

Indian Institute of Technology (Indian School of Mines), Dhanbad

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING



PROJECT REPORT

**SESSION (2020-21)
VIII SEMESTER**

TOPIC: Landmark Detection and Retrieval

SUBMITTED TO

Dr. Chiranjeev Kumar

(CSE DEPARTMENT)

SUBMITTED BY

Ashutosh Sahu(17JE003421)

Shivansh Awasthi(17JE002860)

ACKNOWLEDGEMENT

We would like to express heartfelt gratitude and regards to my project guide **Dr. Chiranjeev Kumar**, Department of Computer Science and Engineering, IIT (ISM) Dhanbad. We convey a humble thanks to him for his valuable cooperation, support and suggestion throughout the project work which made this project successful.

We are thankful to all the faculties of the Department of Computer Science and Engineering, IIT (ISM) Dhanbad for their encouraging words and valuable suggestions towards the project work. Last but not the least we want to acknowledge the contribution of my parents, family members, and friends for their constant and never-ending motivation.

We would like to take this opportunity to express my gratitude to the people who have been instrumental in the successful completion of the project.

Date: 6th May 2021

Place:

Ashutosh Sahu

Admission No. 17JE003421

Shivansh Awasthi

Admission No. 17JE002860

TABLE OF CONTENTS

<i>Contents</i>	<i>Page No.</i>
Chapter 1	
Abstract	4
Introduction	5
Chapter 2	
Literature review	6
Dataset description and annotations	7
Chapter 3	
Saliency based feature extraction techniques	8
Semantic Segmentation techniques	10
Chapter 4	
Convolutud Neural Networks	12
Resnet-50	13
InceptionV3	14
MobileNetV2	15
Chapter 5	
API and Database Creation	16
Chapter 6	
Results	19
References	22

Chapter 1

1.1 Abstract

Landmark recognition is considered to be a challenging domain in the field of image classification. It mostly deals with architectures of high variances. Different orientations, colors and the intensity of the images of the monuments play an important role in the recognition of the landmarks in their images. Various features need to be detected by the convolutional neural network and were used to automatically classify these monuments. Apart from automatic feature extraction using CNNs, we used feature extraction and segmentation techniques to manually detect the important part of each image which was then fed into the CNN to improve the accuracy.

Indian Landmarks dataset was not readily available, hence we scraped data from google images which was used for our purpose. We chose 30 classes for our project, with 50-100 images per class. These images were cropped, zoomed, rotated, etc. to increase our sample size. The images were cropped, and important features were extracted which were fed to the neural network. Trials have been done on the physically procured dataset that is made out of pictures of various landmarks where every landmark has pictures from various precise perspectives.

After performing various experiments in feature extractions techniques, tuning out different models, fine tuning and experimenting with different hyper parameters we were able to achieve an accuracy of 98 percent using our model.

1.2 Introduction

Earlier this year I visited the southern part of India with my friends. And the most significant problem we faced while exploring the places was that of tourist guides. To know the history and the importance of the beautiful places we had to hire a tourist guide but were charged a significant amount of money. They were also making false claims for their marketing and to increase the interest of the tourists. This was my motivation behind this project.

We wanted to create something that would make tourism easier, affordable, and enjoyable. This could enhance the experience of a tourist and save them from the hassle of a tourist guide. We have hence designed a landmark detection project.

A landmark suggests a design that has been developed to celebrate an individual, an occasion or which has become a significant part to a gathering of people as a piece of them recollecting notable occasions or social legacy, or to act as an illustration of the notable engineering. India has its own rich history, culture and heritage and hence there are various monuments of significance in India. But most people don't know the history behind those monuments. Often they hire tourist guides, who charge them hefty amounts to share the knowledge about the place.

Our technique will not only classify the query image into its respective class, but will also provide a brief history behind it. There are several significant applications of this especially in indian cities. This technique can be used by tourism departments to promote tourism in their respective states.

Classifying a monument is difficult and has been an ongoing research subject for various reasons. With increase in the similarities among monuments it is very difficult to classify. For example, the Taj Mahal and Bibi Ka Maqbara. Many monuments are made with similar architecture and hence have similar features, which makes it difficult for the neural network architecture. With increase in the size of the dataset and decrease in variance among monuments, leads to a decrease in accuracy. Also the noise(humans, trees, etc.) needs to be removed in order to focus only of the salient pixels of the image.

In this report, we first describe the dataset, then we focus on various feature extraction techniques then we shift to the models and various experiments we had done. Finally the last part deals with future scope and applications.

2.1 Literature Review

The project chosen is a novel idea and it is an ongoing research project with various scope of improvements. We have developed a noble technique in the manual feature extraction techniques and to remove the noises in the image.

Only research paper available for Indian Monuments was by a bunch of students of IIT Roorkee, they used HOG feature extraction techniques. Other papers that we found used SIFT and SURF feature extraction techniques.

Our main purpose was to create masked images to remove the noise present. So we thought of using segmentation techniques. We went through the work of Jonathan Long, Evan Shelhamer, Trevor Darrell of UC Berkeley on Fully Convolutional Networks for Semantic Segmentation. It was one of the initial semantic segmentation techniques. After FCN, a lot of work has been in this field and google's DeepLab-V3 which uses the Resnet 101 backbone is the current state of the art segmentation technique.

