A Comparative Study of Different CNN Models in City Detection Using Landmark Images

Chapter · July 2019

DOI: 10.1007/978-981-13-9181-1_48

5 authors, including:

Masum Shah Junayed
University of Connecticut
37 PUBLICATIONS **137** CITATIONS
SEE PROFILE

Afsana Ahsan Jeny
Wayne State University
25 PUBLICATIONS **98** CITATIONS
SEE PROFILE

Nafis Neehal
Islamic University of Technology
20 PUBLICATIONS **119** CITATIONS
SEE PROFILE

Syeda Tanjila Atik
Daffodil International University
21 PUBLICATIONS **182** CITATIONS
SEE PROFILE

A Comparative Study of Different CNN Models in City Detection Using Landmark Images

Masum Shah Junayed, Afsana Ahsan Jeny(✉), Nafis Neehal, Syeda Tanjila Atik, and Syed Akhter Hossain

Daffodil International University, Dhaka 1207, Bangladesh
{junayed15-5008,ahsan15-5278,nafis.cse,syeda.cse}@diu.edu.bd, aktarhossain@daffodilvarsity.edu.bd

Abstract. Navigation assistance using different local Landmarks is an emerging research field now-a-days. Landmark images taken from different camera angles are being vividly used alongside the GPS (Global Positioning System) data to determine the location of the user and help user with navigation. However, determining the location of the user by recognizing the landmarks from different images, without the help of GPS, can be a worthy research trend to explore. Hence, in this paper, we have conducted a comparative study of 3 different popular CNN models, namely - Inception V3, MobileNet and ResNet50, and they have achieved an overall accuracy of 99.7%, 99.5% and 99.7% respectively while determining cities using landmark images.

Keywords: City detection · Landmark · Inception · ResNet50 · MobileNet · CNN

1 Introduction

A landmark is a recognizable physical or artificial characteristic used for navigation, a feature that stands out from its near circumstance and is often able to see from long distances. Landmarks are nationally momentous historic places denominated by the Secretary of the Interior because they occupy exceptional standard or quality in explaining or interpreting the tradition of the cities.

Landmark is an official determination that a possession is of significance to the people, the kingdom, or the topical community. To increase the community's wariness and pride in its past, it is this "learning of place" that motivates people to put down roots in a society. It supports to ennoble the ocular and aesthetic nature, beauty and rareness of the city. It also helps safeguard the city's generation and tradition, stabilize and raise possession values and enhance the city's fascination for occupants, the visitors, tourists, and expected residents. Historical and cultural travelers expend much more than other travelers do.

Actually, it is one of the main components that shape the picture and figure of historic cities. It is manifest that recent and improper developments have diluted

K. C. Santosh and R. S. Hegadi (Eds.): RTIP2R 2018, CCIS 1035, pp. 547–560, 2019.
https://doi.org/10.1007/978-981-13-9181-1_48

the precedence of landmarks. For landmarks recognition, the researcher already did so many works like topological navigation of mobile robots for landmark identification using the 2D pattern search engine [10], using Iconic Scene Graphs for recognizing landmark images [21].

The city's historical landmarks have always been exoteric among tourists because of its historical values and also observed as components of reference. Actually, we choose landmarks for city identification for the landmark's popularity. When people want to visit different countries to see landmark images, sometimes they can't detect these landmark places. And the other country's children have the same problem. That's why we select this idea for our paper so that we can solve this problem and remove the city identification problem without the help of GPS (Global Positioning System).

In this paper, we have used 3 different popular CNN [6] models. They are Inception-v3 [2], MobileNet [7] and Resnet50 [9]. These are pre-trained [3] models. We have used 6 different landmarks images of 6 different cities which are Taj Mahal for Agra, Burj Khalifa for Dubai, Pyramid for Giza, Statue of Liberty for NewYork, Eiffel Tower for Paris and finally Opera House for Sydney. By using a little training time we fulfilled a useful city detection model and obtain a higher accuracy from 3 different models. The remaining paper is arranged in the following manner: Details of Convolutional Neural Network (CNN) [6], Inception-v3 [2], MobileNet [7] and Restnet50 [9] model are discussed in Sect. 2. The comparison with other papers is discussed in Sect. 3. Data collection, model installation, and training are discussed in Sect. 4. Performance analysis is done in Sect. 5. Finally conclusion with some future work scopes is described in Sect. 6 and Sect. 7.

