Bouygues Floor Plan Detection Final Report

Kaihong Wang, Chi Zhang, Qitong Wang, Jiangshan Luo
{kaiwkh, czhang1, wqt1996, Jasonluo}@bu.edu
Boston University Department of Computer Science

1. Project Task

The goal of this project is to develop a machine learning model to estimate each type of the room given its floor plan. More precisely, the objectives are:

- Create a model that reads a PDF floor plan and convert it into an image floor plan (see Fig 1.) and its segmentation image (see Fig 2.).
- Find each room contour in a given image floor plan.
- Feature extraction: Implement a model calculates the surface area of each room, the compactness of each room and the density of non-white pixels in each room (Note: the white color is the background of image floor plan), the adjacent rooms number of each room.
- Design different machine learning algorithms to detect the type of each room, such as living room, kitchen, and bedroom and compare their accuracy and analyze these results.
- Output all predictions of the type of rooms and saves them in a CSV file (see Table 1.).

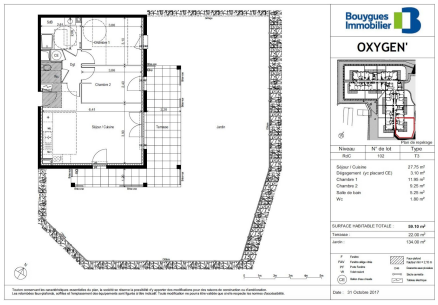

Figure 1. Input PDF Floor Plan

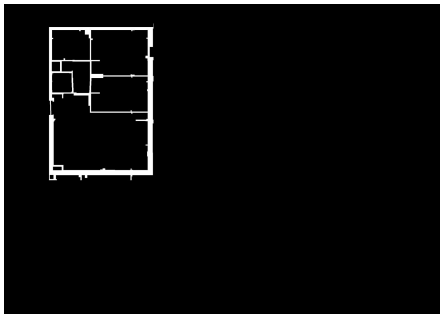

Figure 2. Input Segmentation Floor Plan

Room ID	Area	Compactness	Density	Adjacent room number	Type of Room
OXYplan_010116_room1	13.27398554	12.04696651	9.93E-05	3	1

Table 1. Output Table Template

2. Related Work

Some previous works used deep neural networks to gain important information from floor plans [1] [3]. Then they implemented an integer programming aggregated junctions into a set of simple primitives (which are the contour of the floor plan). However, we would like to try a different approach. For room type detection, previous works mainly had two strategies: one way is using a neural network with a set of primitives [1]; the other way is creating a text layer and extracting text information from the input image [4].

3. Dataset and Metric

3.1 Dataset

Bouygues Group provided 133 training floor plans, which contain two buildings “Eko” and “Equation” with 49 and 84 pdf floor plan files, respectively, along with the training dataset ground truth table. The testing dataset contains 139 floor plans from the “Oxygen” building. (We found that in the) The template table for the testing dataset result is provided, however, the table requires us to manually fill in the floor plan information (Number of Rooms, Surface Areas, etc) for each apartment. Note that all words and labels in the training and testing floor plans are in French. Bouygues Group also provided a reference room type chart, which translates French (some with abbreviations) to English. For example, (see Table 2.).

French	Chambre	ch	ch1
English	Bedroom	Bedroom	Bedroom

Table 2. French Reference Table

Besides, Bouygues Group provided a template for the output of floor detection, where 29 types of rooms are mandatory targets and 20 room's types are optional tasks.

For the type of rooms, we defined 6 kinds of labels: Living room, Bedroom, Bathroom, WC, Intersection and Kitchen. The Y labels correspond to room's type are as follows(See Table 3.).

Since the original type of room in image floor plans is French, so we need to translate from French to English by ourselves. And then labeling room type to CSV file as our ground truth.

Type of Room	Y label
Living Room	0
Bedroom	1
Bathroom	2
WC	3
Intersection	4
Kitchen	5

Table 3. Label Reference Table

3.2 Metrics

For room type detection, we calculate the accuracy of prediction of room types and drawing its confusion matrix. The accuracy function is defined:

$$Accuracy(yoom's\ type) = \frac{Num(true)}{Num(true) + Num(false)} \quad (m1)$$

4. Approach

4.1 Features Definition

Note that our goal is to estimate the type of rooms given a floor plan, so we need to extract several discriminative features of rooms to improve the classification performance of the machine learning models later.