We also went through the work of Olaf Ronneberger, Philipp Fischer, and Thomas Brox on U-Net: Convolutional Networks for Biomedical Image Segmentation. U-Net is more successful than conventional models, in terms of architecture and in terms of pixel-based image segmentation formed from convolutional neural network layers. It's even effective with limited dataset images. Hence, we found it perfect for our purpose.

After extracting the features manually, we had to decide on different Deep Convolutional Neural Net architectures. Since we needed a deep feature extractor we tried Resnet50 and Inception_V3 network, which satisfied our purpose. We also tried MobilnetV2 architecture to make a model suitable for mobile applications.

2.2 Dataset Description and Annotations

The dataset was acquired manually using a python library which scrapes data from the google images. The dataset obtained had similar pictures and many non-significant images which had to be manually removed initially. After manual pruning and annotations, each folder contains 70-80 different images of the same landmark, considering all possible scenarios and noises, the dataset scraped was diversified, to obtain a better result.

As the dataset collected was not enough, we had to preprocess those images to further diversify our dataset. Each image chosen was rotated, tilted, cropped randomly, etc. to obtain more images per class. So these images were considered to be the original class.

After applying feature extraction techniques on the existing original dataset, we obtained a more relevant and more salient dataset. All the noise present in the images were removed using segmentation techniques and the images which had more than 40 percent noise were discarded automatically.

After applying the segmentation technique, the images formed were stored in a new folder and were passed on to the neural network architecture after preprocessing.

3.1 Saliency based feature extraction techniques

Image Saliency is the thing that sticks out and how quick one can rapidly zero in on the most applicable pieces of what you see.

Presently, on account of tourist spots, the less remarkable area is basic foundations, that is of blue sky, the clamor in the foreground(trees, individuals, and so forth).

The compositional plan of the landmarks is the thing that separates between the classes.

Initially discovers include guides and afterward apply non-direct Enactment guides to feature "critical" areas in the picture. We used various segmentation and feature extraction techniques to detect important locations per training image. Those pictures were utilized for multi-stage preparation. It assisted with improving our exactness by 3-4%.

The model considers three features in an image, namely colours, intensity and orientations. These combinations are presented in the saliency map. The model uses a winner-takes-it-all neural network for working with the saliency map.

The steps are given below:

1. The three features mentioned above are extracted from input images.
2. For colours, the images are converted to red-green-blue-yellow colour space. For intensity, the images are converted to a grayscale.
3. The orientation feature is converted using Gabor filters with respect to four angles.
4. All of these processed images are used to create Gaussian pyramids to create feature maps.
5. The feature maps are created with regard to each of the three features. The saliency map is the mean of all the feature maps.

The saliency map has a similar shape as the info picture, the saliency guide can be utilized as a cover on top to feature the significant piece of the information picture, concerning the anticipated mark. According to the meaning of slopes, this saliency map reveals to us how much the expectation score for class c would change if the RGB pixel forces in the featured space of the picture are marginally expanded.

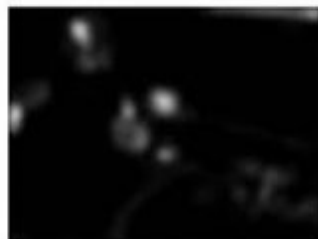
Original Image



Saliency Map



Proto Objects



3.2 Semantic Segmentation techniques

Since the image might contain noise(humans, trees, etc.) we can use semantic segmentation techniques to remove those parts and create a mask.

We used U-Net for our segmentation purpose. It classifies each pixel as one of 21 classes, including backgrounds, humans, trees, etc.

We also tried to perform segmentation using FCN(Fully Convolutional Neural Network). U-Net is more successful than conventional models, in terms of architecture and in terms of pixel-based image segmentation formed from convolutional neural network layers. It's even effective with limited dataset images.

