PROJECT REPORT
DREAM HOUSE

Submitted towards the partial fulfillment of the criteria for award of Post Graduate Program in Analytics & Artificial Intelligence by Imarticus

Submitted By:

Priya Tiwary
(Email: priyatiwary512@gmail.com)

Course and Batch: PGAA 02
January 2021

Abstract

This paper summarizes the findings and feasibility of Convolutional Neural Networks for a simple House image classification problem. As stated in the problem statement we can see spamming of the content will lead to a huge loss for the business of Dream Housing Property Listings. We are building a Deep Learning model with computer vision techniques that will be trained on the given data to direct irrelevant or spam images before getting uploaded to the website. This will help the company to provide a great user experience as users can find relevant and genuine images of the property.

Since we are dealing with image input hence nothing can be better then CNN. CNN is is one of the main categories to do images recognition, images classifications. is a class of neural networks that specializes in processing data that has a grid-like topology, such as an image. A digital image is a binary representation of visual data. It contains a series of pixels arranged in a grid-like fashion that contains pixel values to denote how bright and what color each pixel should be.

.

Acknowledgements

We are using this opportunity to express our gratitude to everyone who supported us throughout the course of this group project. We are thankful for their aspiring guidance, invaluably constructive criticism and friendly advice during the project work. I am sincerely grateful to them for sharing their truthful and illuminating views on a number of issues related to the project.

Further, we were fortunate to have **Ritesh Bhagwat** as our mentor. He/She has readily shared his immense knowledge in data analytics and guide us in a manner that the outcome resulted in enhancing our data skills.

We wish to thank, all the faculties, as this project utilized knowledge gained from every course that formed the PGAA program.

We certify that the work done by us for conceptualizing and completing this project is original and authentic.

Date: January 26, 2021 Priya Tiwary

Place: Mumbai

Certificate of Completion

I hereby certify that the project titled “Dream House” was undertaken and completed under my supervision by Member 1 from the batch of PGAA 2(Jan 2021)

Date: January 26, 2021

Place – Mumbai

Table of Contents :

Table of Contents

CHAPTER 1: INTRODUCTION

1.1 Objective of the study

Dream Housing Property Listings Company has hosted a website where users can search for their dream houses and also sell their houses. In order to enable better filtering of properties for the buyers, the website mandates the sellers to list their properties only after they upload the images of their houses to attract the relevant buyers. The main objective of this study is to help the company to check the images posted on the website for relevancy. Key point of the study is to build a simple classification model which will state the input images is a house or not. This will help the company to post only the relevant house images on it's website.

1.2 Need of the Study

The study is being carried out in association with Dream Housing Property Listings Company which has hosted a website where users can search for their dream houses and also sell their houses. In order to enable better filtering of properties for the buyers, the website mandates the sellers to list their properties only after they upload the images of their houses to attract the relevant buyers. The need to build a simple classification model for the company to filter images as house or not house image came as they noticed that some of the sellers are violating the rules and uploading random images instead of their houses.

1.3 Business or Enterprise under study

The study is being carried out in association with Dream Housing Property Listings Company. The company is a Property sales venture which has hosted it's own website where users can search for their dream houses and also sell their houses. In order to enable better filtering of properties for the buyers, the website mandates the sellers to list their properties only after they upload the images of their houses to attract the relevant buyers

Confidential and restricted. Do not distribute. © Imarticus Learning

1.5 Tools & Techniques

1.5.1 Tools(Software):

i. Jupyter Notebook – Anaconda

ii. Keras – open source library that provides a Python interface for artificial neural networks.

iii. Flask – Web Framework.

