IMAGE RECOGNITION AND CLASSIFICATION ON REAL ESTATE IMAGES

Abhijeet Chawhan
School of Data Science
University of Virginia
unn8qd@virginia.edu

Ahmed Soliman
School of Data Science
University of Virginia
as7vc@virginia.edu

Karan Manwani
School of Data Science
University of Virginia
akp4he@virginia.edu

Matt Litz
School of Data Science
University of Virginia
msl2t@virginia.edu

December 8, 2022

1 Abstract

'A good picture is worth a thousand words' and a good picture does help in selling a house. Posting pictures is a necessary part of advertising a home for sale. Agents typically sort through dozens of images from which to pick the most complimentary ones. This is a manual effort involving annotating images accompanied by descriptions (bedroom, bathroom, attic, etc.). When volumes are small, manual annotation is not a problem, but there is a point where this becomes too burdensome and ultimately infeasible. Here, we propose an approach based on computer vision methodology to radically increase the efficiency of such tasks. We present a high-confidence image classification framework, whose inputs are images and outputs are labels. The core of the classification algorithm involves use of the Convolution Neural Network (CNN), and fully connected neural networks, along with a substantial preprocessing for image enhancement. Our project for Deep Learning is based on the computer vision with the goal of image classification and recognition. We are provided with the images of the rooms of houses and these images are taken from different angles. We will use these images to recognize them and classify them into images of living room, bedroom, kitchen, bathroom, etc. We need to create a robust model that can handle such classification.

2 Motivation

Our goal is to determine the type of room in an image to enhance the user experience for platforms using home images for their operations, such as real estate and travel sites. This model should benefit both the provider and the consumer of the service. From the provider's perspective, such as the real estate agent or the host of a vacation home, this classification model will aid in organizing their home images by room type even if they upload them in a random order. This will save them time and provide a more professional display of their home to potentially drive up sales. From the consumer's perspective, when they visit a site to rent a vacation home or purchase a new house, having the room type classified will give them an organized view of the home, and also the ability to filter by room type if they want to view certain rooms or toggle easily between room types.

The project category. We approach this task as a image recognition and classification problem. For every image, we need to classify into one of the classes which are based on the different rooms of houses.

3 Dataset

The data set used for studying this motivation was downloaded from Kaggle.com and is available at

https://www.kaggle.com/datasets/robinreni/house-rooms-image-dataset. The data consists of images of all the different types of rooms of a house. Each image size is approximately about 30 KB. Total number of images utilized for this study is 5250.

3.1 Exploratory Data Analysis

The tables below show the image statistics for each image class for training and validation set.

Image Stats	
Room Type/Classes	Image Count
Living Room	1273
Kitchen	965
Bedroom	1248
Bathroom	606
Dining	1158

Table 1: Training Data set: Number of Images of each Room Type

Image Stats	
Room Type/Classes	Image Count
Living Room	149
Kitchen	185
Bedroom	104
Bathroom	235
Dining	189

Table 2: Validation Data set : Number of Images of each Room Type

The figure below shows the image count for each image class from the dataset.

