Landmark Detection Using Squeeze and Excitation Residual Neural Networks

M Uma1*, Anirudh Mugundhan 2 and Pooja2
1 Asst.Professor, SRM Institute of Science and Technology, Department of Software Engineering, Kattankulathur, Chennai, TamilNadu, India,
2 B.Tech Students, SRM Institute of Science and Technology, Department of Software Engineering, Kattankulathur, Chennai, TamilNadu, India.

* Corresponding Author: umaprabhu78@gmail.com

Abstract. Landmark Recognition is the technology that can anticipate landmark names straightforwardly from picture pixels, to help individuals better comprehend and sort out their photograph accumulations and for law enforcement officials to gauge the location of images submitted as evidence. Image classifications techniques have shown remarkable improvements over the last few years. To further improve computer vision technologies and methodologies, researchers are now concentrating on highly specific types of classification. Instead of classifying cats, cars or buildings, researchers are trying to classify among different types of landmarks - both natural and man-made. In the present age, a tremendous roadblock in landmark recognition research is the lack of large, well labelled datasets. To rectify this, Google has come up with the Google Landmark Recognition Dataset. The dataset contains 1.2 million images of 15000 categories of landmarks. For the project, a subset of Google Landmark Recognition dataset has been used. Various latest classification algorithms, like AlexNet, ResNet, SE-ResNet, VGG-16 and Inception v3 have been implemented to classify the images. Among them, the SE-ResNet architecture achieves the lowest loss value of 0.1822 and accuracy of 94.08% on the training set.

