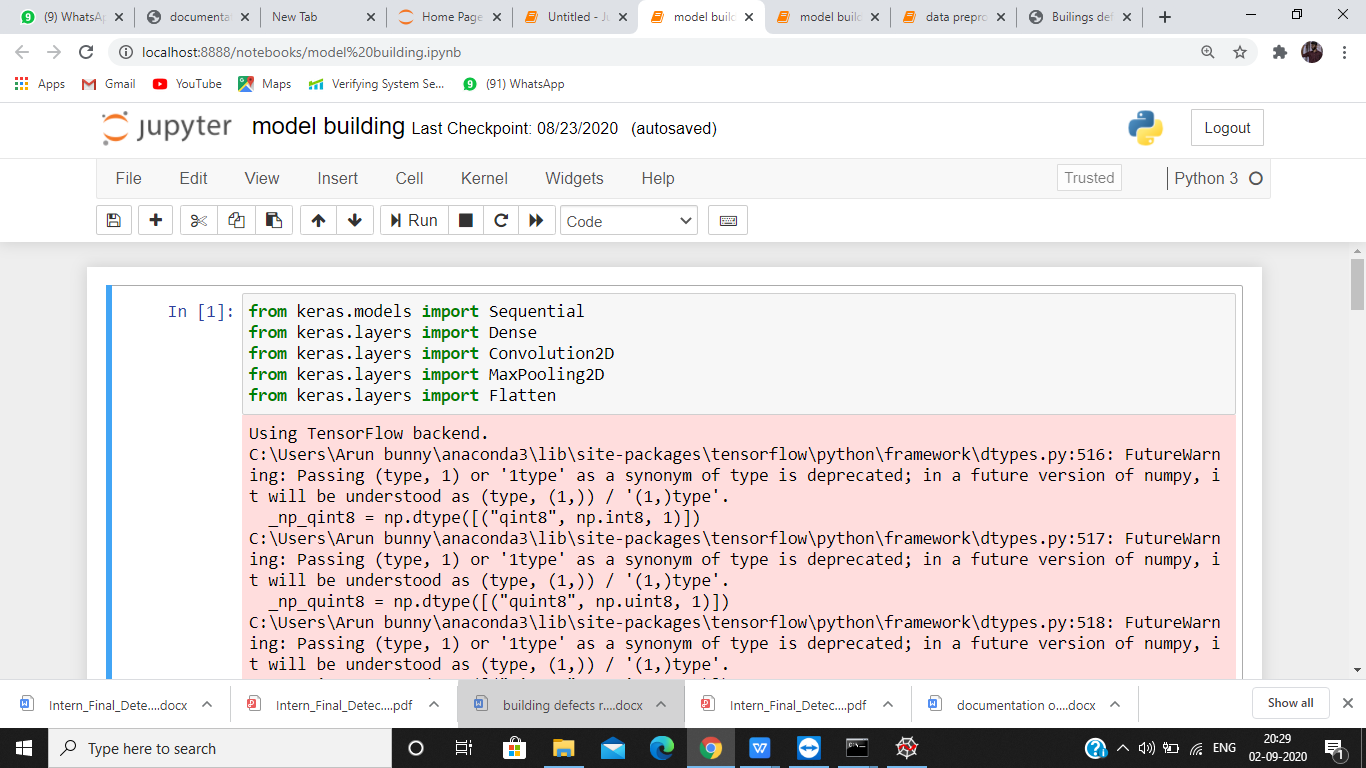
from keras.layers import Dense

from keras.layers import Convolution2D

from keras.layers import MaxPooling2D

from keras.layers import Flatten

The layers are imported along with the activation function for the training and test validation of the images.