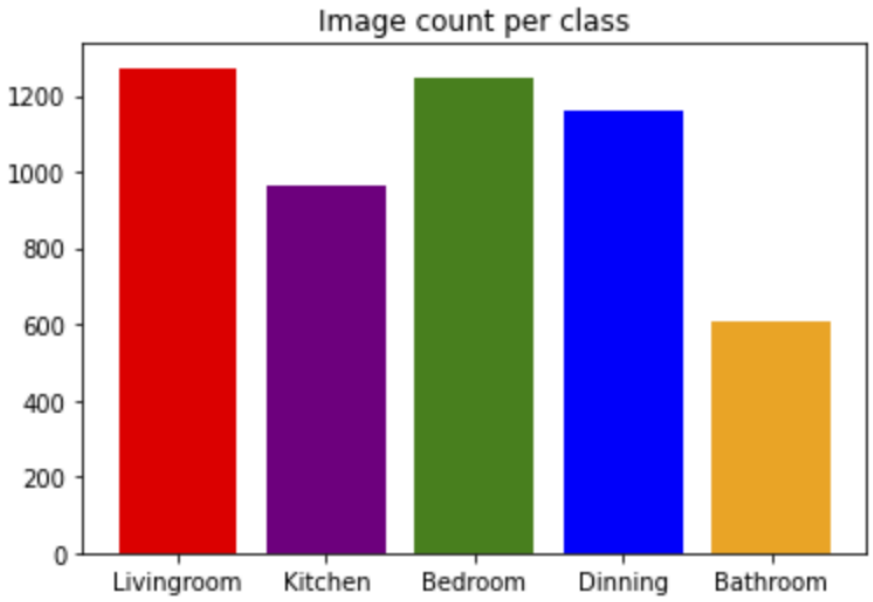

Figure 1: Sample Images from the classes

This data set will be augmented by another data set which will enable performing the another study which is the stretch goal (further fine tune the classification) of this project. These images were downloaded from https://www.kaggle.com/datasets/barelydedicated/airbnb-duplicate-image-detection

4 Literature Survey

Image Classification of Real Estate is a highly sought after topic due to the value it brings all parties involved in the Real Estate transactions. Buyers spend a considerable time browsing through pictures to estimate the characteristics of a property. Large property management companies analyze large volume of images for independent evaluation of property values. Making a software for automatic recognition of certain highlights of a property, which can be used to further evaluate the value of a property.

Although there are numerous examples of similar studies online, The University of California at Riverside stuck out to us due to their detailed explanation of challenges they faces and the goal they were attempting to achieve. Their goal was to build a model that classifies about 7000 Real Estate images based on (1) scene classification, (2) countertop classification, and (3) floor classification. They initially attempted to use CNN since it shows promising performance in image classification, although CNN is computationally expensive requiring lots of labeled data to train a model for high input dimensionality. However, they decided on a different approach which is Long-short term memory (LSTM) along with fully connected layers is utilized in order to classify images. They picked LSTM due to its key advantages over CNN since it reduces computational complexity and has good recognition performance. For their choice of LSTM network, 4 layers were used and 128 cells per layer. The training model was done by Cross entropy loss with an input dimension of 128x128 where each cell of input layer receives 128 pixels at a time.

The results of both their CNN and LSTM showed that LSTM showed a better and higher accuracy result in all categories which was 96.92% in the scene classification, 86.12% in the counter-top classification, and 81.94% in the floor classification. In conclusion, the proposed LSTM model performs better, while the CNN model faces overfitting problem due to large set of parameters, according to UC Riverside researchers. The proposed LSTM model can be used as alternative in any recognition task when less amount of data is present and it reduces computational complexity with respect to other deep models while maintaining good recognition performance. Here we are attempting to challenge their claim that LSTM performs better than CNN by testing our model using CNN and aiming to get a similar or a higher accuracy result.

University of California Riverside Research Reference:

`https://www.researchgate.net/publication/316494092_Real_Estate_Image_Classification`

5 Intended experiments

This project principally involves image classification, therefore we will employ Convolutional Neural Networks (CNNs) to classify the images.

6 Methods

6.1 Data Pre-Processing

Our technique for classifying the different room images is to leverage pre-trained Convolutional Neural Network (CNN) architectures, and then apply our own layers with hyperparameter tuning. To achieve this, we first applied various pre-processing steps to the data. The dataset has five classes, which include Living Room, Bedroom, Bathroom, Dining Room, and Kitchen with each class having between 600 and 1200 images. Image labeling (using the Datasets API) was performed by splitting the classes into different folders with each folder containing images that belong to the corresponding class. The image dimensions were set to 150 x 150 and this can easily be adjusted if certain models require a different size. The data was loaded as categorical for the lable mode setting to enable use to use categorical crossentropy as the loss function instead of sparse categorical crossentropy. We opted for this approach as it works better for some metrics functions, such as F1 score using the tfa.metrics module. A training and validation split of 0.2 was applied to the dataset with 5,250 images resulting in 4,200 images for training and 1,050 for validation. Figure 1 contains a sample of the images along with the class name.

6.2 Preliminary Experiments : Models Tried

After the preprocessing of the image data and splitting the same in Test and validation set, the section below describes about the training exercise using the pre-trained algorithm. This study employed the pre-trained CNN architecture as these pre-trained algorithms provide the accuracy which is hard to beat.