Our approach integrates some classifier models to predict the type of each room.

4.1.1 Relative Area

We define the area of the room as a feature which we computed below using OpenCV since we notice that some rooms, for example, living rooms, are always larger than other room, which makes it a quiet discriminative indicator.

Preliminarily, we planned to build a contour detector which extracts each room from the segmentation image file of the floor plan and calculates the pixel surface area by using the cv2.contourArea() function from OpenCV[5]. The definition of one room's relative area sees F1.

$$Relative\ Area = \frac{Pixel\ Room\ Area}{Sum\ of\ Pixel\ room\ area\ in\ a\ floor\ plan}$$

F1. Relative Area Definition

4.1.2 Compactness

For the compactness, we compute it by using the square of relative perimeter divided by the area. This feature can indicate the geometric properties of rooms, which plays an important role in the process of prediction of rooms' type.

In order to get a room's compactness, we need to get its perimeter. For convenience, we define a room's relative perimeter (see F2.).

The definition of compactness for a polygon sees F2.

$$Relative\ Perimeter = \frac{Pixel\ Room\ Perimeter}{Sum\ of\ Pixel\ room\ perimeter\ in\ a\ floor\ plan}$$

F2. Relative Perimeter Definition

To get the room's pixel room perimeter, we need to use the cv2.arcLength() function from OpenCV[6].

Assuming one room is a polygon, the definition of the compactness of it sees F3.

$$Compactness\ of\ Room = \frac{(Relative\ \ Room\ Perimeter)^2}{Relative\ Room\ Area}$$

F3. Compactness Definition

4.1.3 Density of Room

We believe that rooms with higher density have more furniture or other appliances, which indicates the type of room potentially. Using OpenCV, we pick a room's information from the original floor plan (See Fig 3 and 4.). Then we traverse the room image's all pixels, calculating room's number of non-white pixels and total pixels.

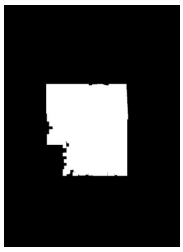

Figure 3. An example of a room's contour

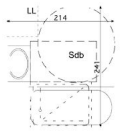

Figure 4. An example of a room's information

We define the density of the room(sees F4).

$$Room\ density = \frac{Num(non-white\ pixels)}{Num(total\ pixels)}$$

F4. Room Density Definition

4.1.4 Adjacent Room Numbers

We found that for some types of room, such as intersection, have many adjacent rooms (see Fig 5.). In addition, adjacent room numbers can tell a room's relative location. In order to get a room's adjacent room number, we used the OpenCV tools.

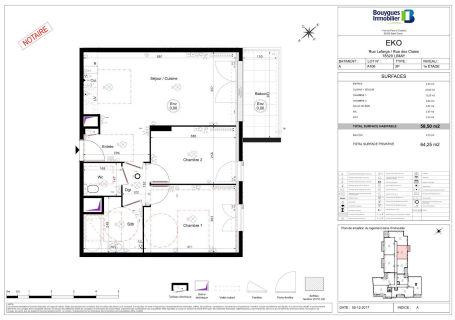

Figure 5. An example of intersection's (Dgt) adjacent room

4.2 Room Type Classification

For detecting the type of each room, we decide to use Multi-class SVM and Backpropagation Neural Network (BPNN). Our BPNN model uses a multiclass softmax regression and its loss layer uses cross entropy loss, where the hypothesis and the loss function defined as follow [7]:

$$p(y^{(i)} = j|x^{(i)};\theta) = \frac{e^{\theta_j^T x^{(i)}}}{\sum_{l=1}^{k} e^{\theta_l^T x^{(i)}}}$$

F6. Hypothesis Function

$$J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}\sum_{j=1}^{k}[1\ \{y^{(i)} = j\}\ log(\frac{e^{\theta_j^T x^{(i)}}}{\sum_{l=1}^{k} e^{\theta_l^T x^{(i)}}})]$$

F7. Loss Function without regularization

Note that in F7, 1{.} is the indicator function, where 1{a true statement} = 1, and 1{a false statement} = 0.

