Data Science Programming Final Project

Clean Room vs. Messy Room Classification

Group Member:
Qian Wang
Xinyu Jiang
Binrui Yang
Zhenglong Wu

Table of Contents

1. Introduction and Motivation

Nowadays, as AI technology developing, image classification has become part of human life for a long time. A picture is worth a thousand words. We continually capture visual content, explain its meaning, and use them for practice. However, it is difficult for a computer to interpret the content of an image because the image seen by the computer is a large digital matrix that knows nothing about the idea, knowledge, and meaning of the image. Classification of very similar images is one of the most important and difficult tasks in the field of computer vision.

In recent times, considerable improvement has been made in machine learning with the information obtained from studies of the neurologists and academic environment. To understand the content of the image, we must apply image classification, which is the task of extracting meaning from the image using computer vision and machine learning algorithms. This operation can simply assign a label to an image, such as a cat, dog, or elephant, or it can be advanced to interpret the content of the image and return a human-readable sentence. The driver less technology relies much on image classification,

Image classification is a vast area of research, including a variety of technologies, and it continues to evolve with the deep learning. We get many knowledges for machine learning through this whole semester that provide us a good opportunity to use machine learning and deep learning to explore image classification area.

2. Related Work

The project we are researching is to build a classifier of clean room and messy room. After arbitrarily inputting a room image, the machine can automatically determine whether the image represents a clean room or a messy room. As with all image classification and recognition tasks, we also need to analyze image data, which is also a typical high-dimensional data. In general, for high-dimensional data, we must first find ways to reduce the dimensions. There are two main approaches in mainstream data analysis methods: feature selection and feature extraction [1].

Feature selection refers to the process of obtaining a subset from a set of original features according to certain feature selection criteria. Feature selection technology can preprocess learning algorithms. Good feature selection results can improve learning accuracy, reduce learning time, and simplify learning outcomes. Feature extraction usually requires conversion of raw data into features with more different pattern recognition capabilities, where the raw data can be viewed as features with weaker recognition capabilities.

At present, deep learning is the most widely used and effective model for the machine to automatically select and extract features, with or without supervision, semi-supervised and supervised. For example, CNN extracts image features through multiple levels, with convolution layer, activation layer, Pooling layer and fully connected layer in the middle [2]. The first layer extracts more common features (such as edge information) while the second layer extracts more specific features.

In addition, on the basis of CNN, a method of learning transfer is proposed, which has a good effect on image classification that is difficult to distinguish [3].

3. Methodology

3.1. Precise Problem Statement

Sometimes to mark a clean and messy room are difficult, such as different people have different standard level to judge the clean and messy. For machine learning, has the same problems. The shade of floor, the wallpaper, and the texture would be the noise for machine learning. So, in our first exploring phase we would like to collect the data set with distinct characteristics.

3.2. Data Library and Source

a. Data source

All of the data sources we collected from Google images. Because this is our initial image classification exploring, so we focus on collecting hotel room images. Also, we choose the wall of the hotel rooms without wallpaper, the wallpaper would cause some negative impact for our model.

b. Data processing

For the training and validation set, we collect 155 clean room images and 125 messy room images. For the test set, we prepare 5 cleanroom images and 5 messy room images. We manually attribute these images to a clean and messy folder. Also, we add the photo label for clean and messy images. Our clean and messy judging criteria are mainly for the bed in the room.

3.3. Methods and Techniques

a. Feature Engineering

Our first approach is the feature engineering, which is mainly to find out the different rules between clean room and messy room image. So, the first problem we meet is to think about how to define the subjective concepts of “clean” and “messy”. For instance, people generally think that Figure 1.1 is a clean room and Figure 2.1 is a messy room. Why do our human intuitions think that a room is very clean or messy? There is a very intuitive explanation that the signal released in a clean room is simple, but the messy room releases lots of information and noise in its signal. Therefore, we used smooth denoising and edge detection to screen out the important features of the image as shown in Figure 1.2 and Figure 2.2 by the convolution operation.