1.5.2 Techniques:

i. **Convolutional Neural Network** : Since we are dealing with image input hence nothing can be better then CNN. CNN is is one of the main categories to do images recognition, images classifications. is a class of neural networks that specializes in processing data that has a grid-like topology, such as an image. A digital image is a binary representation of visual data. It contains a series of pixels arranged in a grid-like fashion that contains pixel values to denote how bright and what color each pixel should be.

ii. **Transfer Learning:** Transfer learning is a machine learning method where a model developed for a task is reused as the starting point for a model on a second task. It is a popular approach in deep learning where pre-trained models are used as the starting point on computer vision and natural language processing tasks given the vast compute and time resources required to develop neural network models on these problems and from the huge jumps in skill that they provide on related problems. It is common to perform transfer learning with predictive modeling problems that use image data as input.This may be a prediction task that takes photographs or video data as input. Various pre-trained models that exist are VGG16, RESNET, MOBILENET, etc.

Confidential and restricted. Do not distribute. © Imarticus Learning

CHAPTER 2: DATA DESCRIPTION, PREPARATION AND UNDERSTANDING

One of the first steps we engaged in was to outline the sequence of steps that we will be following for our project. Each of these steps are elaborated below

2.1 **Dataset Description** :

Data that was provided was collected from various sources. They were in two different folders named as Training and Testing. The training folder contained two folders labelled as House and Not_House with consisting 1000 images in both of them. The testing folder contained 579 images which were not labelled and had to be predicted once model was built. Below 2 sample images of each label has been shown:

HOUSE:

NOT HOUSE:

2.2 **Converting the images into array and fetching the labels**:

For further processing the images had to be converted into array and also append the respective labels in a different array as follows:

```
import os
data = []
labels = []

for category in CATEGORIES:
    path = os.path.join(DIRECTORY, category)
    for img in os.listdir(path):
        img_path = os.path.join(path, img)
        image = load_img(img_path, target_size=(224, 224))
        image = img_to_array(image)
        image = preprocess_input(image)

        data.append(image)
        labels.append(category)
```

2.3 **Converted String labels into Numeric labels and test-train-split as**:

```
lb = LabelBinarizer()
labels = lb.fit_transform(labels)
labels = to_categorical(labels)

data = np.array(data, dtype="float32")
labels = np.array(labels)

(trainX, testX, trainY, testY) = train_test_split(data, labels,
    test_size=0.20, stratify=labels, random_state=42)
```

2.4 **Image Augmentation**:

Image augmentation is a technique used to artificially modifying the images to increase the number of images in the dataset. I have used 8 different kinds of image augmentation techniques for this data, they are, Rotation, zoom, width and height shift, shearing, horizontal flip, fill mod

```
from tensorflow.keras.preprocessing.image import ImageDataGenerator

train_datagen = ImageDataGenerator(rotation_range=20,
    zoom_range=0.15,
    width_shift_range=0.2,
    height_shift_range=0.2,
    shear_range=0.15,
    horizontal_flip=True,
    fill_mode="nearest")
```

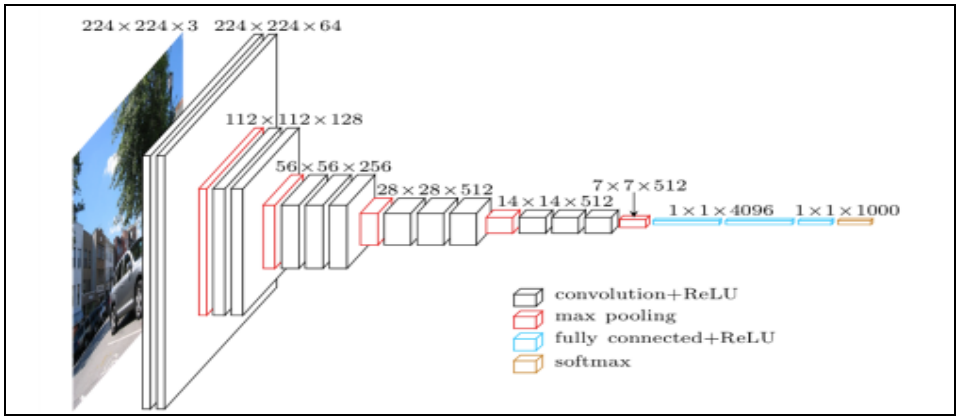
CHAPTER 3: MODELLING APPROACHES ATTEMPTED

As mentioned earlier Transfer Learning has been always best technique to deal with image classification and processing problems, hence tried two pre processed models i.e VGG16 and Mobilenet to develop the Dream House image classification model.