For the NN room's type model, we take extracted features above as input, predict the input data as one of the six classes through a 3-layer neural network

Multi-class SVM also takes extracted features above, we built six independent SVM with Gaussian kernel, whose formula of optimization is shown above. Each of these SVMs is a one vs. rest SVM model, and we take the class with the highest probability among six output of these models as the class of the input data

To implement a one vs. rest SVM models, we need to reconstruct the label of our input data so that only the corresponding class of the model is labeled as positive class and others are labeled as negative classes.

Note that the one vs. rest model may cause imbalanced data problem, which means every model in the total K independent models is very likely to receive K-1 times more negative training data than positive training data. To cope with such problem, we need to set the negative label from -1 to -(1/K) to rebalance data.

$$min_{w,\xi,b}\ \tfrac{1}{2}||w||^2 + C\sum_{n=1}^{N}\xi_n$$

$$subject\ to \quad y_n\hat{y}(x_n) \geq 1-\xi_n, \quad n = 1,...,N$$

$$\xi_n \geq 0, \quad n = 1,...,N$$

F8. The formula of optimization of SVM

5. Results

5.1 Features Extraction

We used the manual labeled segmentation images to extract the segmented floor plan from the original image (Fig 6.).

Firstly, we used erosion and dilation methods in OpenCV to extract all contours of a given segmented image, labeling the walls of the floor plan as black color and the rest of the floor plan image as white color.

Secondly, we manually draw the black lines for the doors of each room in order to get complete segmentation images of floor plans.

Finally, we used the OpenCV tools to extract all the aimed features we wanted and store all of them and their corresponding manual labeled ground truths in order to input these features and ground truth (GT) to train or test our machine learning models (See Fig 7.).

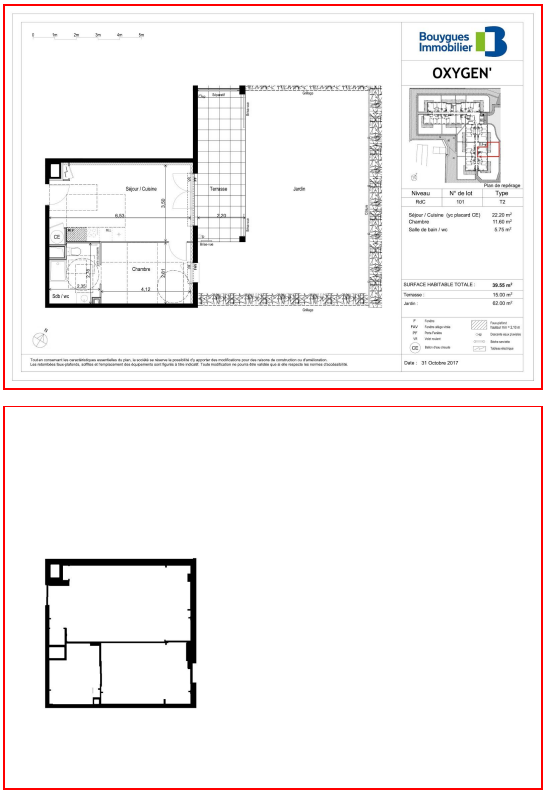

Figure 6. Input floor plan(up), segmented floor plan(down)