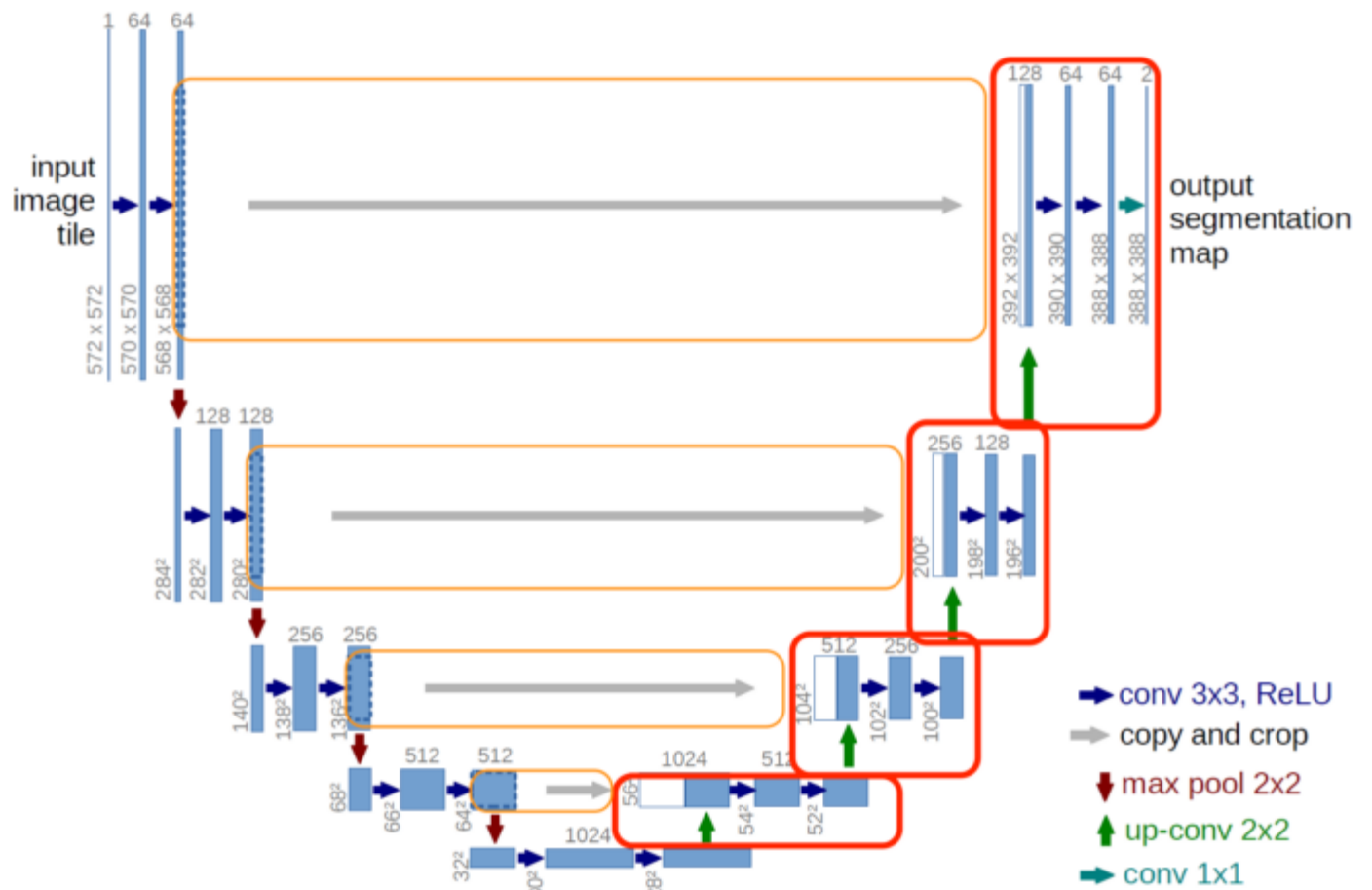
Hence we settled with U_Net.

3.2.1 U-Net(Network Architecture)

The U-Net architecture has a total of 23 convolutional layers. The first part is the encoder part and the latter is the decoder part. It gets its name because of the shape of the architecture(U-shaped).

In the encoder part, each stage has a double convolution layer followed by a max pooling layer to down sample the images. Two 3x3 unpadded convolutions are applied one after the other. Each of the convolution layers is followed by a rectified linear unit(ReLU) activation function. Then it is followed by a downsampling step using the 2x2 max pooling operation with a stride of 2. At each downsampling we increase the number of feature channels in order to extract more features from the image.

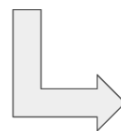
In the decoder part, transposed convolution is used to upsample the images. A 2x2 convolution is used for upsampling, which is often referred to as up-convolutions. It halves the number of feature channels and a concatenation from the corresponding level of the encoder part.



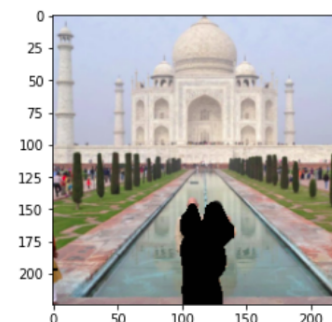
Input Image



Mask created



Output image



4.1. Convolved Neural Networks

Convolutional Neural Network is a Deep Learning algorithm that helps differentiate various objects/images one from another. compared to other classification algorithms it requires much lower preprocessing.

Its architecture is analogous to the connectivity pattern of Neurons in the Human Brain.

Convolutional neural networks have multiple layers of artificial neurons. These artificial neurons are mathematical functions that calculate the weighted sum of multiple inputs and output an activation value.

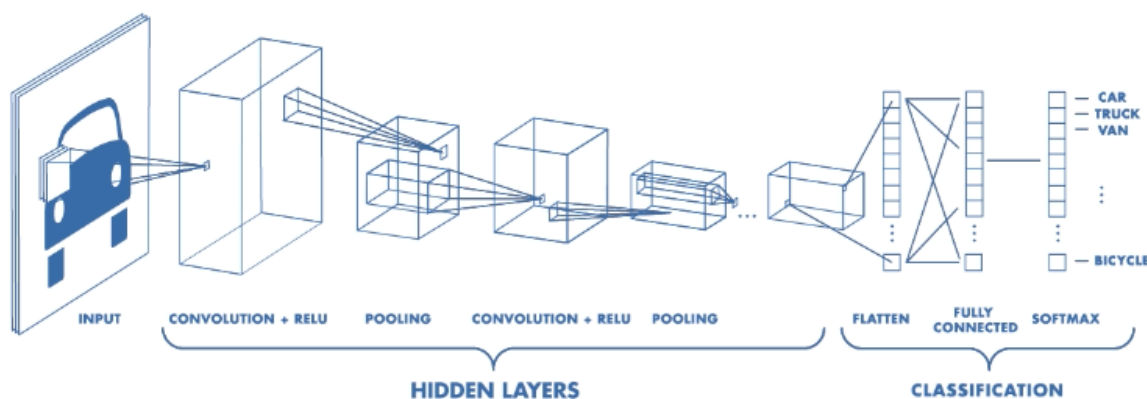
The CNN is fed the pixel values of the images and these artificial neurons pick out various visual features.