1. **VGG -16**:
   VGGNet-16 consists of 16 convolutional layers and is very appealing because of its very uniform Architecture. It has only 3x3 convolutions, but lots of filters. It can be trained on 4 GPUs for 2–3 weeks. It is currently the most preferred choice in the community for extracting features from images. The 16 layers of VGG16 are as follows:

1.Convolution using 64 filters
2.Convolution using 64 filters + Max pooling
3.Convolution using 128 filters
4. Convolution using 128 filters + Max pooling
5. Convolution using 256 filters
6. Convolution using 256 filters
7. Convolution using 256 filters + Max pooling
8. Convolution using 512 filters
9. Convolution using 512 filters
10. Convolution using 512 filters+Max pooling
11. Convolution using 512 filters
12. Convolution using 512 filters
13. Convolution using 512 filters+Max pooling
14. Fully connected with 4096 nodes
15. Fully connected with 4096 nodes
16. Output layer with Softmax activation with 1000 nodes.

This 16 layered architecture in pictorial format is as below:



The *VGG()* class takes a few arguments. Some of them are:

- **include_top** (*True*): Whether or not to include the output layers for the model. You don't need these if you are fitting the model on your own problem.
- **weights** ('*imagenet*'): What weights to load. You can specify None to not load pre-trained weights if you are interested in training the model yourself from scratch.
- **input_tensor** (*None*): A new input layer if you intend to fit the model on new data of a different size.
- **input_shape** (*None*): The size of images that the model is expected to take if you change the input layer.
- **pooling** (*None*): The type of pooling to use when you are training a new set of output layers.
- **classes** (*1000*): The number of classes (e.g. size of output vector) for the model.

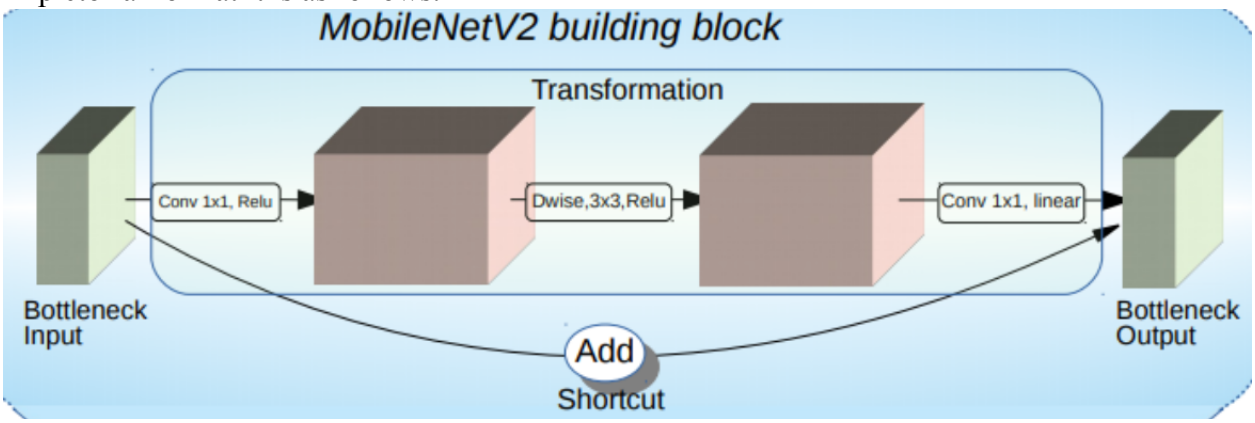
Confidential and restricted. Do not distribute. © Imarticus Learning

2. **MOBILENET**:
   Mobilenet is TensorFlow's first mobile computer vision model.It uses depthwise separable convolutions which basically means it performs a single convolution on each colour channel rather than combining all three and flattening it.
   For MobileNets the depthwise convolution applies a single filter to each input channel. The pointwise convolution then applies a 1×1 convolution to combine the outputs the depthwise convolution. A standard convolution both filters and combines inputs into a new set of outputs in one step. The depthwise separable convolution splits this into two layers, a separate layer for filtering and a separate layer for combining. This factorization has the effect of drastically reducing computation and model size.