# Screenshot (77)

# Screenshot (78)

# Screenshot (79)

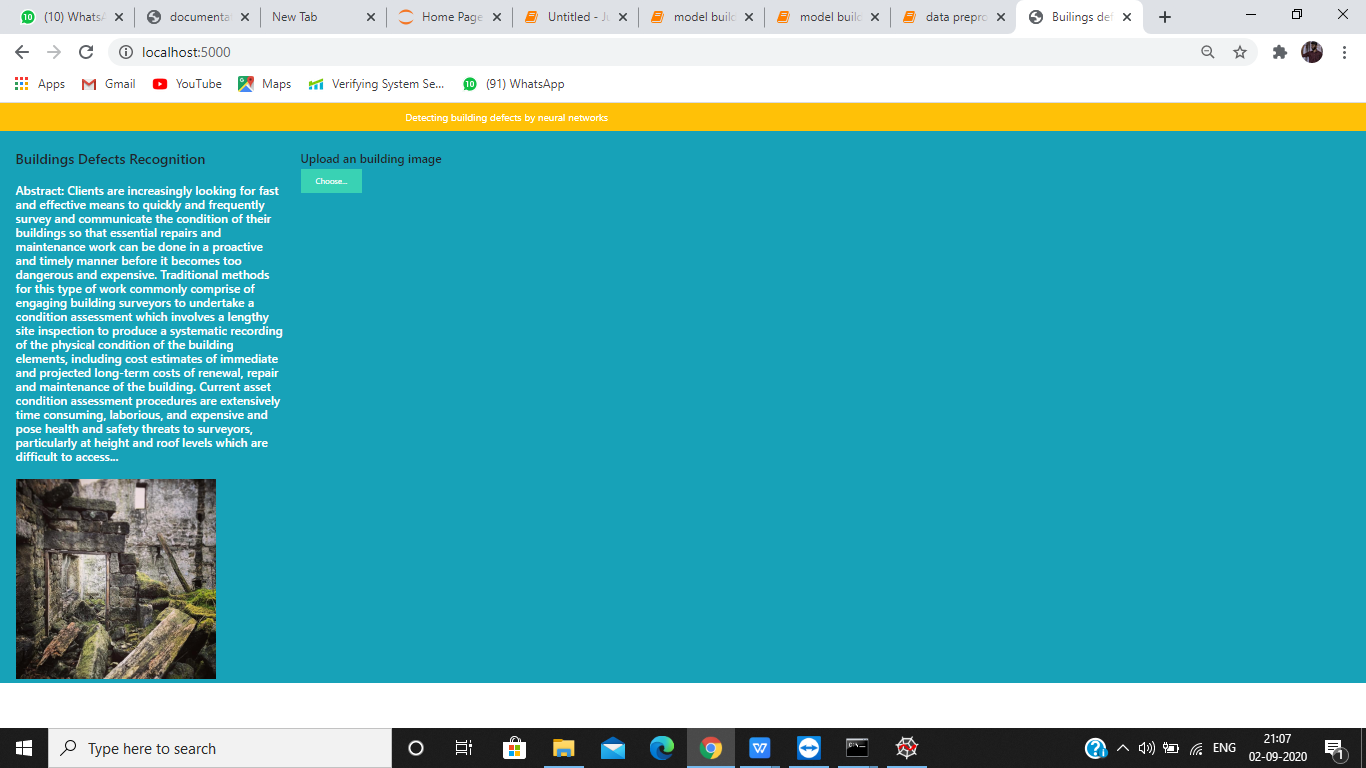
# Screenshot (80)

