

# Landmark Recognition using Inception-v3

Abhijit Nelli<sup>1</sup>, Komal Nalige<sup>2</sup>, Rennith Abraham<sup>3</sup>, Manohar R<sup>4</sup>

<sup>1</sup>Abhijit Nelli: B.E. Student, Dept. of Information Science and Engineering, Sir MVIT, Bengaluru-562157

<sup>2</sup>Komal Nalige: B.E. Student, Dept. of Information Science and Engineering, Sir MVIT, Bengaluru-562157

<sup>3</sup>Rennith Abraham: B.E. Student, Dept. of Information Science and Engineering, Sir MVIT, Bengaluru-562157

<sup>4</sup>Manohar R: Assistant Professor, Dept. of Information Science and Engineering, Sir MVIT, Bengaluru-562157

\*\*\*

**Abstract** - Recognizing a particular landmark in a photograph is not considered a meager issue since it is faced by a multitude of people all over the world. Our paper brings about a solution to this very problem at hand, by recognizing and classifying the landmarks present in images. What we are dealing with is an extreme image recognition and classification problem where the dataset consists of an enormous congregation of images. To surmount this very problem, we have employed the technology of Transfer Learning. Since this problem is an image classification problem we have used the concept of Google's inception V3 model as our architecture. It was developed by Google and is pre-trained on 1000 classes which belong to the imageNet dataset. There are many Transfer learning models specifically geared to deal with image classification. We have chosen inception V3 model because of the accuracy of the results it can produce and the fact that the error is minimized.

**Key Words:** Inception v-3, Bottlenecks, Softmax, CNN, imageNet

## 1. INTRODUCTION

Recognizing landmarks as a series of images is a taxing process. For any given landmark the appearance differs substantially. The variability of observed landmarks are affected by various factors such as different perspectives, illumination change and the geometry of various imaging devices. Using accurate 3D information is a tedious process in landmark recognition applications. Hence it is not possible to use the standard object recognition techniques for landmark recognition. Therefore, the finest approach that can be employed for this process is to use image-based technique's in which landmarks are represented as a collection of images enabling these techniques to capture the "distinctive" appearance of an object. Many a time, when we travel to abstract locations and take photos or while watching a movie, we notice a particular monument and wonder the whereabouts of this monument. By being able to create a model that recognizes landmarks we are able to address to this issue.

This paper aims to create an engine which is capable of classifying and recognizing landmarks combining several procedures and techniques which enables the engine a systematic approach to recognize images accurately. The concept utilized for classifying and recognizing landmarks is Transfer Learning. This was used because of its ability to be able to apply knowledge and skills it gained from a problem

onto to a similar problem. Image acquisition and Data Pre-processing are performed on the data of images, these images are then catalogued into different directories or folders. Folders or directories are created for which contains images of the same landmark as these are further used by the Neural Network. Inception v-3 model which is a Convolutional Neural Network is downloaded and forms the basis for classifying the images in our dataset. Followed by this is, Bottlenecks for each image are calculated before the final layer of our model is added. The final layer in the inception v-3 model is replaced with a Softmax activation which performs the actual classification of the images. The retrain.py amalgamates these processes. Once the training is complete, the trained model is re-used.

## 2. RELATED WORK

### Landmark Recognition with Deep Learning Filippo Galli-2016

This paper proposes a solution for visual landmark recognition by using the applications of deep learning. The main objective was to identify various images for an autonomous robot navigation system. The deep learning application used was Convolutional Neural Networks. This method cannot be applied to our application because their CNN structure and training data is tailored for navigation of a robot.

### Object Detection and Recognition in Images 1Sandeep Kumar, 2Aman Balyan, 3Manvi Chawla Computer Science &Engineering Department, Maharaja Surajmal Institute of Technology, New Delhi, India.-2017

In this the paper images are divided into various cells, if the centre of the image falls in a grid cell it is responsible for analysing that object, a bounding box is created which encompasses the object. For this model a simple CNN is used as the initial layers are used for feature extraction and the fully connected layers are used for predicting the co-ordinates and output probabilities. This approach was not employed as this paper focuses on extracting the contents within the bounding box, whereas in our approach we need to extract not only an image withing the bounding box but the whole image given as input.

**Google Landmark Recognition using Transfer Learning- Project by Catherine McNabb, Anuraag Mohile, Avani Sharma, Evan David, Anisha Garg-2018**

In this paper the dataset used is of a million images, each image in the dataset is in the form of a link and these images need to be downloaded from these links. This process uses the concept transfer learning for training its model but the architecture used is VGG-16. This architecture was easily getting fooled by the unrelated images hence DeLF had to be employed to make the model deal with this problem and increase its efficiency and accuracy of recognizing landmarks. But this process had to make use of 8 CPUs and GPU.

