DETECTING BUILDING DEFECTS

Using Convolutional Neural Network

Developed by: Shreya Karan, Namitha GN, Aishwarya Kalal, Ramya R Shet, Meghana G

Smart Bridge-Remote Summer Internship Program

1. INTRODUCTION

The increasing adverse effects of climate change, demanding legal and regulatory requirements for sustainability, safety and well-being, and increasing competitiveness are forcing Clients to look for fast and effective means to quickly and frequently survey and communicate the condition of their buildings so that essential maintenance and repairs 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 resulting in a systematic recording of the physical conditions of the building elements with the use of photographs, note taking, drawings and information provided by the client. The data collected are then analysed to produce a report that includes a summary of the condition of the building and its elements. This is also used to produce estimates of immediate and projected long-term costs of renewal, repair and maintenance of the building. This enables facility managers to address current operational requirements, while also improving their real estate portfolio renewal forecasting and addressing funding for capital projects. 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 for inspection.

Image analysis techniques for detecting defects have been proposed as an alternative to the manual on-site inspection methods. Whilst the latter is time-consuming and not suitable for quantitative analysis, image analysis-based detection techniques, on the other hand, can be quite challenging and fully dependent on the quality of images taken under different real-world situations (e.g. light, shadow, noise, etc.).

The notable efforts include; structural health monitoring with Bayesian method, surface crack estimation using Gaussian regression, support vector machines (SVM), and neural networks, SVM for wall defects recognition, crack-detection on concrete surfaces using deep belief networks (DBN), crack detection in oak flooring using ensemble methods of random forests (RF), deterioration assessment using fuzzy logic, defect detection of ashlar masonry walls using logistic regression.

The automated detection of defects in earthquake damaged structures has also received considerable attention amongst researchers in recent years. Considering these few studies, far too little attention has been paid to the application of advanced machine learning methods and deep learning methods in the advancements of smart sensors for the building defects detection.

The major objective of this project is an 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, patches, stains, cracks in buildings from images.

1.1 OVERVIEW

Detection of defects including cracks and spalls on wall surface in high-rise buildings is a crucial task of buildings’ maintenance. 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. If left undetected and untreated, these defects can significantly affect the structural integrityand the aesthetic aspect of buildings, timely and cost-effective methods of building condition survey are of practicing need for the building owners and maintenance agencies to replace the time- and labour-consuming approach of manual survey.

1.2 Purpose

Our aim from the project is to make use of numpy & ImageDataGenerator libraries from python to extract the libraries for machine learning for predicting building defects. And learning how Deep learning method of convolutional neural networks (CNN) can be used in automating the condition assessment of buildings.

Our model aims in choosing the random image of a building and detect whether the building is defected or undefected.

2. LITERATURE SURVEY

Data mining is the process of analyzing data from different perspectives and extracting useful knowledge from it. It is the core of knowledge discovery process. The various steps involved in extracting knowledge from raw data as depicted in figure-1. Different data mining techniques include classification, clustering, association rule mining, prediction and sequential patterns, neural networks, regression etc. Classification is the most commonly applied data mining technique, which employs a set of pre-classified examples to develop a model that can classify the population of records at large. Fraud detection and credit risk applications are particularly well suited to classification technique. This approach frequently employs Decision tree based classification Algorithm. In classification, a training set is used to build the model as the classifier which can classify the data items into its appropriate classes. A test set is used to validate the model.

2.1 EXISTING PROBLEM

There are number of models devoted to the detection of defects in infrastructural assets such as cracks in road surfaces, bridges, dams, and sewerage pipelines and buildings. The automated detection of defects in earthquake damaged structures has also received considerable attention amongst clients in recent years. Considering these few studies, far too little attention has been paid to the application of advanced machine learning methods and deep learning by using convolutional neural network in the advancements of smart sensors for the building defects detection. There is a general lack of research in the automated condition assessment of buildings; despite they represent a significant financial asset class.

The previous models have high time complexity and space complexity whereas this model is constrained with the lot of advantages and with a higher accuracy than any other model already proposed. There is a user-friendly user interface to check loan score for the people who are about get the loan, and lot of the previous models haven't included the UI (User interface) which is so friendly and convenient for the users.