Id	area	compactness	density	adjacentRooms	y_label
lot_plan_01A001_0001_room1	18.8276568	9.217192433	5.25E-05	1	1
lot_plan_01A001_0001_room2	2.54164621	16.5457648	0.000198893	3	3
lot_plan_01A001_0001_room3	20.15557649	9.452352627	9.09E-05	4	1
lot_plan_01A001_0001_room4	44.40724781	12.88893322	6.58E-05	4	0
lot_plan_01A001_0001_room5	4.899322665	24.20203308	8.99E-05	4	4
lot_plan_01A001_0001_room6	7.436698332	13.08521978	0.000235108	2	2
lot_plan_01A002_0001_room1	1.493404846	9.973287041	0.000199075	3	4
lot_plan_01A002_0001_room2	3.072518733	11.14816781	0.000384983	1	2
lot_plan_01A002_0001_room3	5.947910052	8.388729433	0.000176585	2	5
lot_plan_01A002_0001_room4	6.756504064	13.04852028	0.000269073	3	2
lot_plan_01A002_0001_room5	12.15539654	6.735496165	7.21E-05	3	1
lot_plan_01A002_0001_room6	17.44930641	11.16020542	9.77E-05	3	1
lot_plan_01A002_0001_room7	34.36341275	8.699752984	5.48E-05	5	0
lot_plan_01A002_0001_room8	16.5873005	8.512771946	5.74E-05	2	1
lot_plan_01A003_0001_room1	6.635943299	13.8324434	0.000195537	3	2
lot_plan_01A003_0001_room2	15.15633401	6.698096528	0.000113952	3	1
lot_plan_01A003_0001_room3	1.759276616	10.22394686	8.92E-05	5	4
lot_plan_01A003_0001_room4	2.728826198	7.80238074	0.000277951	3	3
lot_plan_01A003_0001_room5	13.30425836	7.131422011	8.00E-05	3	1
lot_plan_01A003_0001_room6	37.66938137	12.29819093	8.85E-05	5	0
lot_plan_01A003_0001_room7	3.361976687	8.382273269	0.00033655	2	2
lot_plan_01A003_0001_room8	16.74439999	7.390066473	6.39E-05	2	1
lot_plan_01A004_0001_room1	3.359378603	10.21438946	0.000299707	2	2
lot_plan_01A004_0001_room2	16.80638203	9.112058368	6.34E-05	2	1
lot_plan_01A004_0001_room3	38.58447634	10.0395594	0.000109563	5	0

Figure 7. Eko feature and GT example of CSV file

5.2 Machine Learning Models Results

At first we selected three hand-crafted features: relative area, compactness and density of room to train our machine learning model, BPNN and multi-class SVM. We use parameter tuning to get the best hyper-parameters and run our BPNN and SVM model. Results are shown below.

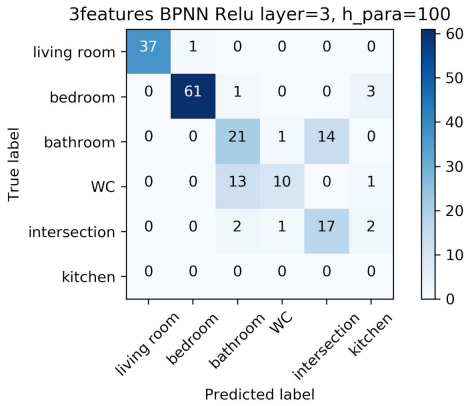

Figure 7. Confusion matrix of BPNN given 3 features after tuning parameters

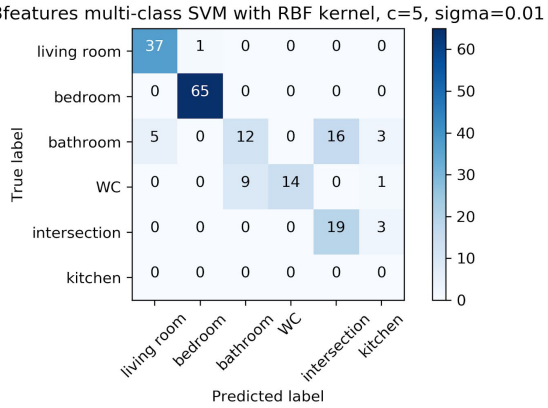

Figure 8. Confusion matrix of multi-class SVM given 3 features after tuning parameters

As we can observe above, our models can identify living room and bedroom accurately, but they cannot distinguish bathroom, WC and intersection very well. Motivated by difference between them, we defined a new feature, number of adjacent rooms as we know that intersection always have most adjacent rooms than others, and bathroom is very likely to have least adjacent rooms because they always located at the inside of bedroom.