**3. DATASET**

Our dataset consisted of images containing landmarks. These are used to train the model. Care was taken to see that in each image, the landmark present was highlighted, to help the model learn more effectively. Images were downloaded from Google, either in .jpeg formats or converted to such format. Also images were cropped and augmented to suit the needs of training the model. In any traditional Convolutional Neural Network, images are given to the network for training in directory format. Therefore, this image data was put into a folder, each sub folder representing 1 class or category of images. Landmark images are put into the subfolders corresponding to the class label that they represented. These are then ready to be given to the neural network for training.

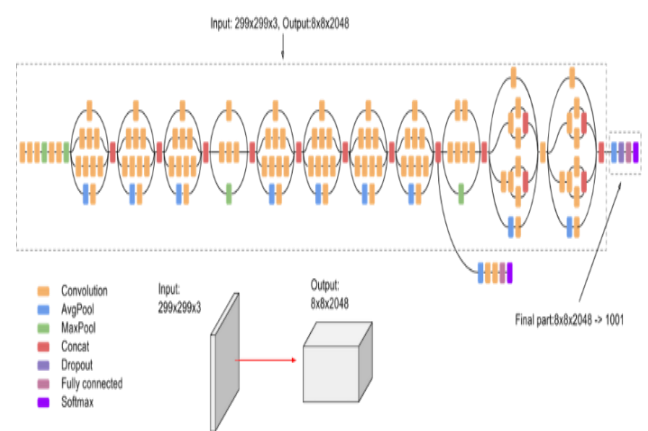
**4. PROPOSED SOLUTION**

**4.1 TRANSFER LEARNING**

Transfer learning is one of the leading areas in the realms of Machine Learning and Artificial Intelligence. Its principle idea is based on the reuse of knowledge. The driving force behind this technology is the process of utilization of information gained in solving another task and thus improving learning. It involves a Neural Network being trained on the dataset of a task similar to the task at hand. The information gained from this exercise is learned and then is used for our purpose of solving the recognition and classification task at hand. For our purpose, we use the pretrained model approach rather than the developed model approach due to time and resource constraints. This approach involves choosing a source model from a plethora of already existing models that is trained on a very large dataset. It is then fine-tuned to meet the specific needs that must be met to solve the problem. Some of the popular Transfer learning models for image classification include Oxford’s VGG14 and VGG16, Google’s Inception v3 and v4 and Microsoft’s Resnet models. For our task, we have chosen the Inception V3 model, due to lower error values as compared to the other models mentioned.

**4.2 INCEPTION-v3 MODEL**

Inception V3 is a Deep Convolutional Network and is Google’s 3<sup>rd</sup> version of Deep Learning Architecture series. It is trained on a dataset containing 1000 different classes that are derived from the original imageNet dataset that had 1 million classes of images. It has been ascertained to have an accuracy greater than 78.1%. The architecture of this model is comprised of symmetric and asymmetric blocks of Deep Convolutional network layers like Convolutional, Average Pool, Max Pool, Concatenation, dropout, fully connected and softmax layers. Inception v3 doesn’t include any of the landmark images that are present in our training set. But the kinds of knowledge gathered on the 1000 classes makes it possible for the network to be able to differentiate between features of our images.



**Fig 1 : Inception v3 architecture**

In any traditional neural network, neurons are arranged in the form of layers. Different layers may perform different forms of transmissions on the inputs they receive. Signals traverse from first input layer to the last output layer ,multiple times usually. As the last hidden layer , the ‘Bottleneck’ has enough summarized knowledge to supply to the final layer that actually performs the classification tasks.

In our solution algorithm, the retrain.py script is the core idea , as it is in any custom image classification task. Here, we remove the previously existing top layer and instead train a new one (Bottleneck) on the downloaded landmark images. A bottleneck is a colloquial term that is used for the layer present just before the final layer, and contrary to it’s name, it doesn’t slow down the network. It receives an input of a 2048 dimensional vector for each image. We have chosen the final layer on top of the bottleneck to be a Softmax layer that receives the bottleneck information that is garnered and does the actual classification of images. This exercise of final layer retraining can work because the kinds of data needed to differentiate between the 1000 imageNet classes is also useful to work on our landmark classes.

If we assume that the Softmax layer has ‘N’ class labels:

$N + 2048 * N$  (or  $1001 * N$ ) number of model parameters corresponding to learned biases and weights are learned. In

this way, we first inspect all the images and calculate bottlenecks for each of them. Each image is used many times during the training process and calculations may take up a considerable amount of time. But since the lower layers of the model aren't changed, their outputs that were previously stored in the cache memory of the system are recycled, thus saving some training time.