2.2 PROPOSED SOLUTION

Deep Learning (Convolutional Neural Network):

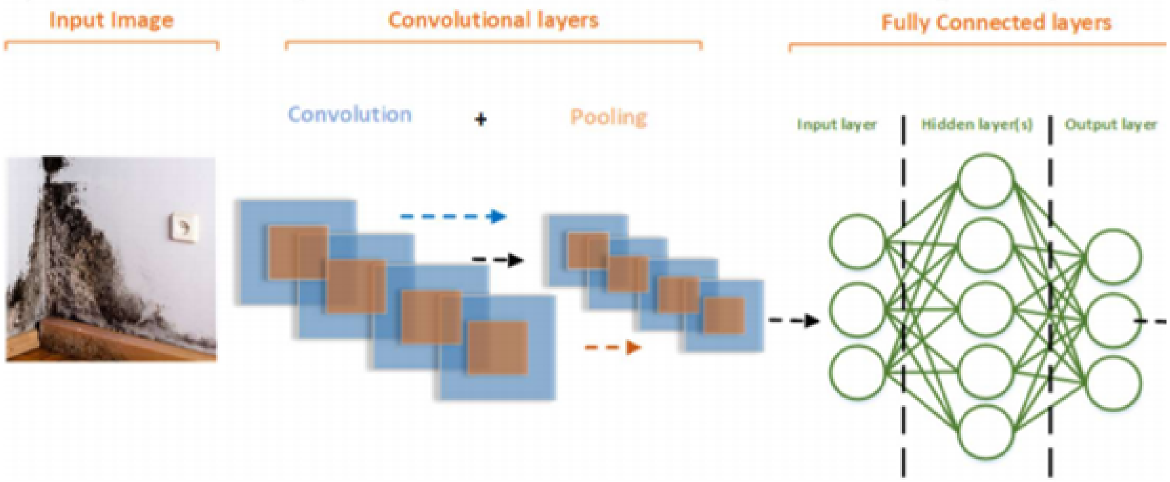
The proposed model has proven to be robust and able to accurately detect and localise building defects. The approach is being developed with the potential to scale-up and further advance to support automated detection of defects and deterioration of buildings in real-time using mobile devices and drones. 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 in buildings from images.

3. THEORITICAL ANALYSIS

The simplest form of a neural network is called perceptron. This is a single-layer neural network with exactly one input layer and one output layer. Multiple perceptron's can be connected together to form a multi-layer neural network with one input, one output and multiple inner layers, which also known as hidden layers. The more hidden layers, the deeper is the neural network (hence the name deep neural network). As a rule of thumb when designing a neural network, the number of nodes in the input layer is equal to the number of features in the input data, e.g. since our inputs are images with 3-channel (Red, Green, Blue) with 224x 224 pixels in each channel, therefore, the number of nodes in our input layer is 3x224x224. The number of nodes in the output layer, on the other hand, is determined by the configuration of the neural network. For example, if the neural network is a classifier, then the output layer needs one node per class label, e.g. in our neural network, we have four nodes corresponding to the four class labels: mould, stain, deterioration and normal. When designing a neural network, there are no particular rules that govern the number of hidden layers needed for a particular task. One consensus on this matter is how the performance changes when adding additional hidden layers, i.e. the situations in which performance improves or becomes worse with a second (or third, etc.) hidden layer. There are, however, some "empirical driven" rules of thumbs about the number of nodes (the size) in each hidden layer. One common 195 way suggest that the optimal size of the hidden layer should be between the size of the input layer 196 and the size of the output layer.