After adding new feature into our dataset, we retrain our models with tuned parameters and results are shown below.

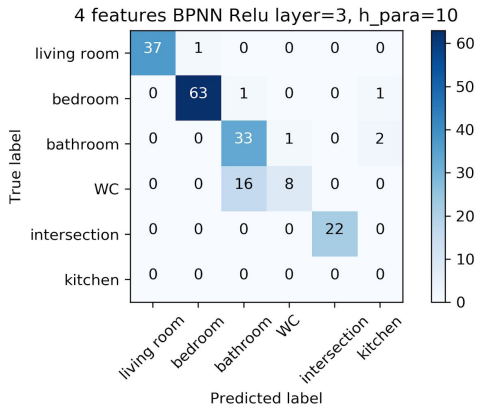

Figure 9. Confusion matrix of BPNN given 4 features after tuning parameters

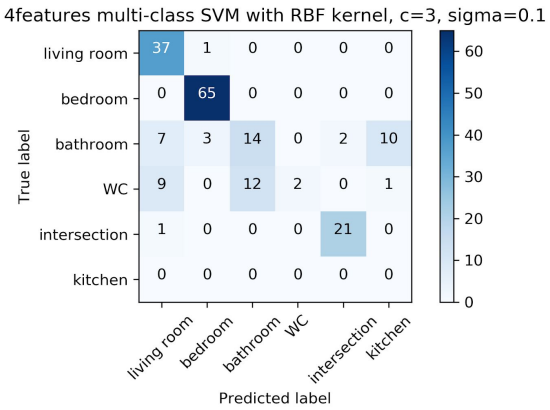

Figure 10. Confusion matrix of multi-class SVM given 4 features after tuning parameters

6. Analysis of Machine Learning Models

(1) The accuracy of SVM from 3 features to 4 features (See Fig 15, 16, 17, 18 in Appendix) does not improve too much. Some room types' prediction, such as intersection, whose adjacent room number is quite bigger than other room, can be improved from 3-feature SVM model to 4-feature SVM model. However, other room types' prediction, such as living room and bathroom, whose adjacent room number is quite similar, can be worse from 3-feature SVM model to 4-feature SVM model.

(2) The accuracy of BPNN from 3 features to 4 features (See Fig 11, 12, 13, 14) get improved. More features can describe each room's information more completely, which can cause the positive effect on the training process of BPNN's parameters.

(3) When predicting room's type, linear SVM's (See 16, 18) performance is worse than the RBF SVM (See 15, 17), and the BPNN's performance with sigmoid activation function is almost like the BPNN's performance with ReLU activation function. In the case of linear inseparability that our data are, RBF SVM is significantly better than linear SVM.

(4) With the increasing of sigma parameters, the variance of REF SVM decreases (See 15, 17.).

(5) With the increasing of neuron's number, the variance of BPNN increases. The variance's increasing of ReLU BPNN is much bigger than that of Sigmoid BPNN (See 11, 12, 13, 14.).

(If you want to see the SVM's and BPNN's process of adjusting parameters, please refer to the appendix.)

7. Conclusion

We proposed automatic models that extracted the features from the original and segmented floor plan and predict type of room base on them.

In terms of the prediction models, we ran several back-propagation network (BPNN) and support vector machine (SVM) models with different configurations. As a result, our models successfully detected the room types in most cases. Specifically, we found the BPNN models work significantly well for the room type prediction. The best model we have so far, 4-feature BPNN with Sigmoid, has reached the accuracy above 80%. The performance of our SVM models, on the other hand, do not seem to be promising under the current circumstance.