1. Introduction
In this paper, the task is to classify images based on the landmarks contained in them. For example, if the sample image is an image of the Taj Mahal, the output of the system should be Taj Mahal. For this task, Convolutional Neural Networks and its various architectures have been chosen. Previously, landmark recognition used to be done by using the GPS information present in the image itself. Without an active internet connection, this information will not be collected by the camera. The proposed system will be trained on the Google Landmark Dataset, which contains 1.2 million images of around 15000 types of landmarks. This system does not need the use of any internet connection to determine the location or the landmark present in the image. By running the trained model on the test image, the landmark present in the image is predicted. Also, this system will return results within 1-2 ms. For deciding upon the architecture to be used for the model, various research papers were read, and experiments were run on a subset of the dataset with the various papers architectures. The existing methodology relies on GPS information and the metadata present in the image for classifying images based on location and landmark. The disadvantage of this method is that without an internet connection while clicking the picture, this metadata does not exist. In such a case, classifying this image would require searching the Internet which can be a cumbersome process. Also, some not well known landmarks have still not been mapped, which results in a wrong classification. A method for classifying images using Deep Learning has been proposed, which does not require the use of any Internet connection. The system will be faster than existing systems and can be trained to learn images which it does not currently recognize too. The system will be tested on the dataset using various existing algorithms and the architecture which gives the best results will be chosen.
2. Literature Survey
Alex Krizhevsky et al [1] proposed a design made up of 5 convolutional layers pursued by 3 completely associated layers. AlexNet utilizes ReLU as the non-straight enactment work. An extra downside that this engineering settled was diminishing overfitting by methods for a Dropout layer following each totally associated layer. This engineering accomplished a best 5 test mistake rate of 15.3%, contrasted with 26.2% accomplished continuously best section on the ImageNet 2012 dataset. Karen Simonyan and Andrew Zisserman [2] proposed an improvement over AlexNet by substituting immense bit estimated channels with various 3X3 channels in a steady progression. Inside the gathering field, a lot of littler size pieces are prevalent than one huge size part since bunches of non-direct layers builds the profundity of system which enables it to find additional convolutions at a lesser expense. The convolutional layers are trailed by 3 completely associated layers. The width of the system begins at 64 and develops by a factor of 2 following each pooling layer. It achieves the main 5 exactness of 92.3% on ImageNet. Sergey et al [3] proposed the VGG design. It accomplishes a great exactness on ImagesNet dataset. GoogLeNet built up a segment considered commencement segment that gauges a little CNN with a normal design. Since just few neurons can be worked, the quantity of the convolutional channels of a particular piece estimate is kept up to a low number. To add to the current engineering, they utilize convolutions of various sizes to catch subtleties at changed scales. Additionally, it has a bottleneck layer(1X1 convolutions). It helps in the abatement of handling power. It accomplishes 93.3% top-5 precision on ImageNet and is a lot speedier than VGG. Kaiming He et al [4] proposed a remaining square which can be connected in the middle of the layers of a neural system. The profundity of a neural system would improve the exactness of the framework, if overfitting is dealt with. In any case, issue with developed profundity is that the qualities important to change the loads, end up being next to no at the underlying layers, because of the expanded profundity. Remaining systems license preparing of these systems by structure the system through segments named leftover squares. It achieves improved precision than VGG and GoogLeNet while utilizing less assets than VGG. ResNet-152 accomplishes 95.51% top-5 correctness. Jie Hu et al [5] proposed the utilization of another crush and excitation square to help in viably learning the assignment given close by. CNNs use channels to acquire information from pictures. Lower layers find minor subtleties like edges, while upper layers recognize appearances and content. This works by joining the three-dimensional and channel data of a picture. This implies adding a sole parameter to all channels and giving it a direct scalar to pass judgment on how pertinent every one is. These qualities would now be able to be utilized as loads on the highlights framework, positioning the channels dependent on its significance. SENet accomplishes 2.251% top-5 blunder rate in the ImageNet 2012 dataset. Barret Zoph et al [6] NASNet was fabricated utilizing the Neural Architecture Search system. The target of NAS is to utilize an information driven and insightful strategy to building the system plan rather than intuition and experimentations. In the Inception paper, it was exhibited that a muddled gathering of channels in a phone can impressively expand results. The NAS structure traces the structure of such a cell as a streamlining procedure, and afterward stacks the numerous duplicates of the best cell to construct a vast network. Finally, two unique cells are manufactured and used to prepare the full model. Gao Huang et al [7] proposed the DenseNet, made up of Dense squares, which is fundamentally a completely associated layer. That implies that the yield from past layers is completely associated with the contributions of the following layer. This outcomes is no loss of data from past layers. The Dense square is comprised of 3 sections a Batch Normalization task, trailed by a ReLU non-linearity and a convolutional channel of size 3*X. DenseNet does not require any wide or shallow layers since there is no loss of data from the information picture to the yield layer. This outcomes in an extensive number of complex computations to be registered, which builds the quantity of loads required for the model. This legitimately relates to an expansion in model size [8-10]. A DenseNet model is right around multiple times the span of a comparable ResNet model, both giving comparable correctness’s. Therefore, DenseNet is definitely not a suitable choice for lean, productive applications.

3. Data Source
The data source used is the Google-Landmarks dataset provided by Google. The dataset contains a CSV file with the filenames of the images along with a unique ID for each image. Each image can be downloaded by following the url provided in the CSV file. For the project, a Python script has been written to automatically download the dataset and skip any missing or duplicate images. The dataset contains 1.2 million images of 30,000 categories. The following image depicts the geographical distribution of the landmarks: URL for dataset- https://www.kaggle.com/c/landmark-recognition-challenge/data
4. System Architecture
The flow of the system starts from the CSV file. Using a download script, all the images are downloaded from the URL’s provided in the file. Next, the images are sorted into the various folders with respect to the landmark ID. Using another Python script, the images are resized into 128*128 pixels. Next, the images are split into a training and test dataset. The training images are fed into the model and the accuracy and loss values are monitored. The trained model is saved into a .h5 file for future testing. In the GUI, the user selects the image to be classified and clicks on the predict button. Using the saved model, the system predicts the landmark of the selected images. Fig1 shows architecture of Landmark Recognition.

![Architecture for Landmark Recognition System](image)

Figure 1. Architecture for Landmark Recognition System.

The following modules help to achieve the result.

4.1. Image Downloader
It is used to download the dataset used for training and testing the model. The links for the various images are provided in a CSV file. Using a python script, we automatically download the images.