2 Background Study

Our paper is based on the Inception-v3 [2] model and MobileNet [7] of TensorFlow [4] platform and Resnet50 [9] model of Convolutional Neural Network [6].

TensorFlow [4] is an open-source software library for high-performance analytical calculations. Its flexible architecture allows easy deployment of various platforms (CPUs, GPUs), and desktop clusters from the desktop to mobile and end devices. Originally developed by Google's brain team of researchers and engineers in Google's AI organization, it is used across many other scientific domains for strong support and flexible numerical calculation for machine learning and deep education. It is malleable and has been used for amplifying machine learning sections of computer science which including recognition of speech, computer vision related works, natural language processing and so on.

Convolutional neural network (ConvNet or CNN) [6] is a class of deep artificial neural network that has been successfully implemented in visual image analysis. The following layers are used to create the ConvNet architectures:

Convolutional Layers: An example of the Convolutional Layer. Note that there are multiple neurons (5 in this example) with depth, all inputs are connected to the same local area. Convolutional layers CNN's [6] main part. Its output volume can be interpreted as descending neurons in a 3D volume. And

its parameters are able to learn the filter set. Each filter is only connected to the input volume in a local area (width and height), but with full depth [6].

During the forward pass, each filter is applied to a small local area, computing dot production between filters and input. Then it iterates across the width of the input volume and height. The result of each filter is a 2-dimensional activation map. In this way, they start to learn some filters on the network when they show some local location of input with certain specifications.

Pooling Layers: Pooling layers another significant part of CNN [6]. It is a form of nonlinear sub-sampling. The input image outputs the pond partitions in a set of non-overlapping rectangles and outputs maximum for each sub-region. Insight is that once a feature is found, its exact position is not as important as its rough position compared to other features [6].

The layer of pooling significantly reduces input size and network parameters and counting numbers significantly. As a result, overfitting will not be a serious problem (Fig. 1).

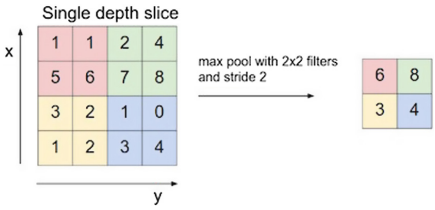

Fig. 1. An example of (max) pooling layer.

ReLU Layers: Rectified linear unit (ReLU) layers apply to non-saturating activation function $f(x) = maximum(0, x)$. The release layers increase the overall architecture's linearity, without affecting the acceptable areas of the coverage layer [6].

Other types of activation functions increase the linearity, such as the sigmoid function $f(x) = (1 + e^{-x})^{-1}$. However, RELU accuracy should be faster training without significant losses. So ReLU is used more on CNN [6].

Fully-connected layers: Fully-connected layers are usually the last layer of the overall architecture. They have a full connection to their previous layers, the same as the regular layers in the normal CNN [6] (Fig. 2).

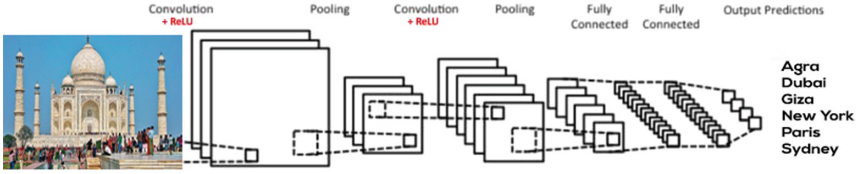

Fig. 2. Structure of Convolutional Neural Network (CNN).

