| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | Impact Factor: 7.488 |

|| Volume 8, Issue 6, June 2020 ||

Distributed Deep Learning Framework for Image Similarity-Based House Recommendation Using BIGDL

Akshaykumar I Gutte[1], Prof. Vipin K. Wani[2], Prof. Narendra Joshi[3], Mr. Aditya K Sinha[4]

Department of Computer Science and Engineering, SOCSE, SandipUniversity, Nashik, India[1]

Assistant Professor, Department of Computer Sciences and Engineering, SOCSE, SandipUniversity, Nashik, India[2]

Assistant Professor, Department of Computer Sciences and Engineering, SOCSE, SandipUniversity, Nashik, India[3]

Associate Director, HOD-ACTS , CDAC, Pune, India[4]

ABSTRACT*:* Agents upload thousands of images from which customers have to pick their favorite one. We propose a house recommendation system that will recommend a similar kind of house to the user with their favorite one. , the recommendation system is designed to play a role in the home buying experience using semantic similarity model and visual semantic model among millions of house images. In this survey paper, the study has been carried out on image-based house recommendations using Intel's BIGDL and VGG16. Users can upload any house images or select a listing photo in the database and have the system recommend listings of similar visual characteristics that may be of interest. Recommend houses based on title image characteristics and similarities. Most title pictures are front exterior, while others can be a representative image for the house. The study is paving the road for innovation in advanced analytics applications for the real estate industry.

KEYWORDS: deep learning, convolutional neural network, vgg16.

I. INTRODUCTION

An enormous number of issues in the computer vision space can be tackled by ranking pictures according to their similarities. For example, e-retailers show clients items that are comparable things from past buys, to sell more on the web. For all intents and purposes, each industry considers this to be a game-changer, including the real estate business, as it has gotten progressively digital over the previous decade. In excess of 90 percent of homebuyers search online during the time spent looking for a property. Homeowners and realtors give data on house qualities, for example, area, size, and age, just as numerous inside and outside photographs for real estate posting look. However, because of specialized limitations, the huge amount of data in the photographs can't be extracted and filed to improve the search or serve land posting results. In fact, “show me a comparable home” is a top list of things to get demand among clients. Over the most recent three years, mainly because of the advances of deep learning, more solidly convolutional neural systems, the nature of image recognition and object detection have been advancing at a dramatic pace. So with the help of deep learning and BIGDL, we can build an answer for individuals who are interested to purchase a house.
A recommendation system is a system that used to predict the "rating" or "preference" a user would give to an object, in the recent years, mainly due to the progress of deep learning and convolution neural networks, the quality of image similarity or classification has been progressing. With the help of deep learning and the BIGDL library, we can be built a solution for people who are seeking to buy a house. The goal of House recommendation system is to find "similar" images of a house.
There are two kinds of image similarities

1) Semantic Similarity
2) Visual Similarity

A semantic similarity implies the pictures can have a similar class of elements or objects.
Visual similarity implies picture might not have the elements or object classifications yet decides how pictures will look each other from a visual standpoint, for instance, an apartment picture and a conventional house picture might be very comparative

As a developing distributed deep learning structure, BigDL gives simple and integrated deep learning abilities for enormous information networks. With a rich arrangement of help for deep learning applications, BigDL allows users to compose their deep learning applications as standard Spark programs, which can directly run over existing Apache Spark or Apache Hadoop* cluster

II. TECHNOLOGIES USED IN THE PROJECT

A. Convolutional Neural Network (CNN)

A Convolutional Neural Network (CNN) is a Deep Learning architecture that can take an input image, offer significance to different highlights in the image and separate one image from the other picture. VGG16 is a convolution neural net (CNN) design which was introduced in ILSVR (Imagenet) rivalry in 2014. It is considered as one of the best vision model design till date. The most exceptional thing about VGG16 is that as opposed to having an enormous number of hyper-parameter they concentrated on having convolution layers of 3x3 filters with a stride 1 and always used the same padding and maxpool layer of 2x2 filter of stride 2. It follows arrangement of convolution and pool layers reliably all through the entire architecture. At last, it has 2 FC (fully connected layers) followed by a softmax for output. The 16 in VGG16 refers to it has 16 layers that have weights. This system is a really huge system and it has around 138 million (approx) parameters.