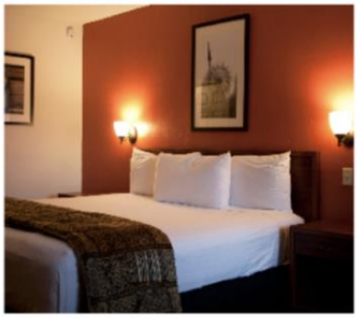

Figure 1.1

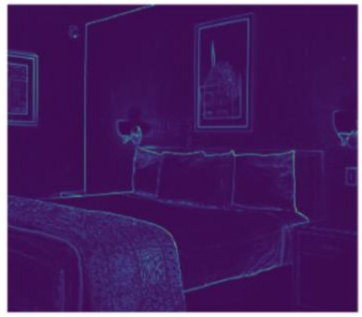

Figure 1.2

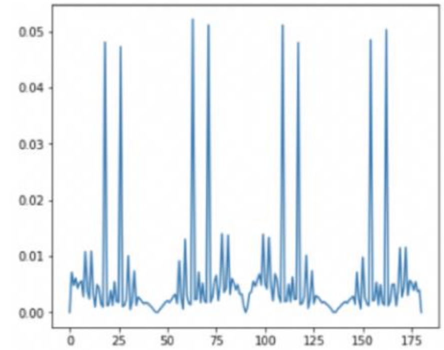

Figure 1.3

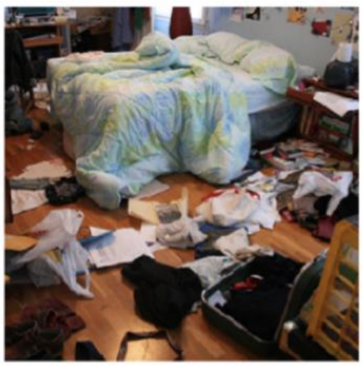

Figure 2.1

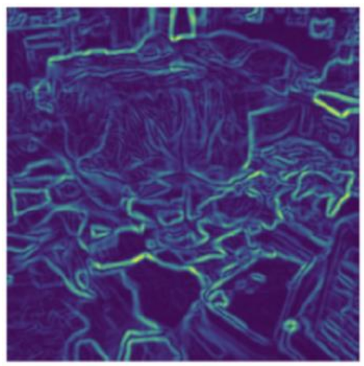

Figure 2.2

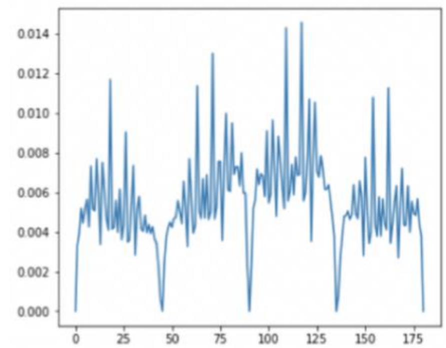

Figure 2.3

Comparing Figure 1.2 with Figure 2.2, which only have the edge information in the image, we found a more objective rule. The clean room generally have parallel straight edges. However, the messy room has lots of irregular curve edges. So how do we make the machine understand this difference? We think that if all pixels on the edge have a direction attribute, then it should be the edge direction of the tangent through the pixel. Edge is description of how fast color changes. Thus, we implement edge detection and edge direction according to the following equation:

$$\frac{\partial I(x,y)}{\partial x} = \frac{I(x,y) - I(x+\Delta x, y)}{\Delta x} \qquad \frac{\partial I(x,y)}{\partial y} = \frac{I(x,y) - I(x, y+\Delta y)}{\Delta y}$$

$$Edge\ Magnitude = \sqrt{\left(\frac{\partial I(x,y)}{\partial x}\right)^2 + \left(\frac{\partial I(x,y)}{\partial y}\right)^2}$$

$$direction\ \theta = \operatorname{atan}\left(\frac{\partial I(x,y)/\partial y}{\partial I(x,y)/\partial x}\right)$$

I(x,y) represents the image. ∂I(x,y)/∂x is the derivative of the pixel in the x direction, ∂I(x,y)/∂y is the derivative of the pixel in the y direction. θ is the direction of the edge pixel. In this experiment, we did the convolution by Sobel edge detection operator to get derivative in both x and y directions.