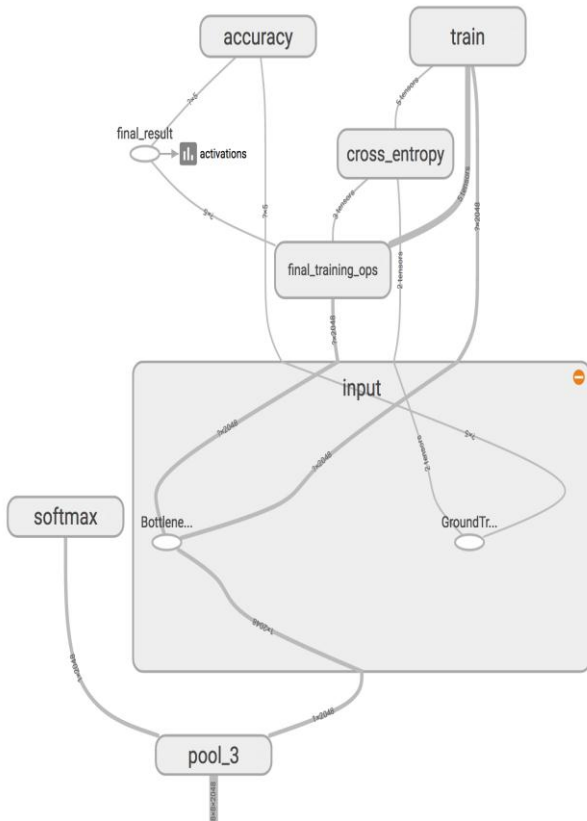


Fig 2 : Inception-v3 Bottlenecks generation

After these bottlenecks are generated, actual training of the final layer above the bottleneck layers, that is Softmax begins. We then start with the training process, for 500 epochs and a learning rate of 0.01 initially. Each step picks 10 random images from the training set, finds their cached bottleneck values, and forwards them to the final Softmax layer to generate predictions. These are checked against actual labels and results are used to update the Softmax layer weights through a back-propagation process.

Therefore, in a nutshell, we have downloaded the pre-trained inception v3 model, calculated bottlenecks for each image in our dataset, added a final Softmax layer to do the classification, fed the calculated values to it and trained this on the dataset.

As training progresses, step outputs are obtained, that is the accuracy and cross entropies of train and validation sets in the Tensorboard tool. After training is completed, the script runs a final evaluation of the test accuracy on a group of images which are kept aside from validation and train images. This test provided the best estimate of the model performance in doing the classification task.

### 4.3 WEB APPLICATION

The front-end web application is built using HTML, CSS and JS. Three HTML scripts were created; index, upload and display. Index is the home/landing page of the web application, upload is used to submit the end user's query image, and display is used to display the result. To integrate the backend model with the web application, we have used the Flask API.