B. Analytics zoo

Analytics Zoo gives a bound together data analytics and AI stage that joins TensorFlow, Keras, PyTorch, Spark and BigDL programs into a coordinated pipeline, which can straightforwardly scale from a PC to huge clusters to process producing large information.

C. BigDL

BigDL is a DDL(distributed deep learning library) for Apache Spark; with BigDL, customers can form their applications as standard Spark programs, which can legitimately run over existing Hadoop clusters or Spark.. To make it simple to construct Spark and BigDL applications, an Analytics Zoo library is given to end-to-end analytics + AI pipelines.

III. PROPOSED WORK

A. SOLUTION WITH BIGDL

To recommend houses based on image similarity, we first compare the query image of the selected listing photo with the title images of candidate houses. Next, a similarity score for every house is generated. Only the top outcomes are picked based on ranking. In this project, both semantic similarity and visual similarity were utilized. BigDL gives a rich set of functionalities to help training or image models, including:

• Providing useful image transformers based on Apache Spark and OpenCV* for image processing on Spark.
• Providing helpful model fine-tuning support and a adaptable programming interface for model modification.
• Users can load pretrained Caffe*, Torch* or TensorFlow* models into BigDL for inference.

B. SEMANTIC SIMILARITY

For semantic similarity, three image classification models are required in the project.

Model 1. Image classification: Determines whether the house front is exterior. We have to recognize if the title picture is or isn't the house front. The model is fine-tuned from pretrained GoogLeNet v1 on the dataset. We utilized the Places dataset for the training.

Model 2. Image classification: House style (contemporary, ranch, traditional, Spanish). Similar to 1,the model is fine-tuned from pretrained GoogLeNet v1 on the Places dataset..

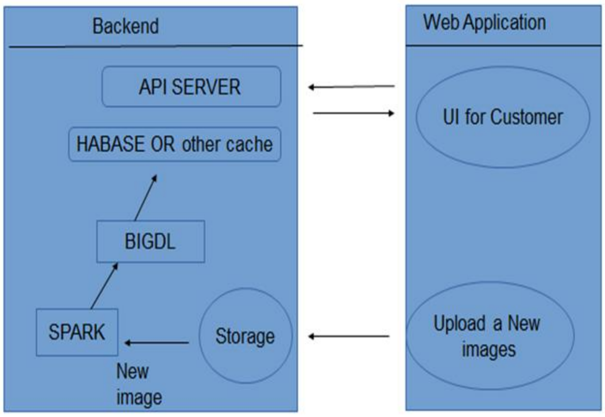
Model 3. Image classification: House story (single story, two story, at least three stories). Similar to 1, the model is fine-tuned from pretrained GoogLeNet v1 on the Places dataset.

C. VISUAL SIMILARITY

We have to compute visual similarity to determine a ranking score. For each query, the user will input an image for comparison against the large number of candidate images. To meet the latency requirement, we calculated a direct comparison against precalculated features from images. We first built an evaluation dataset to pick the best alternatives

for image similarity calculation. In the evaluation dataset, each record contains three images. Triplet (query image, positive picture, negative pict
ure),where positive picture is more likely to the query image. For each record, we can evaluate different similarity functions.

D. PROPOSED ARCHITECTURE



The project can be separated into three parts:

1. *Model training (offline)*
The model training primarily refers to the semantic models which is the backend of the proposed system and also finding best possible embedding for visual similarity estimation. Retraining may happen periodically relying upon model execution.

2. *Image inference (online)*
With the trained semantic models in the initial step and the pretrained VGG-16, we can change over the images to labels and embeddings, and save the results in a key-value cache.All the current images and new images need to go through the inference above and changed over into a table structure. For that we have to create a real estate website.

3. *API serving for query (online)*
We can create an android or Ios application using the backend system .