4.2. Data Preprocessor
This module is used to preprocess the images for optimum training and testing. It involves extracting, resizing, and compressing the images. We use OpenCV to resize the images to a size of 128*128 pixels. Once resized, we use the Keras flow_from_directory API to load images in batches of 32 to the model.

4.3. Data Splitter
This is used to split the dataset into training and test sets. The ratio used is 80:20.

4.4. Machine Learning Model- Residual Block
Instead of hoping each few stacked layers directly fit a desired underlying mapping, we explicitly let these layers fit a residual mapping. Formally, denoting the desired underlying mapping as $H(x)$, we let the stacked nonlinear layers fit another mapping of $F(x) := H(x) - x$. The original mapping is recast into $F(x) + x$.

4.5. Machine Learning Model- shortcut connection
Short-cut connections are the connections which skip one or more layers. In the Fig an identity short-cut is added to the learned residual map $F(x)$ to obtain the desired mapping $H(x)$.

4.6. Machine Learning Model- squeeze and excitation block
The features of the neural network are passed through a squeeze operation, which results in a vector of size $n$, where $n$ is the number of channels. This vector is now passed to the excitation operation, which outputs the weights of the $n$ channels, which can be used on the original features to scale the channels according to their importance.
4.7. Prediction Model
After the model has been trained on the training set, it will be saved in a Hierarchical Data Format (HDF, .h5) file. For predicting the landmark in a given image, the saved model will be loaded and the test image will be fed to the model, which will output the landmark accuracy and loss shown Fig4 and Fig5.

![Image](image1.png)

**Figure 2.** User Screen before Prediction.

![Image](image2.png)

**Figure 3.** User Screen after Prediction.

The main component of the user interface will be the image uploader where the user will be able to upload any image of their choice after this, the system will run the prediction model on the given image to predict which landmark is present in the image. The image landmark before prediction and after the prediction model shown in Fig2 and Fig3.

5. Experiment Results & Analysis
To assess the execution of the SE-ResNet model, tests are performed on the Google Landmark Detection dataset. 300 classifications of tourist spots and 19,931 pictures have been decided for testing. The sum total of what pictures have been resized to a size of 128*128 pixels. Each picture is standardized utilizing mean RGB-channel subtraction. Streamlining is performed utilizing stochastic gradient descent, with a learning rate of 0.01 and force of 0.9. The model is prepared for six epochs.

The 5 algorithms used for testing are as follows-
- AlexNet
- ResNet
- SE-ResNet
- VGG-16
- Inception v3

![Image](image3.png)

**Figure 4.** Loss Values for each Algorithm.

As seen above, SE-ResNet gives the least loss value compared to every other algorithm tested. Eqn. (1) shows the loss formula being used is categorical cross entropy-

$$
\frac{\partial}{\partial \hat{y}_p} \left( - \log \left( \frac{e^{\hat{y}_p}}{\sum_j e^{\hat{y}_j}} \right) \right) = \left( \frac{e^{\hat{y}_p}}{\sum_j e^{\hat{y}_j}} - 1 \right)
$$

(1)
Accuracy function used for Keras is - K.mean(K.equal(K.argmax(y_true, axis=-1), K.argmax(y_pred, axis=-1))). Since SE-ResNet gives the highest accuracy amongst all the tested algorithms, we go ahead with SE-ResNet for the system.

5.1. Evaluation of Different Algorithm
All the algorithms were tested on a 25 class subset of the training set, where the 25 classes were randomly selected. The algorithms were trained on an i3-4030U CPU with no GPU acceleration, for 5 epochs.

Table 1. Comparison of Different Algorithm.

| Algorithm  | Loss  | Training Accuracy |
|------------|-------|-------------------|
| 1. ResNet  | 1.7205| 81.82%            |
| 2. VGG16   | 2.4551| 21.63%            |
| 3. SE-ResNet| 0.1822| 94.08%            |
| 4. AlexNet | 2.4608| 22.95%            |
| 5. Inception v3 | 1.1849 | 67.93%            |

For optimizing the neural network, Stochastic Gradient Descent was used, with a learning rate of 0.01, momentum of 0.9 and decay of 0.001. The output of Screen Shot SE-ResNet algorithm shown in Fig 7 and its loss and accuracy graph shown in Fig 8 and Fig 9.