Inception-v3 [2] is one of the TensorFlow [4] training models. This was followed by the re-review of the computer's start-up structure in 2015 after the Inception-v1, Inception-v2. Inception-v3 [2] model is trained in the ImageNet dataset, among which 1000 consist of two parts of the class Inception-v3 [2]: A convolutional neural network (CNN) [6] and the exhaustive parts of the properties with fully connected and softmax layers with classification parts [2,5].

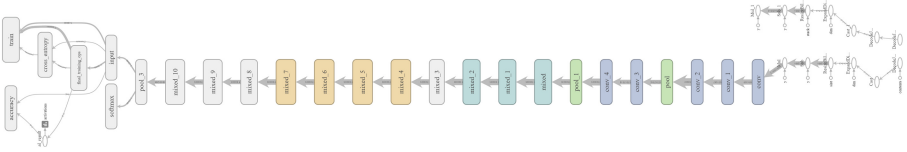

Fig. 3. The architecture on Inception V3 of our dataset.

MobileNet [7] is a small effective convolutional neural network designed by researchers at Google. Limit two parameters in the MobileNets [7] network, which can tune to stop the trade/accuracy of the exact problem: a multiplier of width multiplier and resolution. Width multiplier allows slimming on the network, when the multiplier of the resolution changes the input level of the image, reducing the internal representation of each layer.

We have used 224 1.0 version for our code lab. Here 1.0 is the width multiplier and it can be 1.0, 0.75, 0.50 or 0.25. The 224 is the image resolution and it can be 224, 192, 160 or 128. Here the smallest version of MobileNet [7] is 128 0.25 and the biggest version is 224 1.0. Though 224 is the higher resolution image and takes more processing time but it provides better classification accuracy.

MobileNet [7] is based on a streamlined architecture that can be divided deeply to create a lightweight deep neural network. Here present two common global hyper-parameters that effectively control the latency and accuracy. These hyper-parameter models allow developers to select the right size model for their application based on the limit of the problem (Fig. 4).

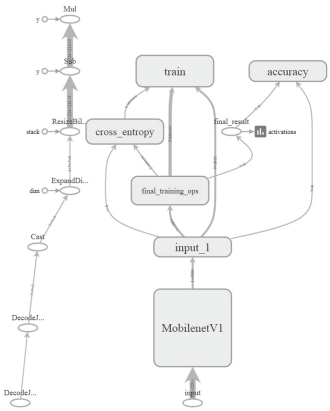

Fig. 4. The architecture on MobileNet of our dataset.

ResNet [9] is a short name of Residual Network which is a special type of neural network which helps us to handle more powerful deep learning tasks and models through a network with many layers. In 2012, the AlexNet had come which contain 8 layers and the error of 16.4%. In 2014 and 2015, the VGG16, VGG19, and GoogleNet model had come with 16 layers, 19 layers, and 22 layers respectively. And finally, the ResNet model has come with 152 layers which are also a top-5 classification error of 3.57%.

The main idea of Resnet [9] is introduced with a shortcut connection which is called identity shortcut connection or identity block that can skip one or more layers. Mathematically, A ResNet [9] layer approximately calculates $y = f(x) + id(x) = f(x) + x$. The gradients can easily be turned back and those shortcuts act like highways and the result can be found out so much faster with much more layers (Fig. 5).

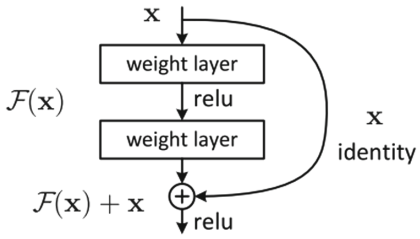

Fig. 5. A residual block of ResNet.

In our paper, we have used a deep CNN model based on the Residual Network architecture with 50 layers which termed as ResNet50 [9]. For implementing this model, we have used keras [1]. Keras [1] is a high-level neural network which is written in python and able to run on the top of TensorFlow [4]. ResNet50 [9] is much deeper than the VGG16 and VGG19. Though it has 50 layers but its size is small because of global average pooling and fully-connected layers. As a result of the model size down to 102 MB (Fig. 6).