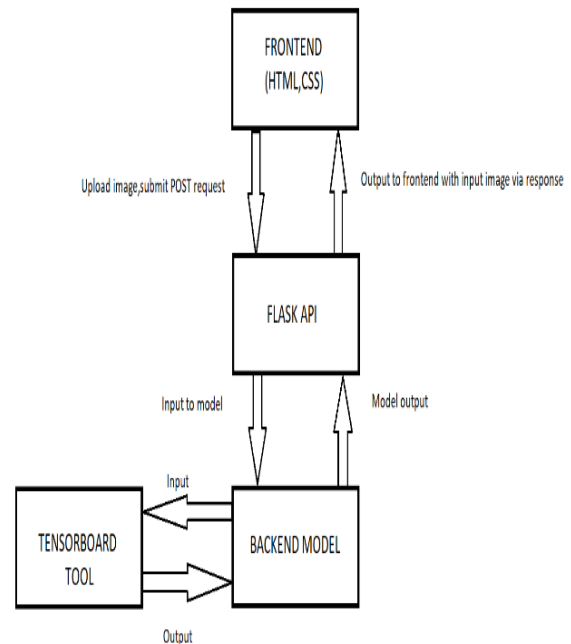


Fig 3 : Frontend-Backend Integration

This is a micro-framework based on Python, used to build an API which works with the AI model that is already trained. The user can upload an input image containing a landmark by first clicking on the upload button, which takes him to the upload page. There he can browse through query images, select the input image and can submit his image by clicking on the submit button. This action calls the backend model with the input image being sent in the message body via a post request by the Flask API to the backend. On clicking the display button, the result of the user query is displayed. Here, the API returns output via the response. The counter in the counter.txt file is incremented to keep track of user queries and the uploaded image is stored in the uploads folder. The output or result of the classification done by the model is also seen on the Terminal as a list of top five probabilities of classes that the image might belong to, sorted in decreasing order of their probabilities as shown in fig [4], where the class 'Colosseum rome' has the highest probability of 0.998, indicating that the input image 'colosseum\_italy.jpeg' belongs to that class. The first line of this list is extracted and sent to the frontend, via Post request, to be displayed as output.

```

Evaluation time (1-image): 0.934s
colosseum rome (score=0.99811)
leaning tower of pisa (score=0.00089)
gateway of india (score=0.00058)
charminar (score=0.00018)
hollywood sign (score=0.00008)
Counter:1
127.0.0.1 - - [28/May/2020 14:08:44] "POST /upload HTTP/1.1" 302 -
['uploads/1.jpeg']
['colosseum rome (score=0.99811)']
1
127.0.0.1 - - [28/May/2020 14:08:44] "GET /display HTTP/1.1" 200 -
127.0.0.1 - - [28/May/2020 14:08:44] "GET /static/uploads/1.jpeg HTTP/1.1" 200 -
    
```

Fig 4 : Backend inference result on the Terminal

### 5. RESULT

We have performed model training for 10000 epochs, with a learning rate of 0.005, using Nvidia GeForce 920M Intel i5 CPU, using Ubuntu Linux OS. For the image 'colosseum\_italy.jpeg' given as input in the Web Application, accurate output was generated, with the proper class label being assigned to input image, as seen in fig [5]. The trained model was also used for evaluating model performance. We have noticed that the model has achieved an accuracy of between 85% and 99% as seen in the graph generated in Tensorboard shown in the fig [6]. The blue line represents test accuracy and orange line represents the accuracy on the training set. The model was successfully able to recognize the landmark image features and classify them. We have also noticed that on increasing the number of epochs and decreasing the learning rate, better accuracy was obtained, but at the cost of time.



Fig 5 : Output generated for 'colosseum\_italy.jpeg'.

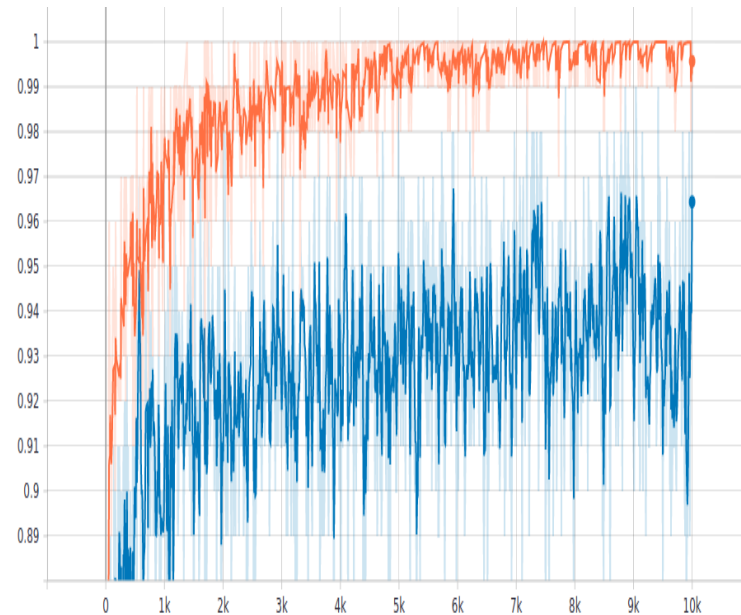


Fig 6 : Accuracy graph for 10000 epochs

### 6. CONCLUSIONS

Though our model has generated accurate results for our landmark images, there is still some scope for improving the efficiency and performance of the model, which can be done by further fine-tuning our model with regards to the model parameters. We can also increase the size of our dataset to accommodate more number of landmarks and images that would be used for our model training, and images of formats other than .jpeg.

In the future, we would like to increase our model's capability by being able to recognize and extract landmark features from videos as we believe this is a potential application for our model.

### ACKNOWLEDGEMENT

The authors would like to thank Mr. Manohar R., Assistant Professor, Dept. of Information Science and Engineering, Sir MVIT, Bengaluru, for his constant guidance and support.

### REFERENCES

- [1] Object Detection and Recognition in Images 1Sandeep Kumar, 2Aman Balyan, 3Manvi Chawla Computer Science &Engineering Department, Maharaja Surajmal Institute of Technology, New Delhi, India.-2017
- [2] Google Landmark Recognition using Transfer Learning - Project by Catherine McNabb, Anuraag Mohile, Avani Sharma, Evan David, Anisha Garg-2018
- [3] Alfredo Canziani and Eugenio Culurciello Weldon School of Biomedical Engineering, Purdue University, Adam Paszke Faculty of Mathematics, informatics and mechanics University of Warsaw. AN analysis of Deep Neural Network Models for practical applications.
- [4] Filippo Galli-Landmark Recognition with Deep Learning-2016