5.2. Result Analysis
5.2.1. Confusion matrix. A confusion matrix is an arrangement of forecast aftereffects of a characterization problem. The number of right and wrong forecasts are included with check esteems and isolated by each class. The confusion matrix shows the purposes behind which the characterization model is defective when it makes predictions. It gives us meaning not just into the mistakes being made by a model however more essentially the classes of blunders that are occurring. Confusion matrix shown in Fig 6.
5.2.2. Recall. Review is the proportion of the number of accurately predicted positive values to the number of positive models. High Recall shows the class is accurately perceived.

Recall = TP/(TP+FN)

5.2.3. Precision. Precision is the proportion of the number of accurately predicted positive values to the number of predicted positive examples. High Precision shows a value predicted as positive is truly positive.

Precision = TP/TP+FP

For Class 1:

TP=19
FP=0
FN=2+3=5

Precision = TP/TP+FP = 19/(19+0)=1
Recall = TP/TP+FN = 19/(19+5)=19/24=0.7916
For Class 3: TP=72 FP=3 FN=5
Precision = TP/(TP+FP) = 72/(72+3)=0.96
Recall = TP/(TP+FN) = 72/(72+5)=0.935

6. Conclusions and Future Work
The objective of the proposed work was to tackle the task of recognizing landmarks from the Google Landmark Recognition Challenge dataset. Various state of the art CNN architectures were tested on a subset of 60 categories of landmarks from the given 15000 categories. From the experiments, it was discovered that Squeeze and Excitation Residual Neural Networks gives the highest accuracy of 94.02% and lowest loss value of 0.18 amongst all the algorithms. This application is a good alternative to GPS based categorization of images, as it does not require the use of the Internet. It can also be used by law enforcement officers to quickly gauge the location of pictures submitted for evidence. There are two major avenues for improvement which have not been explored in this project. One of them is the use of Transfer Learning. Instead of training all the layers of our neural network, one can copy or transfer the weights of several layers from a pre-trained model of the network, usually trained on another dataset (Network-based deep transfer learning). The leftover layers may be trained on the actual dataset, thus significantly reducing the training time. While this method may not lead to desirable results, it is worthwhile to try it out. The second method is to use the Deep Local Features (DELF) algorithm. Originally used for image retrieval, this algorithm is yet to be tested for image recognition. Additionally, the use of GPUs should improve the time taken for training and experimenting on the dataset.

7. References
[1] Krizhevsky A, Sutskever I and Hinton GE 2012 Imagenet classification with deep convolutional neural networks In Advances in neural information processing systems 1097-1105
[2] Simonyan K and Zisserman A 2014 Very deep convolutional networks for large-scale image recognition arXiv preprint arXiv:1409.1556
[3] Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V and Rabinovich A 2015 Going deeper with convolutions In Proceedings of the IEEE conference on computer vision and pattern recognition 1-9
[4] He K, Zhang X, Ren S and Sun J 2016 Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition 770-778
[5] Hu J, Shen L and Sun G 2017 Squeeze-and-excitation networks CoRR abs/1709.01507 arXiv preprint arXiv:1709.01507
[6] Zoph B, Vasudevan V, Shlens J and Le QV 2018 Learning transferable architectures for scalable image recognition In Proceedings of the IEEE conference on computer vision and pattern recognition 8697-8710
[7] Huang G, Liu Z, Van Der Maaten L and Weinberger KQ 2017 Densely connected convolutional networks In Proceedings of the IEEE conference on computer vision and pattern recognition 4700-4708
[8] Sutskever I, Martens J, Dahl G and Hinton G 2013 On the importance of initialization and momentum in deep learning International conference on machine learning 1139-1147
[9] Tan C, Sun F, Kong T, Zhang W, Yang C and Liu C 2018 A survey on deep transfer learning In International conference on artificial neural networks pp270-279 Springer Cham
[10] Noh H, Araujo A, Sim J and Han B 2016 Image retrieval with deep local features and attention-based keypoints arXiv preprint arXiv:1612.06321