IV. RESULT

1. Following code gives the accuracy of model

```
# evaluate the model
print("Start evaluation:")
predictionDF =scene_classification_model.transform(validationDF).cache()
predictionDF.sample(False, 0.01).show()

correct =predictionDF.filter("label=prediction").count()
overall = predictionDF.count()
accuracy = correct * 1.0 / overall
predictionDF.unpersist()
trainingDF.unpersist()
print("Accuracy = %g " % accuracy)
```

```
Start evaluation:
+--------------------+-----+----------+
|               image|label|prediction|
+--------------------+-----+----------+
|[file:/home/aksha...|  1.0|       1.0|
|[file:/home/aksha...|  2.0|       2.0|
+--------------------+-----+----------+

Accuracy = 0.692308
```

2. Following code for query image to find the similar image of the house

```
from pyspark.sql.types import DoubleType
from pyspark.sql.functions import udf, struct, col
import numpy as np
from heapq import nlargest

query_origin = "file:./samples/00000099.jpg"
query_record = imageDict[query_origin]
query_classification = query_record[0]
query_embedding = query_record[1]

def get_score(x):
    candidate_classification = x[0]
    candidate_embedding = x[1]
    classScore = 1.0 if query_classification == candidate_classification else 0.0
    visualScore = np.dot(query_embedding, candidate_embedding)
    return classScore + visualScore

score_dict = {k: get_score(v) for k, v in imageDict.items()}

# select the top 3 most similar images
top_3 = nlargest(3, score_dict, key=score_dict.get)
print(top_3)
```

```
['file:./samples/00000099.jpg', 'file:./samples/00003201.jpg', 'file:./samples/00000104.jpg']
```

3. Following image is the query image and we have to find the similar image for this query image

Figure 15Query Image

4. Following code will find out the similar images to query image

```
from pyspark.sql.types import DoubleType
from pyspark.sql.functions import udf, struct, col
import numpy as np
from heapq import nlargest

query_origin = "file:./samples/00000099.jpg"
query_record = imageDict[query_origin]
query_classification = query_record[0]
query_embedding = query_record[1]

def get_score(x):
    candidate_classification = x[0]
    candidate_embedding = x[1]
    classScore = 1.0 if query_classification == candidate_classification else 0.0
    visualScore = np.dot(query_embedding, candidate_embedding)
    return classScore + visualScore

score_dict = {k: get_score(v) for k, v in imageDict.items()}

# select the top 3 most similar images
top_3 = nlargest(3, score_dict, key=score_dict.get)
print(top_3)
```

```
['file:./samples/00000099.jpg', 'file:./samples/00003201.jpg', 'file:./samples/00000104.jpg']
```

5. Following two images are the result of the query image

Figure 16Result

V. CONCLUSION

This paper describes how to build a house recommendation system based on image analysis utilizing Intel's BigDL library on Apache sparks. Three deep learning classification models are used to extract the important semantic tags from real estate images. We further compared different visual similarity computation methods such as GoogleNet, AlexNet, DeepBIT & VGG16 and found image embedding from VGG16 to be the most helpful inference model in our case.

REFERENCES

[1] Dai, J., Wang, Y., Qiu, X., Ding, D., Zhang, Y., Wang, Y., Jia, X., Zhang, C., Wan, Y., Li, Z. and Wang, J., 2018. Bigdl: A distributed deep learning framework for big data. *arXiv preprint arXiv:1804.05839*.

[2] Simonyan, K. and Zisserman, A., 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | Impact Factor: 7.488 |

|| Volume 8, Issue 6, June 2020 ||

[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* (pp. 1-9).

[4] Real Estate Image Classification J. H. Bappy1, J. Barr2, N. Srinivasan3, and A. K. Roy-Chow.dhury41, 4ECE, University of California, Riverside, CA2,3HomeUnion Inc., Irvine, CA

[5] Appalaraju, S. and Chaoji, V., 2017. Image similarity using deep CNN and curriculum learning. *arXiv preprint arXiv:1709.08761*

[6] You, Q., Pang, R., Cao, L. and Luo, J., 2017. Image-based appraisal of real estate properties. *IEEE Transactions on Multimedia*, *19*(12), pp.2751-2759.

[7] Poursaeed, O., Matera, T. and Belongie, S., 2018. Vision-based real estate price estimation. *Machine Vision and Applications*, *29*(4), pp.667-676.

[8] Chen, L., Yang, F. and Yang, H., 2017. Image-based product recommendation system with convolutional neural networks.