This operation of multiplying pixel values by weights and summing them is called “convolution” (hence the name convolutional neural network).

CNNs have several such convolution layers which extract different features of the image.

CNNs have two components:

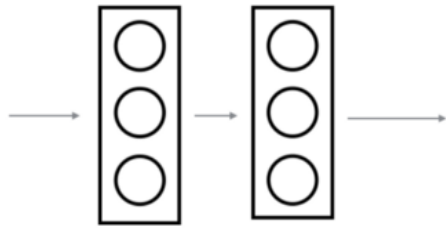
1. Feature extraction part- uses filters to extract various features
2. Classification part- uses activation functions to assign a probability to predict what the image is.



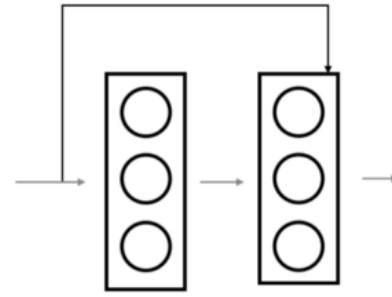
4.2. Resnet-50

The Resnet model was first to introduce the skip connection. Without the skip connection the layers used to be stacked one after the other while in skip connection we stack the layers as before but we now also add the original input to the output of the convolution block.

without skip connection



with skip connection



The skip connection removes the problem of vanishing gradient as it allows an alternate path for gradient flow.

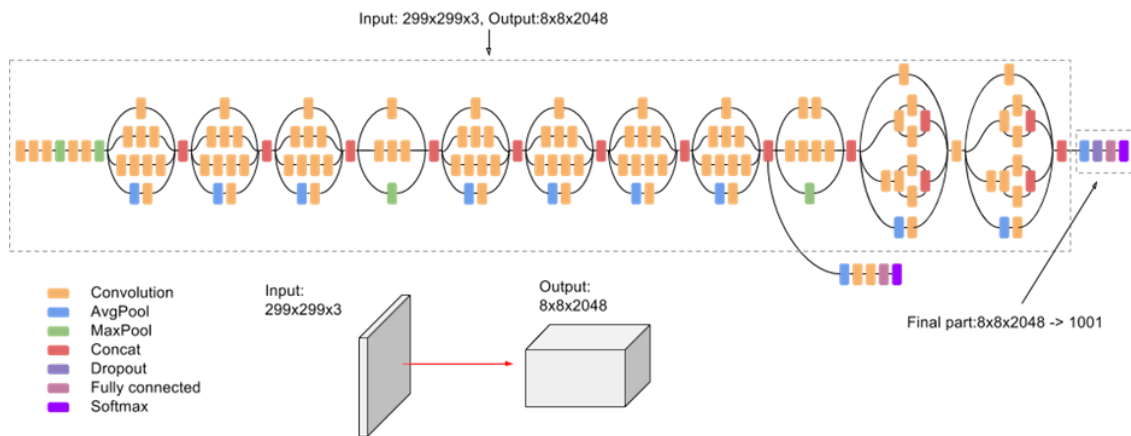
The ResNet-50 model has 5 stages each with a convolution and Identity block. Each convolution block has 3 convolution layers and each identity block also has 3 convolution layers. Resnet-50 is a 50 layers deep CNN. The pre-trained model can classify images into 1000 different categories. It has an input image size of 224×224 .

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$
conv3_x	28×28	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 8$
conv4_x	14×14	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 23$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 36$
conv5_x	7×7	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

4.3. InceptionV3

InceptionV3 is a widely used image recognition model that has symmetric and asymmetric building blocks, including convolutions, average pooling, max pooling, concatenations, dropouts, and fully connected layers. This model extensively uses batch norm and the Softmax function is used to calculate the loss.

It is a 48 layers deep CNN. The pre-trained model can classify images into 1000 different categories. It has an input image size of 299×299 .



4.4. MobileNetV2

This is a convolutional neural network model that is designed to be used on mobile devices. This uses the concept of depthwise separable convolution neural networks that tend to reduce the size of the model. It is a very effective feature extractor for object detection and segmentation. MobileNetV2 is an improved version of mobileNetV1.

MobileNetV2 has two added features:

1. Linear bottlenecks between the layers
2. Shortcut connections between the bottlenecks

MobileNetv2 is a 53 layers deep CNN. The pre-trained model can classify images into 1000 different categories. It has an input image size of 224*224.