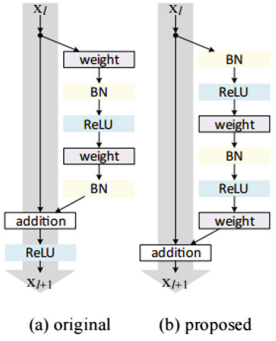

Fig. 6. (Left) the original residual module. (Right) The updated residual module using pre-activation.

3 Literature Review

The following tasks that were used by the Inception-v3 [2], MobileNet [7] and ResNet50 [9].

In 2015, He, Zhang, Ren, and Sun used deep Residual Learning for their image classification. They clearly described the different types of Resnet [9] model which is very important for all. They also discussed how the Resnet [9] had come with many layers and discussed these models accuracy [18].

In 2016, Elizalde, Chao, Zeng, Lane from Electrical and Computer Engineering Carnegie Mellon University Mountain View exposed a paper on City-Identification of Flickr Videos Using Semantic Acoustic Features. It was based on the UrbanSound8K set containing audio clips labeled by sound type. They showed to what extent the city-location of videos correlates with their acoustic information. But no CNN [6] models were used here [13].

In 2009, Li Crandall Huttenlocher expressed a paper on Landmark Classification in Large-scale Image Collections. In this paper, they study image classification on a much larger dataset of 30 million images and used a Support Vector Machine model and the accuracy was 53.58% [11].

In 2009, Zheng, Zhao, Song, Adam exposed a paper on Tour the World: building a web-scale landmark recognition engine. The resulting landmark recognition engine incorporates 5312 landmarks from 1259 cities in 144 countries and the accuracy was 80.8% [12].

In 2017, Gavai, Jakhade, Tribhuvan and Bhattad used MobileNets for Flower Classification using TensorFlow on the flower category datasets of Oxford-I7 and Oxford-102 for Flower Classification. The accuracy of 1.0 MobileNet-224 was 70.6% and 0.5 MobileNet-160 was 60.2% [14].

In 2017, Kim, Choi, Jang and Lim used Convolutional Neural Network [6] model such as Inception, ResNet, and MobileNet on Driver Distraction Detection which was pre-trained with the ILSVRC2012 dataset. Their accuracy was increased when they used MobileNet rather than Inception and ResNet [15].

Our experiment is also based on Inception-v3 [2], MobileNet [7] and ResNet50 [9] model of Convolutional Neural Network [6] for city detection using landmark images. City identification has not been done by using landmark images before. This is the first approach from us. And those model of CNN [6] has worked well in our experiment and also given a high accuracy.

4 Methodology

In this section, the list of items to be discussed are as follows: first of all we make a flowchart [16] which explains the process of our experiment; second, we provide a simple introduction to our dataset; third, we provide the data preprocessing of our experiment; then, we discuss the model installation; last of all, we introduce the train model.

A flowchart [16] is a type of diagram that represents an algorithm very easily. Here, the following flowchart Fig. 7 illustrates a solution model to our system.

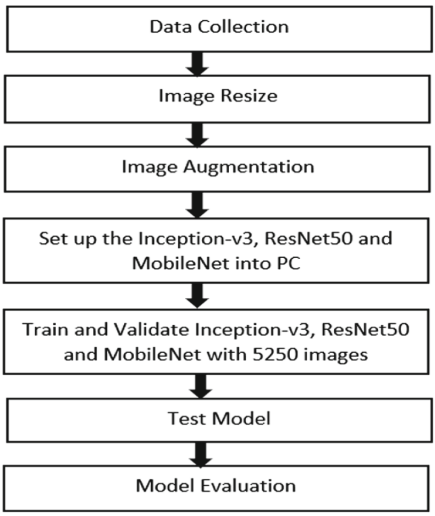

Fig. 7. Flowchart of our experiment.

4.1 Dataset

For our experiment, we have collected 900 images on six different landmarks of six different cities. The six landmarks are Taj Mahal for Agra, Burj Khalifa for Dubai, Pyramid for Giza, Statue of Liberty for New York, Eiffel Tower for Paris and finally Opera House for Sydney.