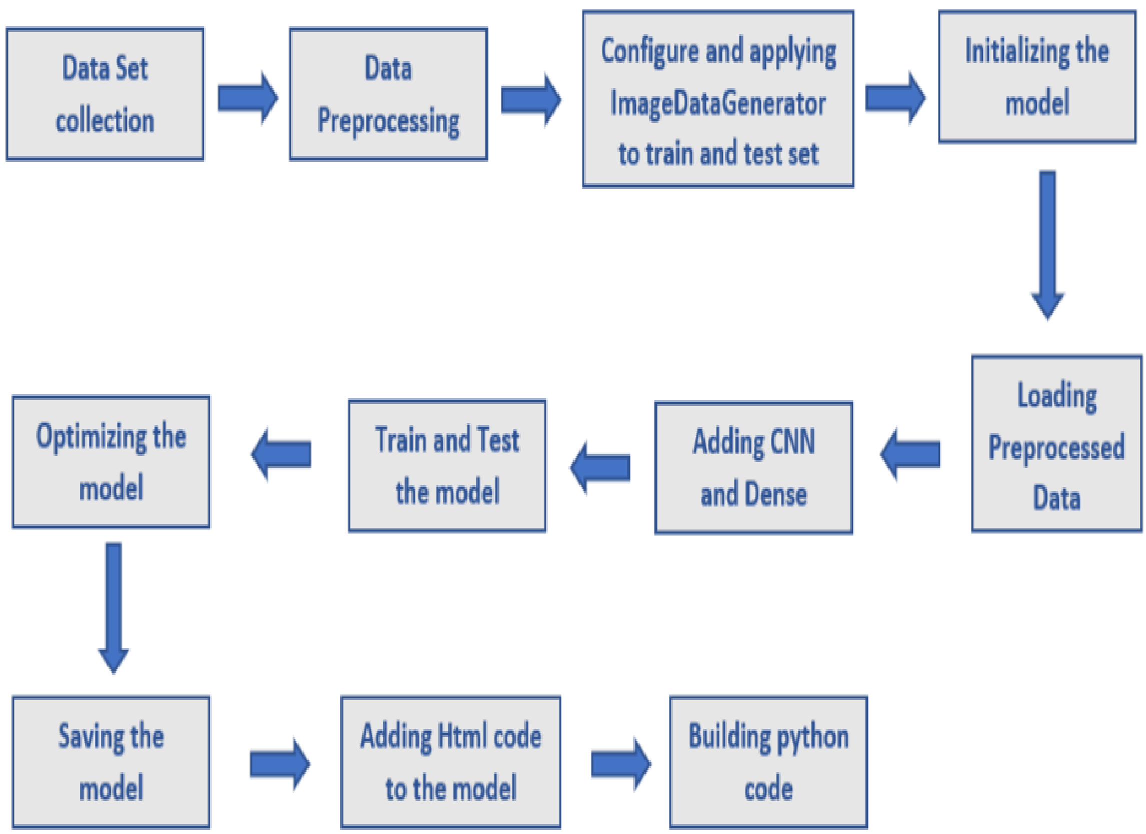
3.1. CNN LAYERS

Although, CNN have different architectures, almost all follow the same general design principles of successively applying convolutional layers and pooling layers to an input image. In such arrangement, the ConvNet continuously reduces the spatial dimensions of the input from previous layer while increasing the number of features extracted from the input image (see Figure 1).



Input images in neural networks are expressed as multi-dimensional arrays where each colour pixel is represented by a number between 0 and 255. Grey scale images are represented by a 1-D array, while RGB images are represented by a 3-D array, where the colour channels (Red, Green and Blue) represent the depth of the array. In the convolutional layers, different filters with smaller dimensions arrays but same depth as the input image (dimensions can be 1x1xm, 3x3xm, or 5x5xm, where m is the depth of the input image), are used to detect the presence of specific features or patterns present in the original image. The filter slides (convolved) over the whole image starting at the top left corner while computing the dot product of the pixel value in the original image with the values in the filter to generate a feature map. ConvNets use pooling layers to reduce the spatial size of the network by breaking down the output of the convolution layers into smaller regions were the maximum value of every smaller region is taken out and the rest is dropped (max-pooling) or the average of all values is computed (average-pooling). As a result, the number of parameters and computation required in the neural network is reduced significantly. The next series of layer in ConvNets are the fully connected (FC) layers. As the name suggests, it is a fully connected network of neurons (perceptrons). Every neuron in one sub-layer within the FC network, has a connection with all neurons in the successive sub-layer (see Figure 1). At the output layer, a classification which is based on the features extracted by the previous layers is performed. Typically, for a multi-classifier neural network, a Softmax activation function is considered, which outputs a probability (a number ranging from 0-1) for each of the classification labels which the network is trying to predict.