References

1. *Liu, C., Wu, J., Kohli, P., & Furukawa, Y. (2017). Raster-to-Vector: Revisiting Floorplan Transformation. 2017 IEEE International Conference on Computer Vision (ICCV). doi:10.1109/iccv.2017.241*
2. *Dodge, S., Xu, J., & Stenger, B. (2017). Parsing floor plan images. 2017 Fifteenth IAPR International Conference on Machine Vision Applications (MVA). doi:10.23919/mva.2017.7986875*
3. *Ahmed, S., Liwicki, M., Weber, M., & Dengel, A. (2011). Improved Automatic Analysis of Architectural Floor Plans. 2011 International Conference on Document Analysis and Recognition. doi:10.1109/icdar.2011.177*
4. *Ahmed, S., Liwicki, M., Weber, M., & Dengel, A. (2012). Automatic Room Detection and Room Labeling from Architectural Floor Plans. 2012 10th IAPR International Workshop on Document Analysis Systems. doi:10.1109/das.2012.22*
5. *Structural Analysis and Shape Descriptors¶. (n.d.). Retrieved from https://docs.opencv.org/2.4/modules/imgproc/doc/structural_analysis_and_shape_descriptors.html?highlight=contourarea#cv2.contourArea*
6. *Structural Analysis and Shape Descriptors¶. (n.d.). Retrieved from*

https://docs.opencv.org/2.4/modules/imgproc/doc/structural_analysis_and_shape_descriptors.html?highlight=arclength#cv2.arcLength

7. *Softmax Regression. (n.d.). Retrieved October 29, 2018, from http://ufldl.stanford.edu/wiki/index.php/Softmax_Regression*