4.2 Data Preprocessing

In data preprocessing, we have artificially expanded the dataset for avoiding overfitting. This data will make a few variances that can happen when someone else takes refresh data from the web or in real life. After collecting data for each class we have augmented the dataset in 5 different methods. They are Rotate +30, Rotate -30, Translation, Shearing, and Flip.

Finally, we have found 5400 images for training from this augmentation. It is very difficult to display all data. So we display four images for each class in the following Fig. 8.

Fig. 8. The example of our dataset.

4.3 Model Installation

This experiment is based on the MobileNet [7] and Inception-v3 [3] model of TensorFlow [4] platform and ResNet50 [9] which is Keras [1] application. The processor is 2 GHz Intel i3, memory 4 GB 1600 MHz DDR3, System type: 64-bit Operating System, x-64 based processor.

First of all, we have downloaded TensorFlow [4]. Then we have also installed MobileNet [7] and Inception-v3 [2] model. Then we have used Keras [1] for ResNet50 [9].

4.4 Train Model

Inception-v3 [2], MobileNet [7] and Resnet50 [9] are deep neural network models that's why it is very difficult to train in a low-level configuration computer. Inception-v3 [2] takes one day, MobileNet [7] takes 6–8 h and Resnet50 [9] takes two days for training.

Tensorflow [4] offers a tutorial on how to rearrange the final layer of the installation of a new class using transfer learning. We use the transfer learning method that keeps the previous layer parameters and remove the last layer of the Inception-v3 [2] model, then try again at the end layer. The number of output nodes in the last layer is equal to the number of classes in the dataset. Fig. 3 is the Inception-v3 [2] architecture of our model.

Then the bottleneck files are generated. After finishing this, the original training begins of the final layer of the network. Our script has run 4000 training steps and each step have selected 10 images randomly from the training set and have found their bottlenecks from the cache. To get prediction images have fed into the final layer. That prediction is then compared against the actual label, and the results of this comparison are used to update the final layer weight through a backpropagation process. At the time of training the training accuracy,

validation accuracy, and the cross-entropy graph are generated. After completing all the training steps, the script has run for the evaluation of final test accuracy. The finally the accuracy have been generated and have shown a value of accuracy. This number indicates the percentage of images in the standard test set that the model is labeled as perfect after being fully trained.

All of these things have happened in the time of Mobilenet [7].

Resnet50 [9] is the model of Residual Networks which using identity blocks for shortcut connection. This block names bottleneck blocks and it follows two rules. One if the map has the same output characteristics then the layers have the same number of filters and two if the map is half then the filters are doubled. Down-sampling is composed directly by the convolution layer, which is an extension 2 and normalization of the batch takes place exactly before each convolution and RELU activation. When input and output levels are the same, identity shortcut is conducted. If the dimension is enhanced, then projection shortcut 1×1 is conducted to become level like convolutions. We have shifted the first 49 layers of ResNet-50 [9] by using the Transfer Learning Language. 1000 fully connected (fc) layer in the network ends with the softmax activation. Using bottleneck features of our passport cover images as input, we train 6 fully connected softmax, since we have 6 classes and our trained 6 fully connected software replaces 1,000 fully connected scams.

We have also made three Confusion Matrixes [8] for three different models. From Confusion Matrix [8], we have calculated Precision, Recall, Accuracy, and F1-Score [17]. And finally, we have calculated Macro Average Accuracy of three CNN [6] models for our experiment.

It is very difficult to give three Confusion Matrixes [8]. That's why we have given one Confusion Matrix [8] of ResNet50 [9] model. From the following Confusion matrix [8] of Table 1, we can say that our model has provided a very high number of True Positive values.