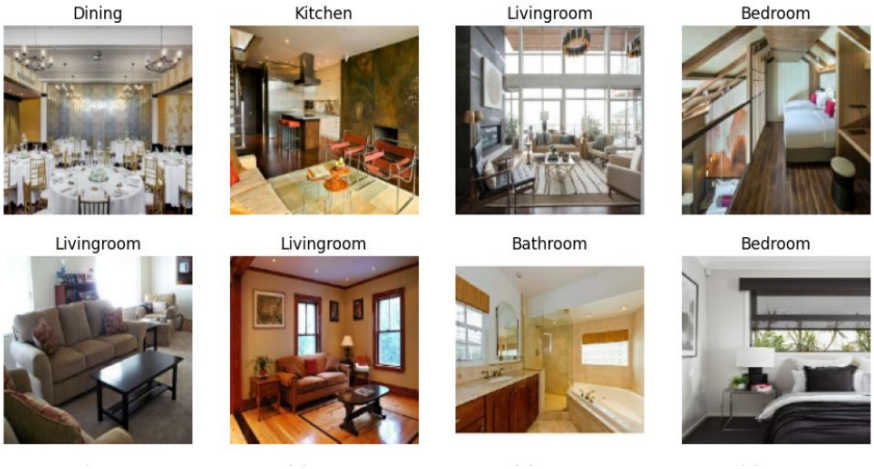

Figure 2: Sample Images from the classes

6.2.1 ML Techniques

EfficientNetB0 First pre-trained model that was applied to the data set used in this project was 'EfficientNetB0'. The idea to use this pre-trained model was based on the highest accuracy it provides on the image classification. The first experiment includes training using only one fully connected layer with 400 neurons fed using the output of pre-trained EfficientNetB0 layer. As a start for this experiment, the model was trained using only 10 epochs. Learning rate of 0.001 was used for the optimizer and the optimizer used was 'Adam'. The output layer was designed to output in 5 neurons to suffice for the five classes in which the images are to be classified from the dataset. The loss function utilized for the training in the first model is "sparse categorical crossentropy" since the study involves multi-class classification.

EfficientNetV2L When applying this architecture to our dataset, and after using average pooling, we also fed 400 neurons to the output layer with batch normalization and relu activation. The Adam optimizer with learning rate of 0.001 was used for 10 epochs. This model ran slower than the EfficientNetB0.

EfficientNetV2M Efficient V2M was tested with Batch Normalization, 5 dense fully connected layers, and "relu" activation for the fully connected layers, and "softmax" for the output layer.

InceptionResNetV2 The pre-trained architecture - InceptionResNetV2 was chosen as our next model for its computational and accuracy efficiency. It's highest depth provides better accuracy in terms of classification and it is computationally less aggressive. However, with regular untuned hyper parameter settings this algorithm didn't yield the accuracy it promises. The model needs to be tuned with appropriate values of hyper parameters.

InceptionV3 Similar to InceptionV2, extensive hyperparameter tuning would be required to get higher accuracy from InceptionV3. It appears that the Inception family of pretrained models are underperforming when compared to the EfficientNet family.

6.3 Preliminary Results

With five pre-trained models in the works with basic tuning parameters, the accuracy measures of these models is as shown in the table below.

7 Experiments

Based on the preliminary results, we ended up using the EfficientNetB0 pre-trained model as our CNN architecture for building a classification model. This model had produced the highest validation Accuracy. The imagenet weights were used, and we applied average pooling in all the models we tried. We experimented with several dense layers, applied different activation layers, dropout functions, and the adam optimizer (with various learning rates). Given the slightly imbalanced classes, we added F1 scores to evaluate the models. F1 Scores were calculated with "Macro" averaging, and without any averaging (individual scores for each class)

Accuracy Measures		
Pre-Trained Models	**Training Accuracy**	**Validation Accuracy**
EfficientNetB0	80.48	79.81
EfficientNetV2L	91.64	79.05
EfficientNetV2M	87.93	74.1
InceptionResNetV2	32.83	29.71
InceptionV3	41.93	40.19

Table 3: Preliminary Accuracy Results : Without Hyper-parameter Tuning

7.1 Training Process

Below lists the parameters we used in our final model along the Hyperparameter tuning we attempted:

- Input image size = 150 x 150 (tried 200 x 200, 300 x 300)
- Batch Size set to 32.
- Softmax "categorical crossentropy" used with imagenet Weight normalization.
- Adam Optimizer with exponential learning rate decay of 0.0001. Learning rates attempted varied from 0.0001 - 0.1.
- Number of Dense layers used - 1, Attempted with 1 through 5
- Number of neurons in each layer - 400. Attempted with 100 to 500
- Batch Normalization after dense layers.
- Activation functions utilized - Relu
- Dropout of 0.2 before the output layer.
- Epoch settings varied from 10 to 200. Final model had 50 epochs.
- F1 score (for evaluation) with Macro averaging and no Averaging.

7.2 Results

Final models were evaluated with the following metrics:

- Accuracy
- F1 Score
- Validation Loss

The model produced an Accuracy score of 81.90% and an F1 Score of 82.17% on the Validation data. The F1 Score was after Macro Averaging. Table 4 has the summary of these results.

Accuracy Results				
Final Model	**Training Accuracy**	**Validation Accuracy**	**Validation Loss**	**Validation F1 Score**
EfficientNetB0	98.43	81.90	68.42	82.17

Table 4: Final Results

Figure 2 shows how the accuracy and F1 scores varied when increasing the epochs and we noticed that with the current set up, the scores leveled off at about 82%.

F1 scores without macro averaging was also produced to view the individual F1 scores for each of the classes. The results are shown in Figure 3, and it can be seen some classes reached 85% F1 scores, where as some were below 80%.

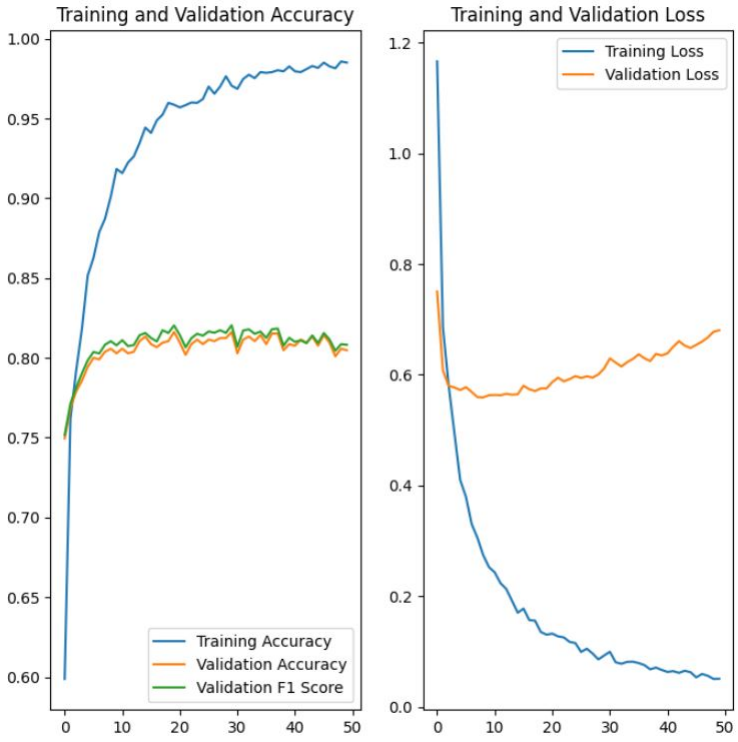

Figure 3: Training and Validation Loss

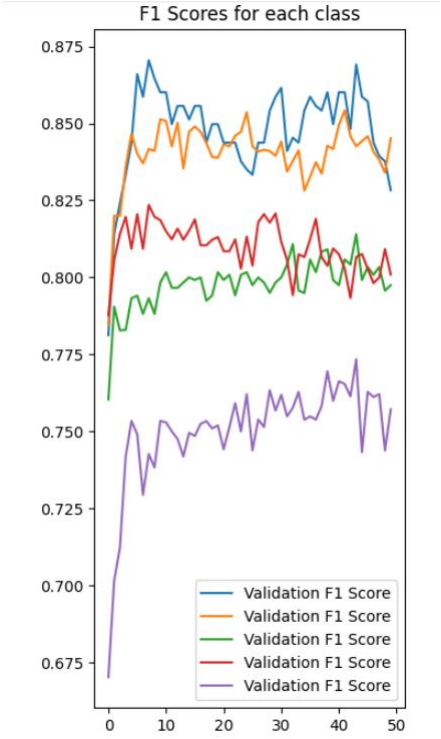

Figure 4: F1 Scores for each class

8 Additional Experiments

Here we developed tools to utilize our models in potential applications in the Real Estate industry.