Input	Operator	t	c	n	s
$224^2 \times 3$	conv2d	-	32	1	2
$112^2 \times 32$	bottleneck	1	16	1	1
$112^2 \times 16$	bottleneck	6	24	2	2
$56^2 \times 24$	bottleneck	6	32	3	2
$28^2 \times 32$	bottleneck	6	64	4	2
$14^2 \times 64$	bottleneck	6	96	3	1
$14^2 \times 96$	bottleneck	6	160	3	2
$7^2 \times 160$	bottleneck	6	320	1	1
$7^2 \times 320$	conv2d 1x1	-	1280	1	1
$7^2 \times 1280$	avgpool 7x7	-	-	1	-
$1 \times 1 \times 1280$	conv2d 1x1	-	k	-	-

MobileNetV2 Overall Architecture

t : expansion factor

c : number of output channels

n : repeating number

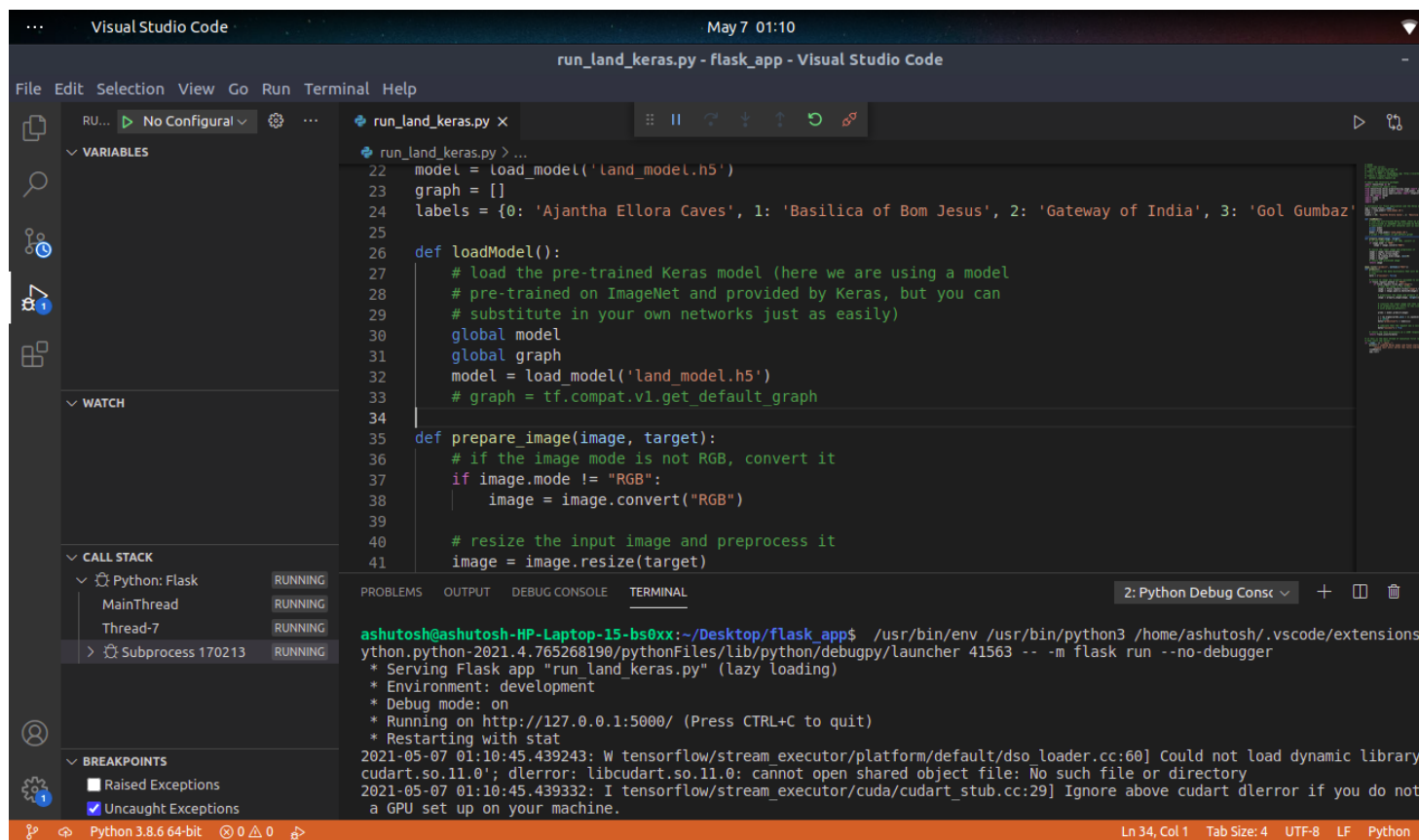
s : stride. 3×3 kernels are used for spatial convolution

Chapter 5

5.1 API and Database Creation

We deployed our model using flask on the local server, to create an API which would return the following class. We tested the API using Postman, and it worked perfectly. The model so well, that it was able to classify images of hand drawings as well.

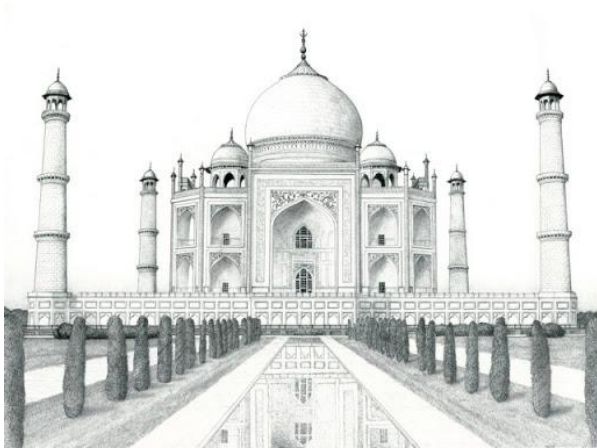
We also made a database which contained information about each landmark that we have chosen. So on call to the API it would redirect to a page containing the information in an organised way.