Appendix

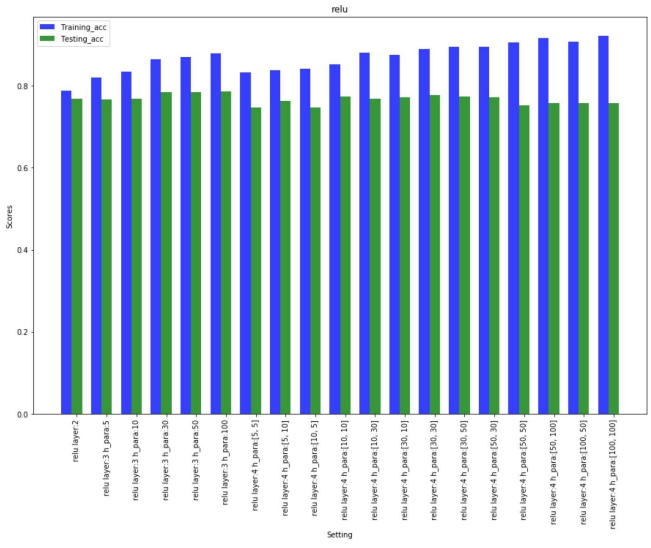

Figure 11. Accuracy of 3-feature BPNN with ReLU

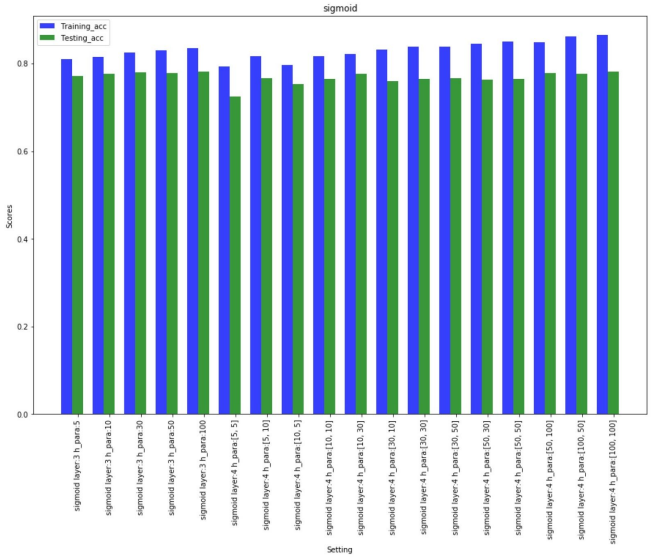

Figure 12. Accuracy of 3-feature BPNN with Sigmoid

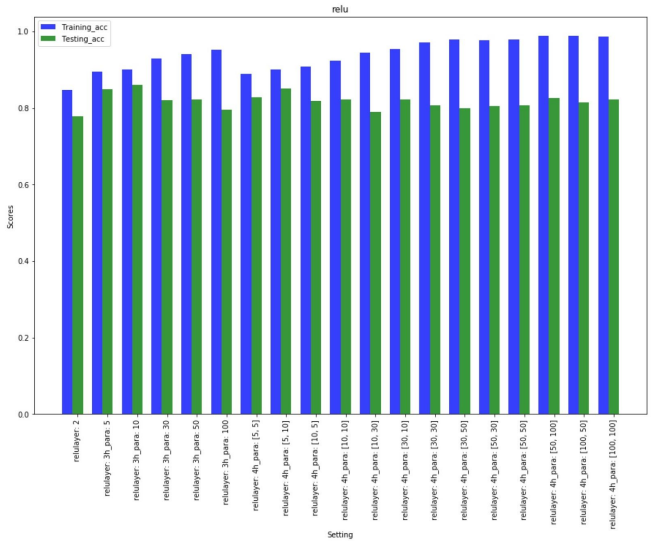

Figure 13. Accuracy of 4-feature BPNN with ReLU

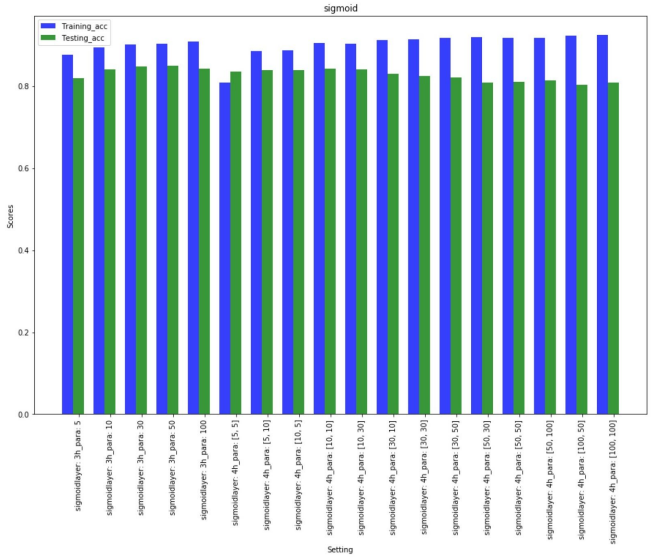

Figure 14. Accuracy of 4-feature BPNN with Sigmoid