We assume that in a clean room, the edge direction degrees should be highly concentrated on several special values. However, in a messy room, due to a lot of curves and circles, the degrees of the edge direction should be evenly distributed from 0 to 180 degrees. Then we count the number of pixels from 0 to 180 degrees and make a histogram for the distribution of pixels at different direction as Figure 1.3 and 2.3. In both two histograms, the x axis is the degree from 0 to 180, the y axis is the ratio of the number of all image pixels. Although they all seem to be fluctuating, the ratios of clean room histogram range from 0 to 0.05, and the ratios of messy room histogram range from 0 to 0.014. Therefore, the pixel direction distribution of the dirty room is indeed more even than the clean room.

In order to manually enhance the difference between both two classes images, we used this transformation method to get the pixel direction histogram of each image and use this histogram as the input data, which is a vector consisting of 180 decimals that less than one. Then we designed 8 classifier models to train the data: Stochastic Gradient Descent, Decision Tree, Naive Bayes, Logistic Regression, Support Vector Machine, Random Forest (Bagging), Voting Classifier (Ensemble), XGBoost (Boosting).

b. Convolutional Neural Network

i. Build baseline model

In order to find a better solution to make a prediction, we decided to build CNN classifier with Keras. We build three models in this part. First developed a baseline Convolutional Neural Network model that helps establish a minimum model performance to make a comparison with other CNN models that added complexity and regularization. The baseline model has one hidden layer that uses the Relu activation function with 15 epochs.

ii. Add regularization and complexity

In order to increase the model accuracy and reduce the errors. We pay much attention on this part. First our target is to increase the accuracy of the CNN classifier, we add more hidden layers, max pooling layer, and trying use change parameter such as different batch size, learning rate, and epochs. As the complexity adding, the training set accuracy significantly increase, however, overfitting problem emerged.

iii. VGG 16 transfer learning

The third model we built with VGG16 that takes the pre-trained weights of an already trained model and use these already learned features to predict new classes.

4. Evaluation Result

4.1. Feature Engineering

a. Evaluation

		Training Set (224)		Validation Set (56)		Problem
		Accuracy	Precision (Clean)	Accuracy	Precision (Clean)	
1	SGD Classifier (loss: log)	0.737	0.679	0.732	0.682	
2	Decision Tree	1	1	0.679	0.71	Overfitting!
3	Naïve Bayes	0.625	0.601	0.625	0.604	
4	Logistic Regression	0.754	0.732	0.768	0.75	
5	Support Vector Machines (linear)	0.772	0.748	0.767	0.75	
6	Random Forest (Bagging)	1	1	0.786	0.88	Overfitting!
7	Voting Classifier Hard (Ensemble)	0.835	0.788	0.75	0.743	
8	Voting Classifier Soft (Ensemble)	0.951	0.919	0.75	0.742	Overfitting!
9	XGBoost (Boosting)	1	1	0.75	0.84	Overfitting!

The Training Set has 224 images. And Validation has 56 images, which were chosen by train_test_split function of sklearn library, and the random state is 42. True is Clean room, False is Messy room. The Voting Classifier includes all models from number 1 to 6.

As we can see, Decision Tree, Random Forest and XGBoost all get 100% accuracy in the training dataset but performs badly in the validation dataset. They have overfitting problem. We all know that decision tree models are very prone to overfitting problems, because it is very easy to be designed as to perfectly fit all samples in the training data. When there is noise in the data, or the number of training examples is too small to produce a representative sample of the objective function, it will affect accuracy when decision tree predicting other samples that are not part of the training set. Even we can avoid overfitting by controlling the depth of the tree, or pre pruning and post pruning, we thought the method of Random Forest model that randomly choose different features to get collection of trees would be better to solve overfitting. However, Random Forest only reduces the degree of overfitting a bit. And decision tree will ignore the correlation between the features, which is not fit for our data, because there are 180 features in our data, they are actually related, representing the proportion of the total 1. Then we abandoned these over-fitting models, including the Soft Voting ensemble model that contains the decision tree and random forest.