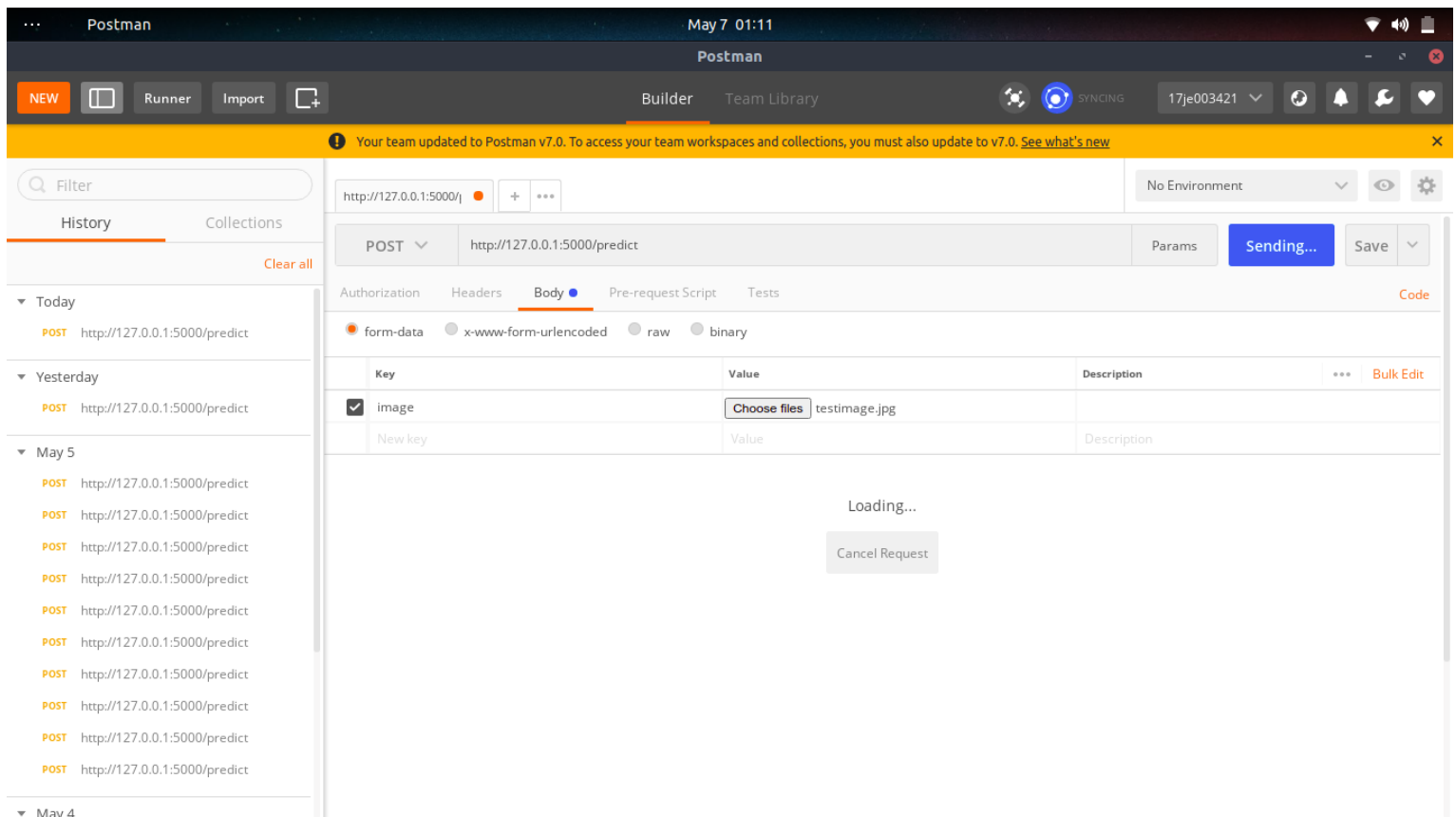
```
run_land_keras.py > ...
22 model = load_model('land_model.h5')
23 graph = []
24 labels = {0: 'Ajantha Ellora Caves', 1: 'Basilica of Bom Jesus', 2: 'Gateway of India', 3: 'Gol Gumbaz'}
25
26 def loadModel():
27     # load the pre-trained Keras model (here we are using a model
28     # pre-trained on ImageNet and provided by Keras, but you can
29     # substitute in your own networks just as easily)
30     global model
31     global graph
32     model = load_model('land_model.h5')
33     # graph = tf.compat.v1.get_default_graph()
34
35 def prepare_image(image, target):
36     # if the image mode is not RGB, convert it
37     if image.mode != "RGB":
38         image = image.convert("RGB")
39
40     # resize the input image and preprocess it
41     image = image.resize(target)
```

```
ashutosh@ashutosh-HP-Laptop-15-bs0xx:~/Desktop/flask_app$ /usr/bin/env /usr/bin/python3 /home/ashutosh/.vscode/extensions/python.python-2021.4.765268190/pythonFiles/lib/python/debugpy/launcher 41563 -- -m flask run --no-debugger
* Serving Flask app "run_land_keras.py" (lazy loading)
* Environment: development
* Debug mode: on
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
* Restarting with stat
2021-05-07 01:10:45.439243: W tensorflow/stream_executor/platform/default/dso_loader.cc:60] Could not load dynamic library 'libcuda.so.11.0'; dlderror: libcudart.so.11.0: cannot open shared object file: No such file or directory
2021-05-07 01:10:45.439332: I tensorflow/stream_executor/cuda/cudart_stub.cc:29] Ignore above cudart dlerror if you do not have a GPU set up on your machine.
```