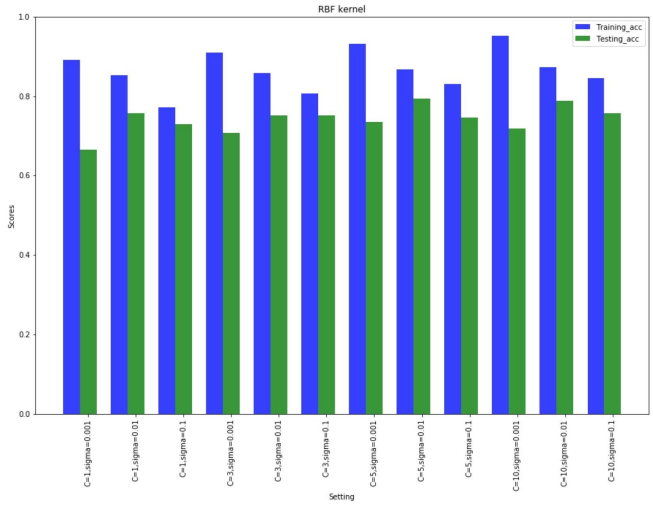

Figure 15. Accuracy of 3-feature REF SVM

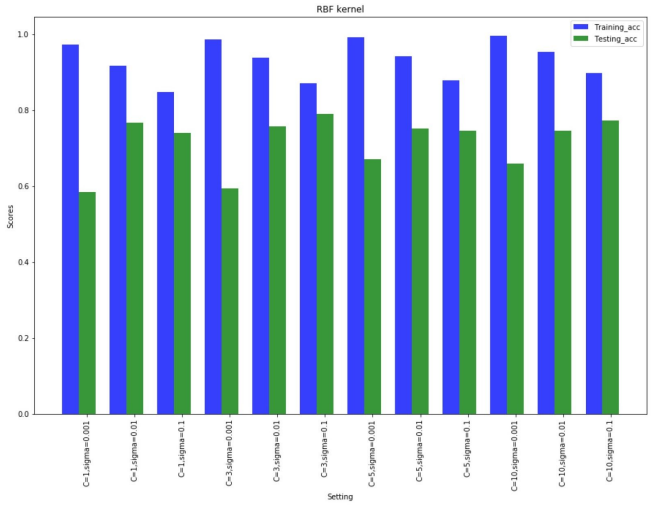

Figure 17. Accuracy of 4-feature REF SVM

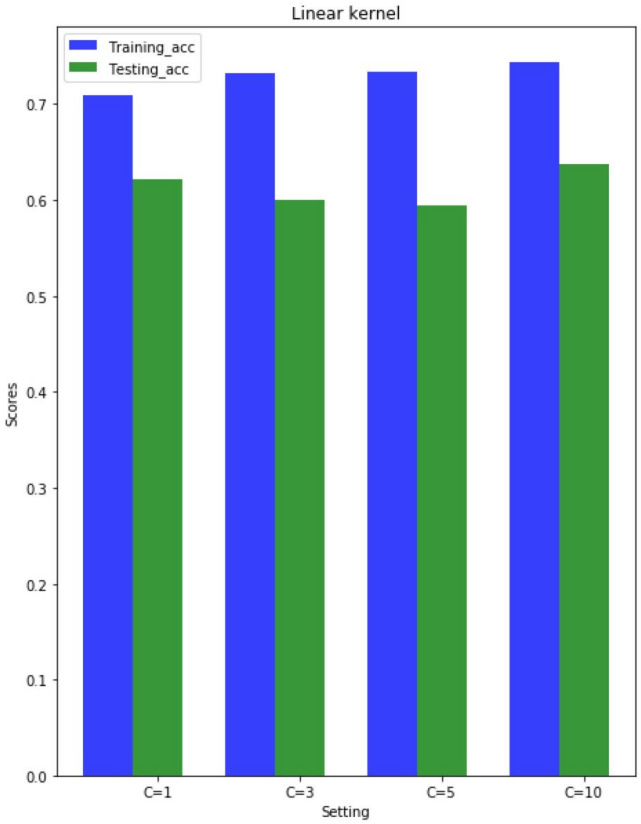

Figure 16. Accuracy of 3-feature Linear SVM

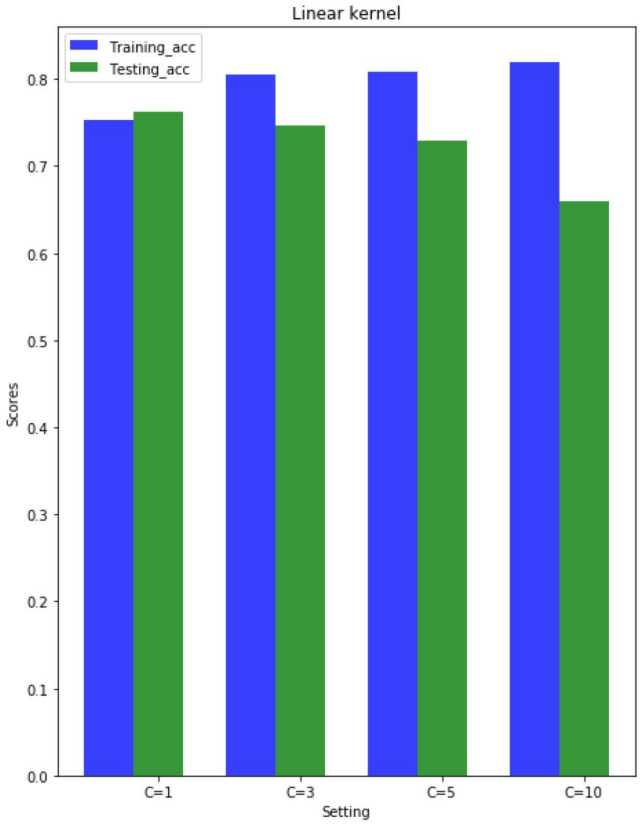

Figure 18. Accuracy of 4-feature Linear SVM