In the remaining models, SGD Classifier, Logistic Regression, Support Vector Machine and Hard Voting Classifier performs similarly. And the Precision of both Logistic Regression and Support Vector Machine are best 0.75. Since our “True” class is set to the clean room, we want to be able to predict clean rooms very strictly, and even classify some of the true clean rooms to be messy, ensuring that the rooms that are predicted to be clean must be truly clean. Thus, we hope to choose the model with the higher precision, and accuracy of the model should not be low.

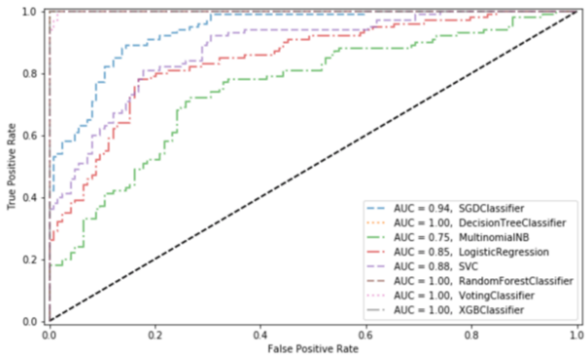

Figure 3.1 Training Set

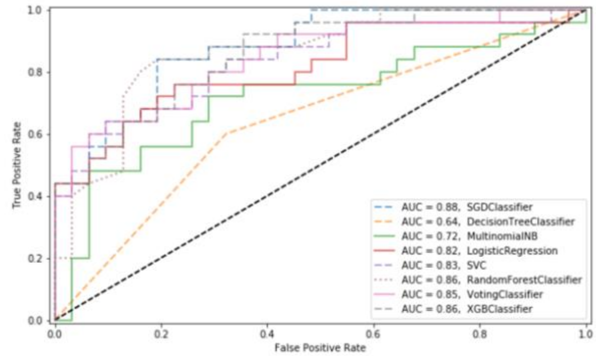

Figure 3.2 Validation Set

From the ROC curve in Figure 3.1 and Figure 3.2, we can notice that SGD classifier is the best model on performance in both Training set and Validation set, after excluding the overfitting models. In order to reduce the possibility of accidental factors in the performance of the model, we need to test those again in the prediction dataset.

b. Prediction

		Test Set (10)	
		Accuracy	Precision (Clean)
1	SGD Classifier (loss: log)	0.8	0.714
3	Naïve Bayes	0.6	0.556
4	Logistic Regression	1	1
5	Support Vector Machines (linear)	1	1
7	Voting Classifier Hard (Ensemble)	1	1

Prediction dataset is selected manually by us with 5 clean rooms and 5 messy rooms. They are similar to the test dataset, helping us to evaluate the accuracy of the model again before it is used in production.

There are 5 remaining non-overfitting models in the table above, Logistic Regression, Support Vector Machine and Hard Voting Classifier are 100% accurate in classification. And Naive Bayes performs worst. Even SGD use the “log” as its loss function, it still has a subtle difference with Logistic regression, there might be a reason that SGD has deviated from the optimal parameters by inappropriate learning rate setting. And another reason could be that the test dataset is too small.

If based solely on the background of the current experiment, we think that Logistic Regression and Support Vector Machine are suitable for prediction and evaluation in the next step of production. But we found that the CNN model should be more advantageous.