Model being hosted in the local server



Testing image



Sending Request

Postman May 7 01:11

NEW Runner Import Builder Team Library SYNCING 17je003421

! Your team updated to Postman v7.0. To access your team workspaces and collections, you must also update to v7.0. [See what's new](#)

Filter

History Collections

▼ Today

- POST http://127.0.0.1:5000/predict

▼ Yesterday

- POST http://127.0.0.1:5000/predict

▼ May 5

- POST http://127.0.0.1:5000/predict
- POST http://127.0.0.1:5000/predict
- POST http://127.0.0.1:5000/predict
- POST http://127.0.0.1:5000/predict
- POST http://127.0.0.1:5000/predict
- POST http://127.0.0.1:5000/predict
- POST http://127.0.0.1:5000/predict
- POST http://127.0.0.1:5000/predict
- POST http://127.0.0.1:5000/predict

▼ May 4

http://127.0.0.1:5000/predict No Environment

POST http://127.0.0.1:5000/predict Params Send Save

Authorization Headers Body Pre-request Script Tests Code

form-data x-www-form-urlencoded raw binary

Key	Value	Description	...	Bulk Edit
<input checked="" type="checkbox"/> image	<input type="button" value="Choose files"/> testimage.jpg			
New key	Value	Description		

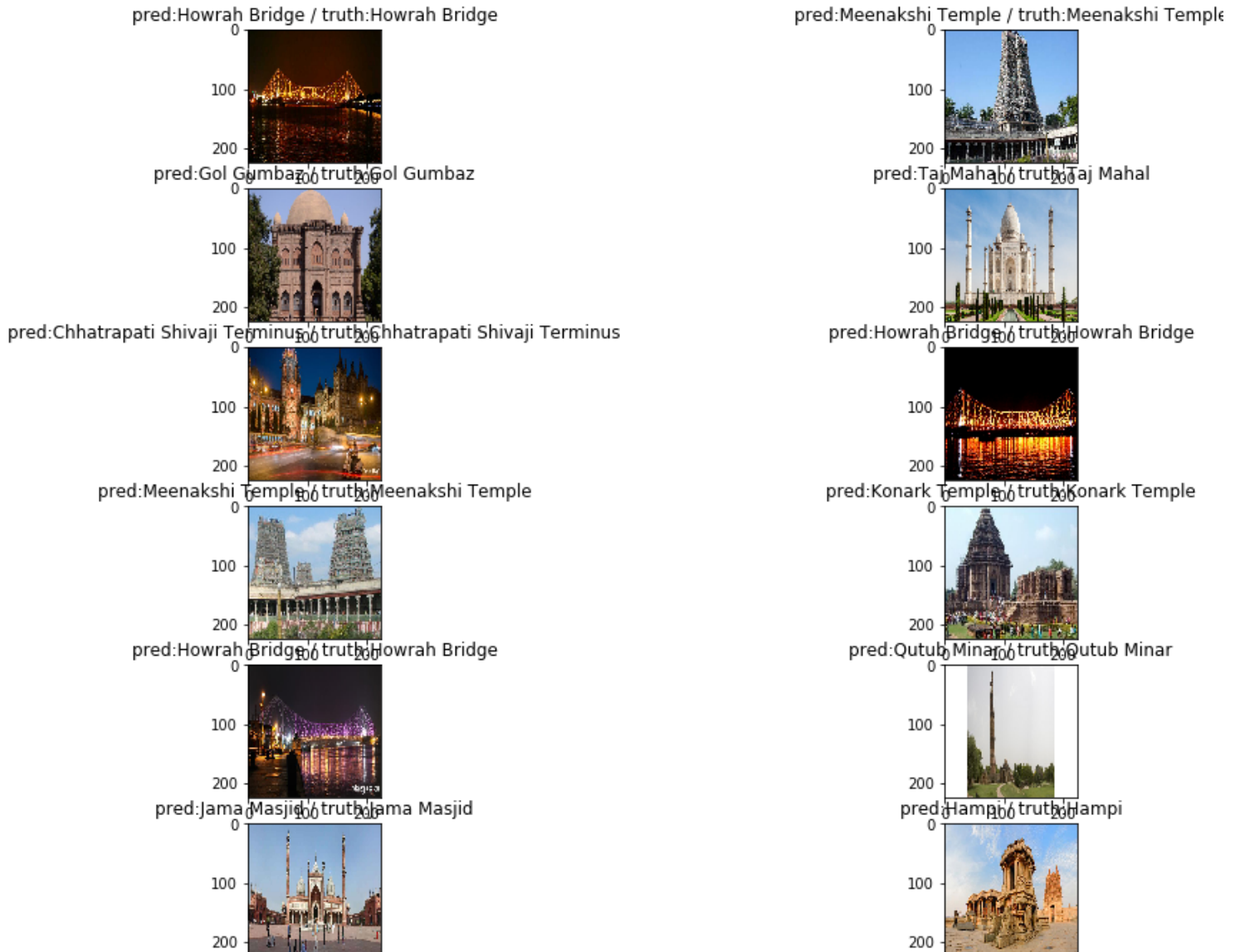
Body Cookies Headers (4) Test Results Status: 200 OK Time: 3069 ms