Table 1. Confusion matrix

	Agra	Dubai	Giza	Newyork	Paris	Sydney
Agra	25	0	0	0	0	0
Dubai	0	24	0	0	1	0
Giza	0	0	25	0	0	0
Newyork	0	0	0	25	0	0
Paris	0	0	0	0	25	0
Sydney	0	0	0	0	0	25

5 Result Analysis

The following figures are expressing the development of the model's accuracy and cross-entropy as it is trained. Figures 9 and 11 show the training progress (x-axis) as a process of the accuracy (y-axis). Again Figs. 10 and 12 show the training

progress (x-axis) as a process of the cross-entropy (y-axis). Here the orange line represents the training set, and the blue line represents the validation set for Figs. 9, 10, 11 and 12.

Now overfitting happens when a model fits too well and when the training accuracy will be higher than the accuracy on the validation set. So we can say that little overfitting occurs among Figs. 9, 10, 11, 12, 13 and 14. Because among those figures, the training set is higher than the validation set.

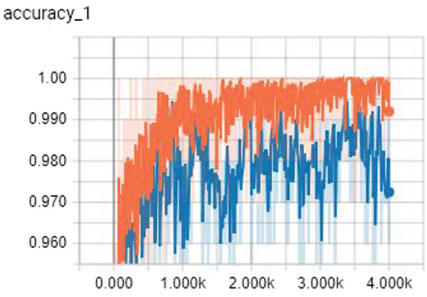

Fig. 9. The variation of accuracy on the training dataset of Inception-v3.

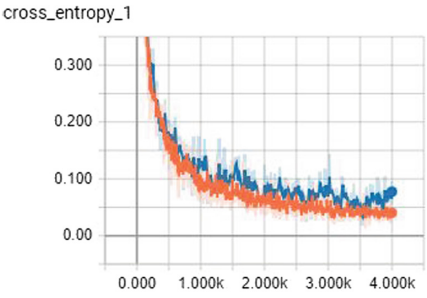

Fig. 10. The variation of cross entropy on the training dataset of Inception-v3.

Table 2. Description of the two figures.

	Index	Performance
Dataset	The training set accuracy	99.7%
	The validation set accuracy	99.1%-99.4%
	The training set cross-entropy	0.03
	The validation set cross-entropy	0.05

Table 2 shows the description of the two figures. For our dataset, the training accuracy can reach 99.7%, and the validation accuracy can be maintained at 99.1%–99.4%.

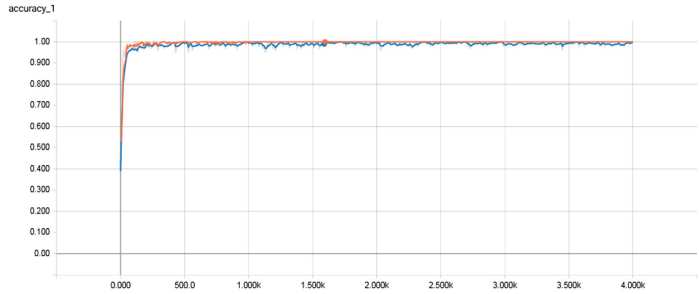

Fig. 11. Accuracy with respect to training and validation of MobileNet.

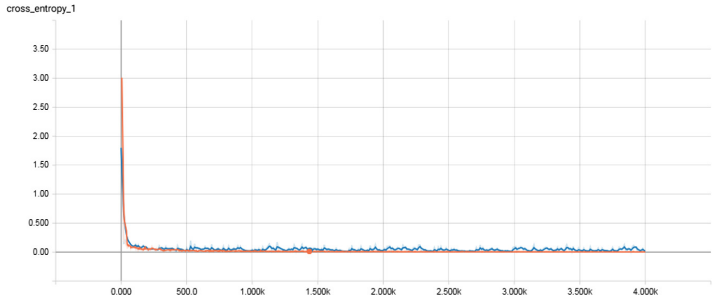

Fig. 12. Cross entropy with respect to training and validation of MobileNet.

Now Fig. 13 shows the number of epoch's progress (x-axis) as a process of the loss (y-axis). Again Fig. 14 shows the number of epoch's progress (x-axis) as a process of the accuracy (y-axis). Here the orange line represents the performance of the validation set and the blue line represents the performance of the training set in Fig. 13. On the other hand, the green line represents the performance of the validation set and the blue line represents the performance of the training set in Fig. 14 [19].