8.1 Image Captioning

Image captioning is a vector-to-sequence model which predicts the captions for an input image. This model utilizes both a CNN and Transformer to receive the paired images and makes predictions on a given image. Or existing data set was lacking captions therefore we manually captioned a subset of the data and paired the caption with the corresponding image in a text file. Additionally, we utilized the CNN architecture implemented for our initial CNN model into our image captioning application.

The validation achieved in this model was a loss of 2.1708 and accuracy of 53.8%. To improve our accuracy score, it is anticipated that the entire dataset would need to be fully captioned. One potential use case for this application would be for real estate agents bringing a property to market and listing the property in an online marketplace. The agent could take a photo of a location on the property, auto-caption the photo, and then feed the image and caption to a database. See Figure 5 for sample output.

Figure 5: Sample captioned images.

8.2 iOS Application

Since the real estate agents are very transient during their workday (for example: visiting new properties, showing properties to prospective customers) we felt that a useful classification application must be functional on a mobile device. We decided to build a Tensorflow Lite model using our dataset and deploy the model within an iOS application (see Figure 6). The model was developed using Tensorflow lite Model Maker and "EfficientNet-lite0" was used as the model architecture. The model achieved a achieved a loss of 0.7256 and a very impressive accuracy of 83.8%. The resulting .tflite model file was 3 MB which could port easily to any mobile device.

Figure 6: Screenshot of iOS Real Estate Classification application.

9 Conclusions and Future Work

- EfficientNetB0 turns out to be the best pre-trained algorithm for the house image classification with the highest F1-score.
- The additional experiment conducted to caption the images yielded high accuracy can be effectively utilized by the real estate firms, broker companies and real estate inspectors.
- With the iOS application developed using TensorFlow Lite, and EfficientNetB0 as the classification algorithm, the app can be utilized by real estate inspectors to conduct real estate inspection with ease.
- House Image classification using computer vision with models trained and tested over several pre-trained efficient algorithms, and with integration of image caption and mobile application turned out to be a great product for the real estate industry in several aspects.
- The future work in real estate domain utilizing the image classification, captioning and the mobile application will be to integrate with the pictures taken using drones and perform the inspection of a real estate property from remote.
- Future work will include the extension the dataset by adding new real estate image categories. We also plan to provide images to analyze anomaly in real estate images such as defective floors, roofs, etc.

10 Team Contribution

Each member of the project team contributed equally to each section of the milestone. Specific contributions and/or sections worked upon by each member is as below:

- Abhijeet Chawhan(unn8qd) studied existing literature, implemented data preprocessing, performed exploratory data analysis and trained the model using the InceptionResNetV2 and InceptionV3 algorithms.
- Karan Manwani (akp4he) Implemented model using EfficientNetB0, trained with different paramenters,performed Hyperparameter Tunning and included measurement parameters - F1 score and Accuracy.
- Ahmed Soliman (as7vc) studied the related work done, models trained, implemented EfficientNetV2L and EfficientNetV2M models, trained them using various parameters and performed the required testing.
- Matt Litz (msl2t) worked on the complex Image captioning implementation by training models on pre-trained algorithms and also built the iOS application using TensorFlow Lite.

The paper was written by all the team members each contributing to one of the sections of the milestone.

11 References

Reference 1: https://www.kaggle.com/datasets/robinreni/house-rooms-image-dataset.

Reference 2: https://www.kaggle.com/datasets/barelydedicated/airbnb-duplicate-image-detection

Reference 3: https://www.researchgate.net/publication/316494092$_{R}eal_{E}state_{I}mage_{C}lassification$

Reference 4: https://keras.io/examples/vision/image$_{c}aptioning/$

Reference 5: https://www.tensorflow.org/lite/models/modify/model$_{m}aker/image_{c}lassificationn$