Pretty Raw Preview JSON

```
1 {
2   "prediction": "Taj Mahal",
3   "success": true
4 }
```

Predictio

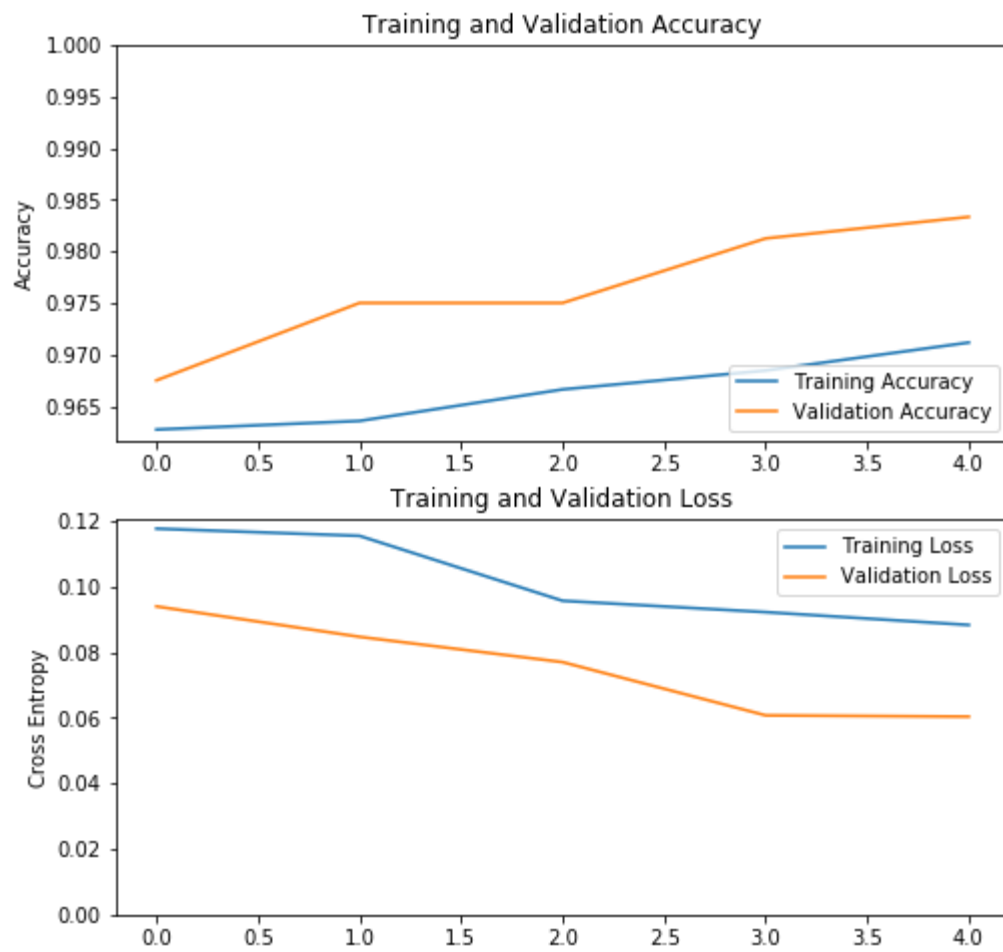
6. Results



We performed using different models such as InceptionV3 and Resnet with fine tuning for deep feature extraction technique and MobileNetV2 for applying the model to mobile application, and the table mentioned shows us the accuracies we got. We observed that using the segmentation technique was beneficial.

Model	Data Subset	Train	Validation
InceptionV3	Original Images	95.72	94.12
	Original + Salient images	98.43	97.61
Resnet50	Original Images	97.71	95.67
	Original + Salient images	99.83	98.87
MobileNetV2	Original Images	94.32	92.21

The variation of loss functions and accuracies with respect to epochs were found out to be ideal as well.



References

- [1] Ronneberger, Olaf, Philipp Fischer, and Thomas Brox. "U-net: Convolutional networks for biomedical image segmentation." *International Conference on Medical image computing and computer-assisted intervention*. Springer, Cham, 2015.
- [2] Sandler, Mark, et al. "Mobilenetv2: Inverted residuals and linear bottlenecks." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018.
- [3] Xia, Xiaoling, Cui Xu, and Bing Nan. "Inception-v3 for flower classification." *2017 2nd International Conference on Image, Vision and Computing (ICIVC)*. IEEE, 2017.
- [4] Saini, Aradhya, et al. "Image based Indian monument recognition using convoluted neural networks." *2017 International Conference on Big Data, IoT and Data Science (BIG)*. IEEE, 2017.
- [5] Harel, Jonathan, Christof Koch, and Pietro Perona. "Graph-based visual saliency." (2007): 545-552.
- [6] Noh, Hyeonwoo, et al. "Large-scale image retrieval with attentive deep local features." *Proceedings of the IEEE international conference on computer vision*. 2017.
- [7] Amato, Giuseppe, Fabrizio Falchi, and Paolo Bolettieri. "Recognizing landmarks using automated classification techniques: Evaluation of various visual features." *2010 Second International Conferences on Advances in Multimedia*. IEEE, 2010.