   So the overall architecture of the Mobilenet is as follows, having 30 layers with

   1. convolutional layer with stride 2
   2. depthwise layer
   3. pointwise layer that doubles the number of channels
   4. depthwise layer with stride 2
   5. pointwise layer that doubles the number of channels.

In pictorial format it is as follows:



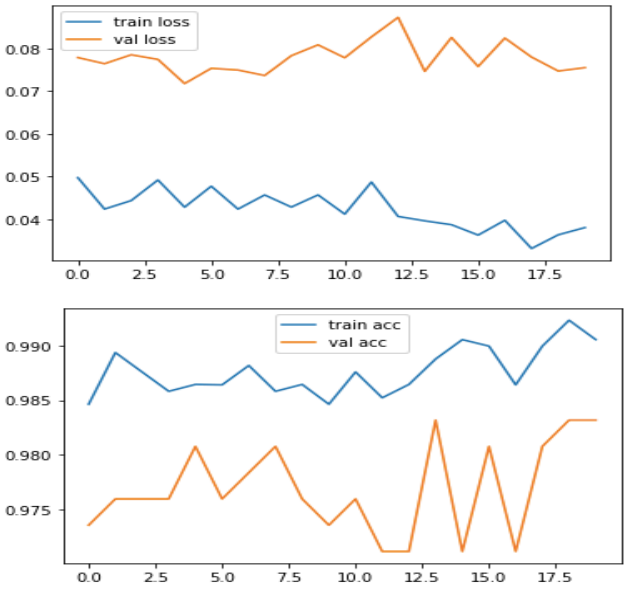
Type / Stride	Filter Shape	Input Size
Conv / s2	$3 \times 3 \times 3 \times 32$	$224 \times 224 \times 3$
Conv dw / s1	$3 \times 3 \times 32$ dw	$112 \times 112 \times 32$
Conv / s1	$1 \times 1 \times 32 \times 64$	$112 \times 112 \times 32$
Conv dw / s2	$3 \times 3 \times 64$ dw	$112 \times 112 \times 64$
Conv / s1	$1 \times 1 \times 64 \times 128$	$56 \times 56 \times 64$
Conv dw / s1	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 128$	$56 \times 56 \times 128$
Conv dw / s2	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 256$	$28 \times 28 \times 128$
Conv dw / s1	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 256$	$28 \times 28 \times 256$
Conv dw / s2	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 512$	$14 \times 14 \times 256$
$5\times$ Conv dw / s1	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
$5\times$ Conv / s1	$1 \times 1 \times 512 \times 512$	$14 \times 14 \times 512$
Conv dw / s2	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
Conv / s1	$1 \times 1 \times 512 \times 1024$	$7 \times 7 \times 512$
Conv dw / s2	$3 \times 3 \times 1024$ dw	$7 \times 7 \times 1024$
Conv / s1	$1 \times 1 \times 1024 \times 1024$	$7 \times 7 \times 1024$
Avg Pool / s1	Pool 7×7	$7 \times 7 \times 1024$
FC / s1	1024×1000	$1 \times 1 \times 1024$
Softmax / s1	Classifier	$1 \times 1 \times 1000$