According to these figures, there shows a stability after the 8 epochs and the Resnet50 [9] model has given a high accuracy which is 99.7%.

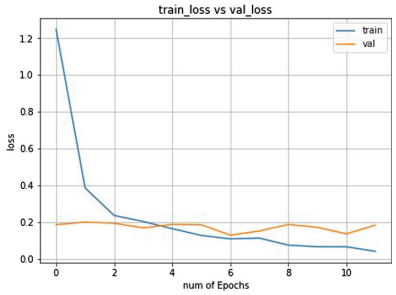

Fig. 13. Validation loss curve by using pre-trained model ResNet-50.

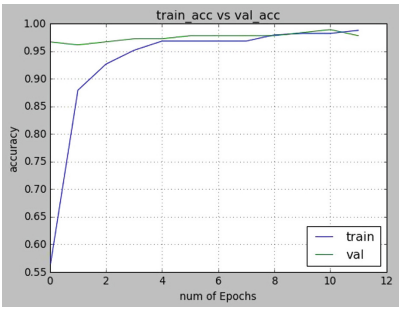

Fig. 14. Validation accuracy curve by using pre-trained model ResNet-50.

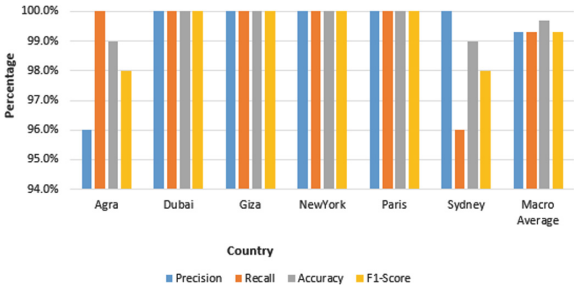

Fig. 15. Precision, Recall, Accuracy and F1-Score graph of Inception-v3.

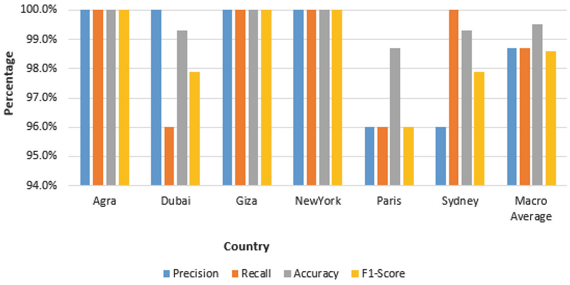

Fig. 16. Precision, Recall, Accuracy and F1-Score graph of MobileNet.

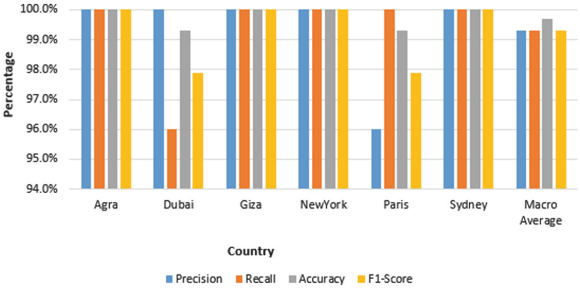

Fig. 17. Precision, Recall, Accuracy and F1-Score graph of ResNet50.

Figures 15, 16 and 17 show the precision, recall, accuracy and F1-Score [17] graph of Agra, Dubai, Giza, NewYork, Paris, and Sydney and also show the macro average [20].

Table 3. The accuracy of six classes and final accuracy of Inception-v3, MobileNet and ResNet50.

	Inception-v3	MobileNet	ResNet-50
City	Accuracy	Accuracy	Accuracy
Agra	99%	100%	100%
Dubai	100%	99.3%	99.3%
Giza	100%	100%	100%
Newyork	100%	100%	100%
Paris	100%	98.7%	99.3%
Sydney	99%	99.3%	100%
Macro Average	99.7%	99.5%	99.7%

Table 3 show the final accuracy of Inception-v3 [2] is 99.7%, MobileNet [7] is 99.5% and finally ResNet50 [9] is 99.7%.