4.2. Convolutional Neural Network

a. Evaluation

CNN models		
Model	Training set	Validation set
	Accuracy	Accuracy
Baseline	83.93%	67.86%
Regularize	94%	81.43%
Transfer learning	95.09%	92.86%

Figure 4.2.1

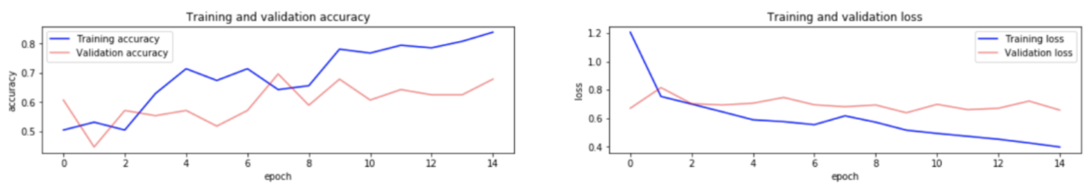

Figure 4.2.2

Figure 4.2.2 shows the accuracy of training and validation and the loss of training and validation loss of baseline model. From the curve, we can see the baseline model does not perform well. From reviewing the learning curves for the baseline model, it shows the strong signs of overfitting. Thus, we will ignore this model in the production.

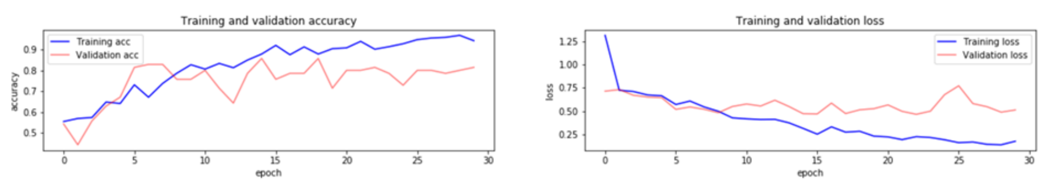

Figure 4.2.3

Figure 4.2.3 shows the model that added complexity and regularization to the baseline model. We do many times experiments to tune the model to find the best model. Finally, we find the model that contains two hidden layers, use l2 regularization, dropout, and early stopping to avoid overfitting. In this model, the accuracy of training and validation has sharply increased. The overfitting problem has efficiently controlled, but the overfitting problem still remaining but much better than the baseline model. And this model is the best model that we explored from the three models. We use this model to make the prediction, and we get 100% percent accuracy, it recognized all ten images correctly.

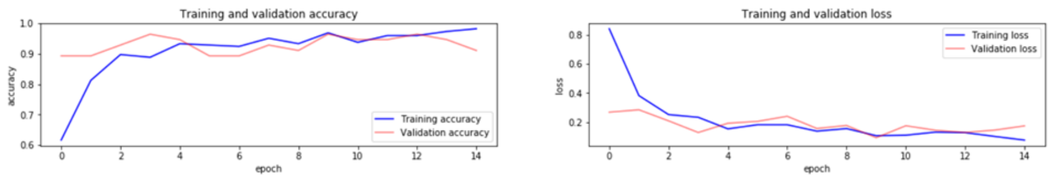

Figure 4.2.4

Figure 4.2.4 shows the VGG 16 transferring model. The accuracy of both training and validation is very good and pretty close. However, when we test it on the test set, we only achieve 50% accuracy.

b. Prediction

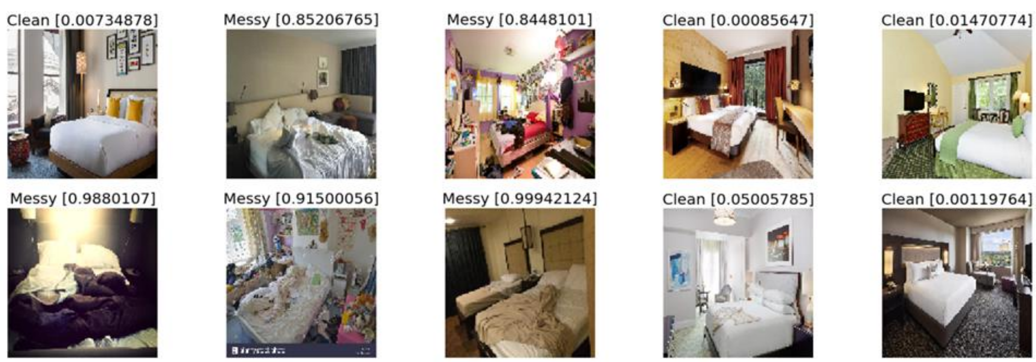

Figure 4.2.5

Figure 4.2.5 show the best result by model 2. All of the image's recognition are correct.