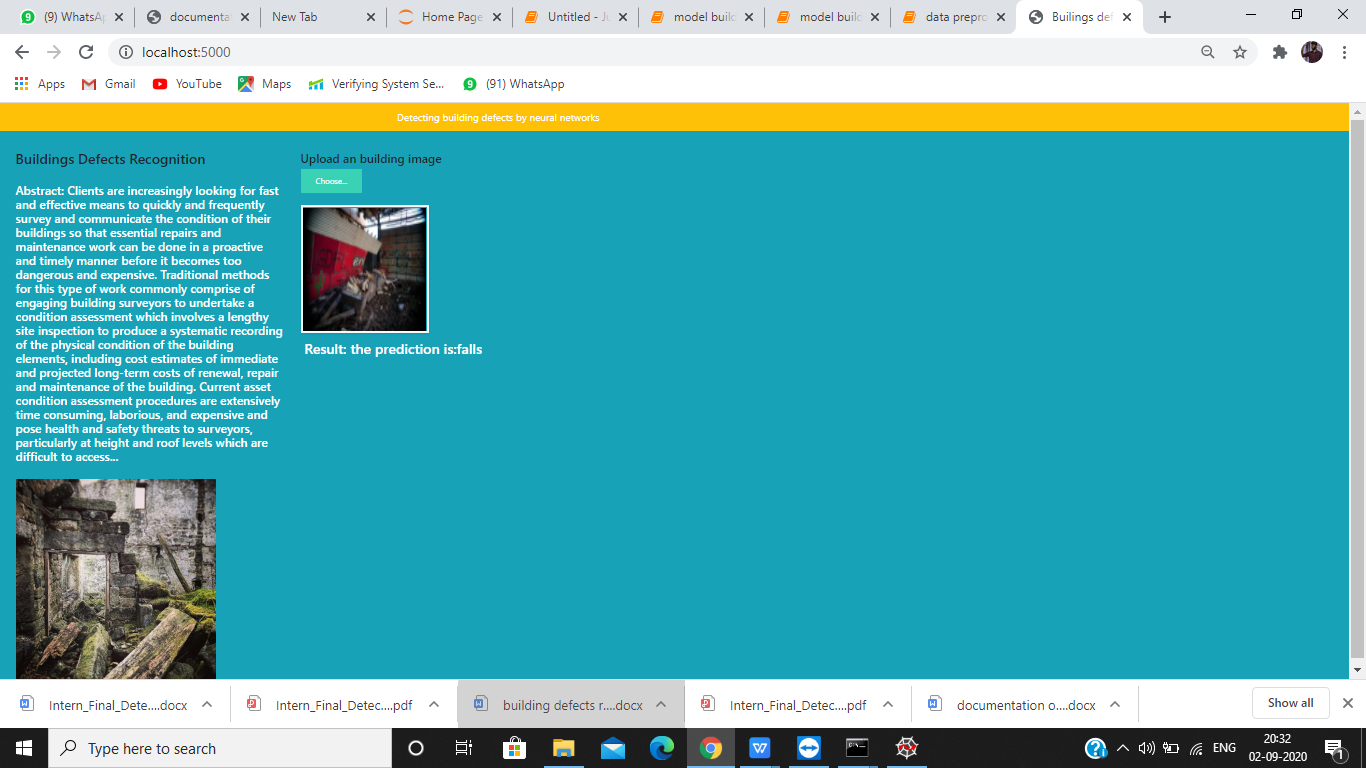
# Screenshot (81)

**RESULT:**

CNN algorithm is used to recognize the image uploaded and predict the result.

The css used here is internal css and spyder ide has been used.





# ADVANTAGES AND DISADVANTAGES

#### ADVANTAGES:

* 1. The main advantage is that it automatically detects the important features without any human supervision.
  2. Classification of the defect can be predicted.
  3. High reliability and robustness in classifying and localising the defects. 4.Man power for defecting defect is reduced5.Frequent monerating of any type of defect in the building is reduced.

#### DISADVANTAGES:

1.There may be slit chances of error while predicting the images 2.Detection is difficult when there so many similar images 3.Classification of images is similar so it is difficult to detect

# APPLICATIONS:

* 1. It can be used in detecting the defect in the house. 2.It is very useful in buildings,companies.

1. It is customizable by any type of person.
2. Building can be virtually projected and spots can be defected.
3. The application can be modified for portability in such a way that smart phone cameras can be used to detect the defects or modified for drones for a complete survelience for a defects

## CONCLUSION:

The work is concerned with the development of a deep learning-based method for the automated detection and localisation of key building defects from given images. This project is part of work on condition assessment of built assets.

# FUTURE SCOPE:

Future scope is simply focus only on detecting cracks on concrete surfaces which is a simple binary classiﬁcation problem, we oﬀer a method to build a powerful model that can accurately detect and classify multi-class defects given are latively very small datasets. In the future works, these limitations will be considered to be able to get closer to the concept of a fully automated detection. Through fully satisfying these challenges and limitations, our present work will be evolved into a software application to perform real-time detection of defects using vision sensors including drones. The work will also be extended to cover other models that can detect other defects in construction such as cracks, structural movements, spalling and corrosion. Our long- term vision includes plans to create a large, open source database of diﬀerent building and construction defects which will support world-wide research on condition assessment of built assets.