6 Future Work

As we have overfitting problem in our models. In future for removing this overfitting, we want to add more data and regularization, use other augmentation methods and architectures and also reduce architecture complexity. The Inception-v3 [2], MobileNet [7], and ResNet50 [9] model of CNN [6] which is already generated and we have used it. So, our future work is also to study and make a new model so that we can use that model and can also obtain a high accuracy.

7 Conclusion

In our paper, we have represented three models of Convolutional Neural Network (CNN) [6] for city detection using landmark images. In Inception-v3 [2], we have used 21 million parameters, the loss and accuracy we have got from this model is 0.3% and 99.7% and the resulting model size was 96 MB. Then in MobileNet [7], we have used 4.24 million parameters, the loss and accuracy we have got from this model is 0.5% and 99.5% and the resulting model size was only 17 MB. And lastly in ResNet50 [9], we have used 23.5 million parameters, the loss and accuracy we have got from this model is 0.3% and 99.7% and the resulting model size was only 102 MB. Now we can say that MobileNet [7] has done better than the other two models because it is small in size and so much faster to run from others. We have tried to highlight the procedure and the performance of three models in a short time. Hopefully, our approach will be helped in the future.

References

1. https://en.wikipedia.org/wiki/Keras
2. https://arxiv.org/pdf/1512.00567.pdf
3. https://www.kaggle.com/keras/resnet50/home
4. https://arxiv.org/abs/1603.04467
5. Xia, X., Xu, C.: Inception-v3 for flower classification. In: 2017 2nd International Conference on Image, Vision and Computing (2017)
6. http://wiki.ubc.ca/Course:CPSC522/Convolutional_Neural_Networks#cite_note-wiki-3
7. https://arxiv.org/abs/1704.04861
8. https://en.wikipedia.org/wiki/Confusion_matrix
9. https://www.pyimagesearch.com/2017/03/20/imagenet-vggnet-resnet-inception-xception-keras/
10. Mata, M., Armingol, J.M., de la Escalera, A., Salichs, M.A.: A visual landmark recognition system for topological navigation of mobile robots. In: Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No. 01CH37164, 21–26 May 2001
11. Li, Y., Crandall, D.J., Huttenlocher, D.P.: Landmark classification in large-scale image collections. In: 2009 IEEE 12th International Conference on Computer Vision, 29 September 2009–2 October 2009
12. Zheng, Y.-T., Zhao, M., Song, Y., Adam, H.: Tour the world: building a web-scale landmark recognition engine. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition, 20–25 June 2009
13. Elizalde, B., Chao, G.-L., Zeng, M., Lane, I.: City-identification of flickr videos using semantic acoustic features. arXiv: 1607.03257v1 [cs.MM], 12 July 2016
14. Gavai, N.R., Jakhade, Y.A., Tribhuvan, S.A., Bhattad, R.: MobileNets for flower classification using tensorflow. In: 2017 International Conference on Big Data, IoT and Data Science (BID), 20–22 December 2017. Vishwakarma Institute of Technology, Pune (2017)
15. Kim, W., Choi, H.-K., Jang, B.-T., Lim, J.: Driver distraction detection using single convolutional neural network. In: 2017 International Conference on Information and Communication Technology Convergence (ICTC), 18–20 October 2017
16. https://en.wikipedia.org/wiki/Flowchart
17. https://en.wikipedia.org/wiki/Precision_and_recall
18. He, K., Zhang, X., Ren, S., Sun, J: Deep residual learning for image recognition. arXiv: 1512.03385v1 [cs.CV], 10 December 2015
19. https://github.com/keras-team/keras/issues/3755
20. https://datascience.stackexchange.com/questions/15989/micro-average-vs-macro-average-performance-in-a-multiclass-classification
21. Li, X., Wu, C., Zach, C., Lazebnik, S., Frahm, J.-M.: Using iconic scene graphs for modeling and recognition of landmark images collections, 16 April 2011