Confidential and restricted. Do not distribute. © Imarticus Learning

CHAPTER 4: KEY FINDINGS

1. VGG-16 MODEL FINDINGS:

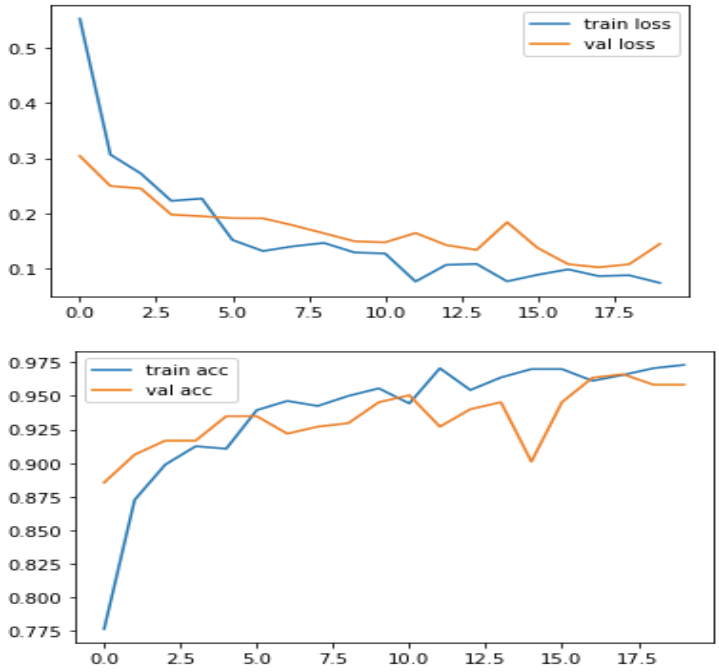
Though the accuracy achieved with VGG16 model was good but seeing the loss and accuracy plot below I felt model isn't fitting and adapting well with the data.
We can see a wide gap in between the losses as well as accuracies which bothered me much and I doubted to finalize this model.



	precision	recall	f1-score	support
House	0.97	0.99	0.98	214
Not_House	0.99	0.97	0.98	217
accuracy			0.98	431
macro avg	0.98	0.98	0.98	431
weighted avg	0.98	0.98	0.98	431

2. MOBILENET MODEL FINDINGS:

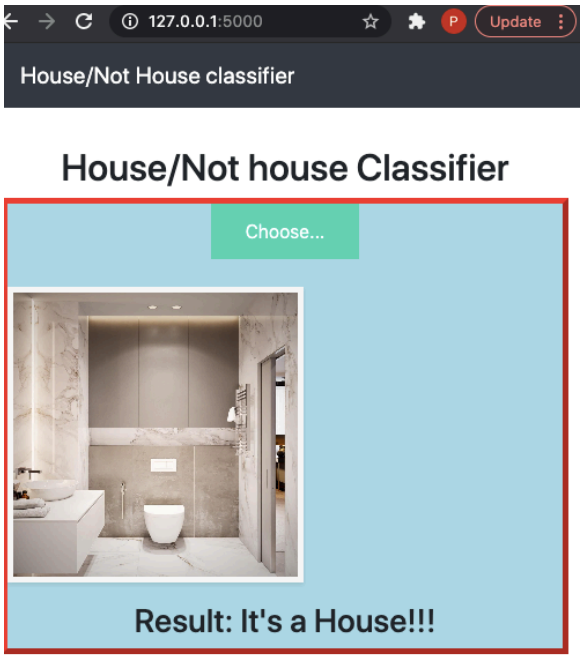
When I compared this finding with VGG16 I found Mobilenet to be fitting very well , not just on accuracy but seeing the classification report as well. With high precision as well as f-1 score as compared with VGG16 , I found Mobilenet to be performing good on the available data.



	precision	recall	f1-score	support
House	0.97	1.00	0.98	214
Not_House	1.00	0.97	0.98	217
accuracy			0.98	431
macro avg	0.98	0.98	0.98	431
weighted avg	0.98	0.98	0.98	431

CHAPTER 5: CONCLUSION

- Deployed the model using FLASK wherein you can choose a image and predict it . A snapshot of the same is:



- We predicted the test data and found results to be good.
- The model was predicting images correctly and also saved the results in csv file.
- The MobileNet V2 algorithm working amazingly well in predicting both test data as well as the real world data through Flask Web Application.