**APPENDIX:**

**Python code:**

from flask import Flask, render\_template,request

import os

from keras.preprocessing import image

from werkzeug.utils import secure\_filename

from keras.models import load\_model

import tensorflow as tf

global graph

graph = tf.get\_default\_graph()

import numpy as np

model = load\_model("defect.h5")

app = Flask(\_\_name\_\_)

@app.route('/')

def index():

return render\_template("base.html", methods = ['GET'])

@app.route('/predict',methods = ['GET','POST'])

def pred():

if request.method == "POST":

f = request.files["image"]

print("hie")

"""take the path of the current

running prog,and concatenate to the

folder where you wouldlike to save the file"""

basepath = os.path.dirname(\_\_file\_\_)

print(basepath)

file\_path = os.path.join(basepath,"uploads",secure\_filename(f.filename))

print(file\_path)

f.save(file\_path)

img = image.load\_img(file\_path,target\_size = (64,64))

x = image.img\_to\_array(img)

x = np.expand\_dims(x,axis = 0)

with graph.as\_default():

p = model.predict\_classes(x)

print(p)

index = ["Defect in Drainage System and water sewage","cracks","falls"]

text = "the prediction is:" +index[p[0]]

return text

if \_\_name\_\_ == "\_\_main\_\_":

app.run(debug = True)

**HTML code:**

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<meta http-equiv="X-UA-Compatible" content="ie=edge">

<title>Builings defects Recognition System</title>

<link href="https://cdn.bootcss.com/bootstrap/4.0.0/css/bootstrap.min.css" rel="stylesheet">

<script src="https://cdn.bootcss.com/popper.js/1.12.9/umd/popper.min.js"></script>

<script src="https://cdn.bootcss.com/jquery/3.3.1/jquery.min.js"></script>

<script src="https://cdn.bootcss.com/bootstrap/4.0.0/js/bootstrap.min.js"></script>

<link href="{{ url\_for('static', filename='css/main.css') }}" rel="stylesheet">

<style>

.bg-dark {

background-color:blue;

}

result {

color: #0a1c4ed1;

}

input[type='file'] {

color:blue;

}

</style>

</head>

<body>

<nav class="navbar navbar-dark bg-warning">

<div class="container">

<a class="navbar-brand" href="#">Detecting building defects by neural networks</a>

</div>

</nav>

<div class="navbar navbar-dark bg-info">

<div id="content" style="margin-top:2em">

<div class="container">

<div class="row">

<div class="col-sm-6 bd" >

<h3>Buildings Defects Recognition </h3>

<br>

<h4><font color="white">Abstract: Clients are increasingly looking for fast and effective means to quickly and frequently survey and communicate the condition of their buildings so that essential repairs and maintenance work can be done in a proactive and timely manner before it becomes too dangerous and expensive. Traditional methods for this type of work commonly comprise of engaging building surveyors to undertake a condition assessment which involves a lengthy site inspection to produce a systematic recording of the physical condition of the building elements, including cost estimates of immediate and projected long-term costs of renewal, repair and maintenance of the building. Current asset condition assessment procedures are extensively time consuming, laborious, and expensive and pose health and safety threats to surveyors, particularly at height and roof levels which are difficult to access...</font></h4>

<br>

<img src="https://thumbs.dreamstime.com/b/derelict-building-falling-apart-seems-abandoned-building-falling-apart-104213329.jpg" style="height:400px"class="img-rounded" alt="Gesture">

</div>

<div class="col-sm-6">

<div>

<h4> Upload an building image</h4>

<form action = "http://localhost:5000/predict" id="upload-file" method="post" enctype="multipart/form-data">

<input type="file" name="file" id="file" accept="image/\*" onchange="previewImage();">

<label for="imageUpload" class="upload-label">

Choose...

</label>

<input type="file" name="image" id="imageUpload" accept=".png, .jpg, .jpeg">

</form>

<div class="image-section" style="display:none;">

<div class="img-preview">

<div id="imagePreview">

<img id="uploadedImage" style="border:"none" />

</div>

</div>

<div>

<button type="button" class="btn btn-secondery btn-lg " id="btn-predict">submit</button>

</div>

</div>

<div class="loader" style="display:none;"></div>

<h3>

​

<font color="white"><span id="result"> </span></font>

</h3>

</div>

</div>

</div>

</div>

</div>

</div>

</body>

<footer>

<script src="{{ url\_for('static', filename='js/main.js') }}" type="text/javascript"></script>

<script>

function previewImage() {

var file = document.getElementById("file").files;

if (file.length > 0) {

var fileReader = new FileReader();

fileReader.onload = function (event) {

document.getElementById("uploadedImage").setAttribute("src", event.target.result)

document.querySelector(".image-section").style.display = "block"

};

fileReader.readAsDataURL(file[0]);

}

}

</script>

</footer>

</html>