3.2 BLOCK DIAGRAM



3.3 SOFTWARE DESIGNING

- Jupyter Notebook Environment

● Spyder Ide

● Deep Learning Algorithms

● HTML

● Flask

We developed this building defects detecting by using the Python language which is an interpreted and high-level programming language and using the Deep Learning algorithms. for coding we used the Jupyter Notebook environment of the Anaconda distributions and the Spyder, it is an integrated scientific programming in the python language. For creating a user interface for the prediction, we used the Flask. It is a micro web framework written in Python. It is classified as a microframework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions, and a scripting.

4. EXPERIMENTAL INVESTIGATION

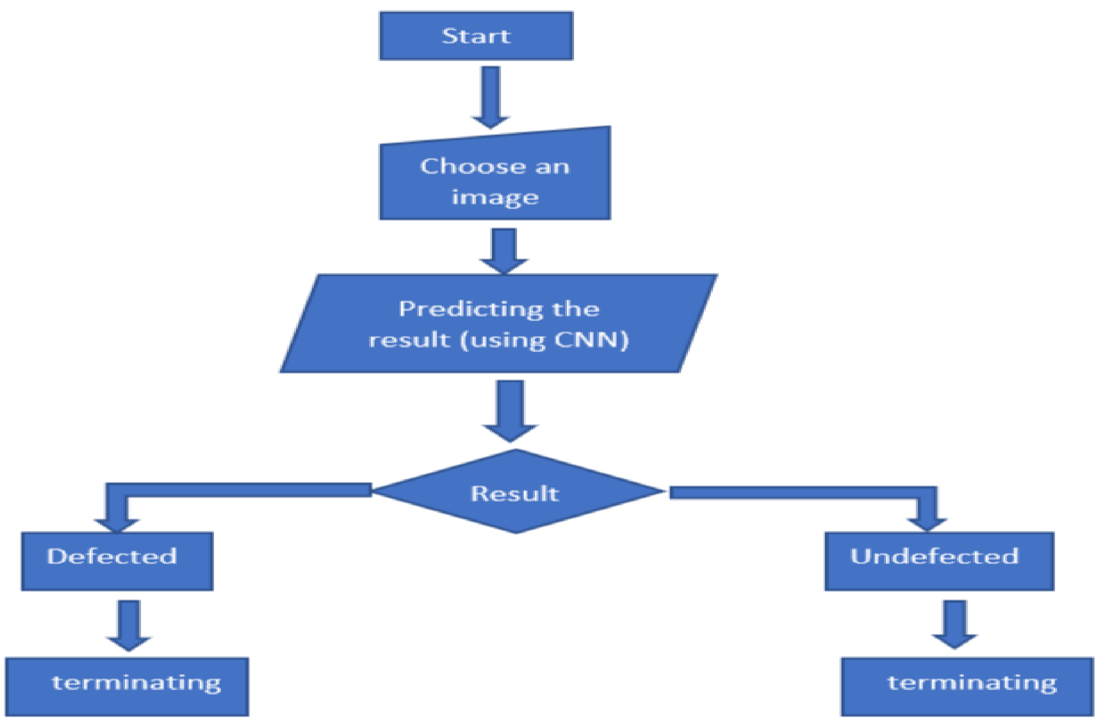
In this model we have created the dataset by collecting the images of both defected and undefected buildings. Next, we have initialized the model and created the CNN layers which processes the image using feature detector and convolves onto the raw image to produce the feature map. Then follows the pooling and flattening layers which helps in getting trained. These dense layer uses the **sigmoid** activation function and on providing the proper epoch gives the accuracy of 96.11%.

The dataset of those collected images are shown below.