4.3. Comparison

Comparing the feature extracting and learning manually (feature engineering) and automatically by machine (CNN), we can notice although in a different way, both have successfully extracted the correct and machine-understandable features, and both of them performs well. However, it is difficult to compare the details of the two feature extractions. Because CNN is a very powerful and complex model, like a black box. It is difficult to interpret how the machine learns its characteristics. On the contrary, artificially extracted features are easier to understand for humans.

When we wanted to artificially extract image features, we found that there are too many features and it is difficult to find very good features. For example, the feature we manually extracted in this experiment is the statistical direction information of the pixels. But if there is a pot of flowers in a clean and tidy room, which contributes lots of irregular curves, the correct classification of the machine will be greatly affected. Because the machine does not understand the noise defined by humans and learning all the things it can learn.

Although the features extracted by CNN are hard to be interpreted, they must be deeper than the features we extracted artificially in the experiment. Therefore, with data increasing, we believe the performance of CNN will get better and better, and the model we only extracted one feature to classify will get worse and worse.

Moreover, in prediction, compared with the simple calculation with the parameter's matrix obtained by CNN, the artificially extracted feature needs to convert the image multiple times. the time complexity and computation are much increased, which is not convenient in the large database model task.

In conclusion, feature engineering focus on understanding the rule in the object, extracting the features and translate them into the language that machines can understand. CNN model focuses on how to reduce the error to increase the effect of feature learning by tuning various parameters, such as learning rate, optimizer, convolutional layers and full connected layers. Therefore, in future research, we should take advantage of the benefit of both technologies. For instance, using feature engineering to extract noise from the data to eliminate, and then use CNN to perfectly learn the correct data features.

5. Future Work

5.1. What

This project is our initial attempt to explore image classification domain. Although we get a pretty good model to predict all test set correct, it is still having a long distance for a complex image recognition. Classify a messy room in a complex environment is our goal, such as the model can recognize what are the wallpaper, furniture, and toys. To classify a messy room is a very tough task. For example, an old wooden room that looks dark and the tree pattern looks like the rubbish on the floor.

Currently, we use concentrate on marking if the bed was messy. For our future exploring process will gradually enlarge the model detect area from bed to the whole room. Also, we need to do much more feature selection to train the model. When we trained the model the overfitting problem always the obstacle for the project, just like the VGG 16 transfer learning model that we get a perfect model, but a garbage outcome.

5.2. How

Set a goal to achieve is easy, but the process is difficult. To avoid overfitting problem, we learned some effective methods during class, we have applied some of them to limit overfitting. But the result doesn't match our goals. So, the most optimized method would be collected large enough data to support the machine learning.

6. Implication

Roomba a top-rated AI product that can reduce the housework burden; we can remote the product by controller or phone. Conveniently, the product can operate by a schedule. But through recently similar products introduced, there is still an insufficient function that cannot satisfy the AI requirement that recognizes a messy room and sweeps the floor automatically. Based on this situation, we are encouraged to use machine learning to classify a messy room. Driverless cars technology can provide us enough skill support in our project.

7. Reference

[1]. Cai, Jie, et al. "Feature selection in machine learning: A new perspective." *Neurocomputing* 300 (2018): 70-79.

[2]. Cengıl, Emine, and Ahmet Çinar. "Multiple Classification of Flower Images Using Transfer Learning." *2019 International Artificial Intelligence and Data Processing Symposium (IDAP)*. IEEE, 2019.

[3]. Golodov, Valentin A., Mariya S. Dubrovina, and Anastasiya S. Paziy. "Transfer Learning Approach to Fine-Grained Image Classification." *2019 International Russian Automation Conference (RusAutoCon)*. IEEE, 2019.