Undefected building

Defected building

5. FLOWCHART

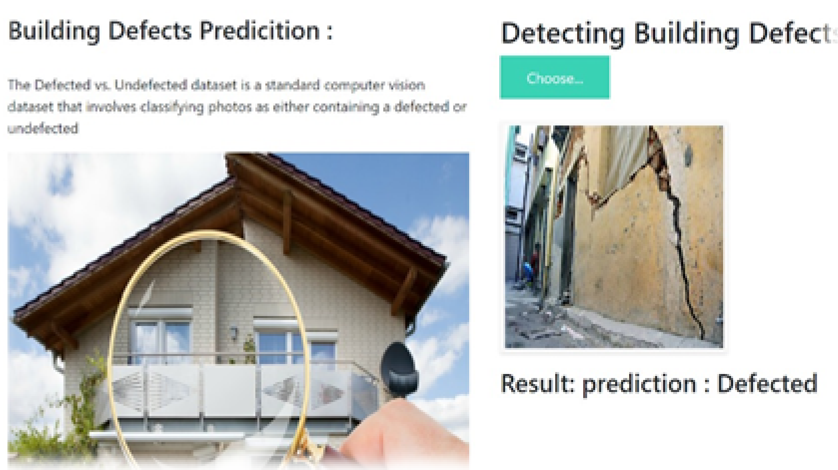


6. RESULT

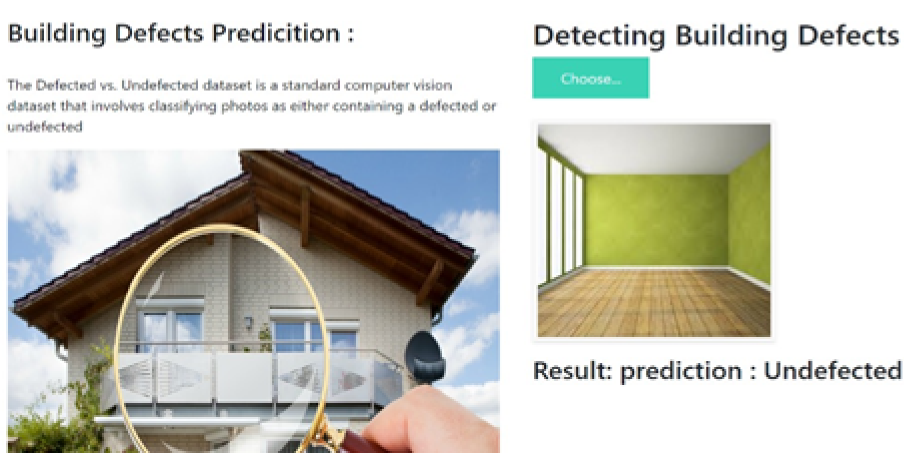
When the user chooses a random image as input our model uses the CNN classifier layers and detect the state of the building as either defected or undefected.

The results are shown below:

1. Result:Prediction: **Defected**



2. Result:Prediction: **Undefected**



7. ADVANTAGES AND DISADVANTAGES

Advantages:

- Image Classification using CNN model is widely used as they are powerful in achieving high accuracy with minimum error rate.

- The main advantage of CNN compared to its predecessors is that it automatically detects the important features without any human supervision.

Disadvantages:

- Lack of ability to be spatially invariant to the input data.
- They do not encode the position and orientation of the object into their predictions.

8. APPLICATIONS

- Decoding Facial Recognition - Facial recognition is broken down by a convolutional neural network into the following major components –
  - Identifying every face in the picture.
  - Focusing on each face despite external factors, such as light, angle, pose, etc.
  - Identifying unique features.
  - Comparing all the collected data with already existing data in the database to match a face with a name.

- Analysing Documents - Convolutional neural networks can also be used for document analysis. This is not just useful for handwriting analysis, but also has a major stake in recognizers.

- Historic and Environmental Collections - CNNs are also used for more complex purposes such as natural history collections.
- Understanding Climate - CNNs can be used to play a major role in the fight against climate change, especially in understanding the reasons why we see such drastic changes and how we could experiment in curbing the effect.
- Advertising - CNNs have already brought in a world of difference to advertising with the introduction of programmatic buying and data-driven personalized advertising.
- Grey Areas - Introduction of the grey area into CNNs is posed to provide a much more realistic picture of the real world.

9. 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.

10. FUTURE SCOPE

For the future works, the challenges and limitation that we were facing will be addressed. The presented paper had to set a number of limitations, i.e., firstly, multiple types of the defects are not considered at once. This means that the images considered by the model belonged 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 study. 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.

11. BIBLIOGRAPHY

1. Mohseni, H.; Setunge, S.; Zhang, G.M.; Wakefield, R. In Condition monitoring and condition aggregation for optimised decision making in management of buildings. Appl. Mech. Mater. 2013, 438, 1719–1725.
2. Shamshirband, S.; Mosavi, A.; Rabczuk, T. Particle swarm optimization model to predict scour depth around bridge pier. arXiv 2019, arXiv:1906.08863.
3. Annamdas, V.G.M.; Bhalla, S.; Soh, C.K. Applications of structural health monitoring technology in Asia. Struct. Health Monit. 2017, 16, 324–346.
4. Mita, A. Gap between technically accurate information and socially appropriate information for structural health monitoring system installed into tall buildings. In Proceedings of the Health Monitoring of Structural and Biological Systems, Las Vegas, NV, USA, 21–24 March 2016.

APPENDIX

Html:

```
{% extends "base.html" %} {% block content %}
<h2>Detecting Building Defects </h2>
<div>
    <form id="upload-file" method="post" enctype="multipart/form-data">
        <label for="imageUpload" class="upload-label">
```

```
            Choose...
        </label>
                <input type="file" name="file" id="imageUpload" accept=".png, .jpg,
.jpeg ,.jfif">
    </form>
    <div class="image-section" style="display:none;">
        <div class="img-preview">
            <div id="imagePreview">
            </div>
        </div>
        <div>
            <button type="button" class="btn btn-primary btn-lg "
id="btn-predict">Predict!</button>
        </div>
    </div>
    <div class="loader" style="display:none;"></div>
    <h3 id="result">
        <span> </span>
    </h3>
</div>
{% endblock %}
```

APP.PY

```
from __future__ import division, print_function

# coding=utf-8

import sys

import os

import glob

import numpy as np

from keras.preprocessing import image

from keras.applications.imagenet_utils import preprocess_input, decode_predictions

 from keras.models import load_model

from keras import backend

import tensorflow as tf

global graph

tf.compat.v1.disable_eager_execution()

graph=tf.compat.v1.get_default_graph()

#global graph

#graph = tf.get_default_graph()

from skimage.transform import resize

# Flask utils from flask import Flask, request, render_template

from werkzeug.utils import secure_filename

from gevent.pywsgi import WSGIServer

# Define a flask app
```

```
app = Flask(__name__)
# Model saved with Keras model.save()
model = tf.keras.models.load_model("models/testmodel.h5")
 # Necessary
# print('Model loaded. Start serving...')
# You can also use pretrained model from Keras
# Check https://keras.io/applications/
#from keras.applications.resnet50 import ResNet50
#model = ResNet50(weights='imagenet')
#model.save('')
print('Model loaded. Check http://127.0.0.1:5000/')
@app.route('/', methods=['GET'])
def index():
    # Main page
    return render_template('index.html')
@app.route('/predict', methods=['GET', 'POST'])
def upload():
    if request.method == 'POST':
        # Get the file from post request
        f = request.files['file']
        # Save the file to ./uploads
        basepath = os.path.dirname(__file__)
```

```
        file_path = os.path.join(
            basepath, 'uploads', secure_filename(f.filename))
        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():
            preds = model.predict_classes(x)
        #index = ['defected','undefected']
        #text = "prediction : "+[preds[0]]
            if preds[0][0]==0:
                prediction="Defected"
            else:
                prediction="Undefected"
        text = "prediction : "+prediction
               # ImageNet Decode
        return text
if __name__ == '__main__':
    app.run(debug=False,threaded = False)
```