Exercise 1. Review the material on Bishop 3.1, 3.1.1., 3.1.2, 3.1.4 and the material on Lagrangian multipliers (especially KKT conditions). Show that the minimization of the regularized error function (3.29) is equivalent to minimizing the unregularized sum-of-squares error (3.12) subject to the constraint (3.30). Discuss the relationship between λ and η.

From the equation,

$$\frac{1}{2}\sum_{n=1}^{N}\{t_n - \mathbf{w}^{\mathrm{T}}\phi(\mathbf{x}_n)\}^2 + \frac{\lambda}{2}\sum_{j=1}^{M}|w_j|^q \qquad (3.29)$$

In order to minimize

$$E_D(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\{t_n - \mathbf{w}^{\mathrm{T}}\phi(\mathbf{x}_n)\}^2. \qquad (3.12)$$

with the constraint,

$$\sum_{j=1}^{M}|w_j|^q \leqslant \eta \qquad (3.30)$$

and then we can get,

$$E_w(w) = \frac{1}{2}\left(\sum_{j=1}^{M}|w_j|^q - \eta\right) \leq 0 * \frac{1}{2}$$

We use the lagrangian multipliers with KKT conditions.

$$E_D(\mathbf{w}) + \lambda E_W(\mathbf{w})$$

Data term + Regularization term

$$g(\mathbf{x}) \geqslant 0 \qquad (E.9)$$
$$\lambda \geqslant 0 \qquad (E.10)$$
$$\lambda g(\mathbf{x}) = 0 \qquad (E.11)$$

$L(w,\lambda) = E_D(w) + \lambda E_w(w)$

$$=\frac{1}{2}\sum_{n=1}^{N}\{t_n - w^T\phi(x_n)\}^2 \quad + \frac{\lambda}{2}\left(\sum_{j=1}^{M}|w_j|^q - \eta\right)$$

From E.11, we know that $\frac{\lambda}{2}\left(\sum_{j=1}^{M}|w_j|^q - \eta\right)$ equals to zero. And we get

$$\sum_{j=1}^{M}|w_j|^q = \eta$$

where

$$\mathbf{w} = \left(\lambda \mathbf{I} + \mathbf{\Phi}^{\mathrm{T}}\mathbf{\Phi}\right)^{-1}\mathbf{\Phi}^{\mathrm{T}}\mathbf{t}.$$

So it can be written as

$$\sum_{j=1}^{M} |w_j(\lambda)|^q = \eta$$

if λ is large, wj are close to zero, which means sparse model and the corresponding basis functions play no role.

Exercise 2. Review the material on Chapter 4 of Bishop's book. Given a set of data points $\{x_n\}$, we can define the convex hull to be the set of all points x given by

$$x = \sum_n \alpha_n x_n$$

where $\alpha_n \geq 0$ and $\sum_n \alpha_n = 1$. Consider a second set of points $\{y_n\}$ together with their corresponding convex hull. By definition, the two sets of points will be linearly separable if there exists a vector w and a scalar w_0 such that $w^T x_n + w_0 > 0$ for all x_n, and $w^T y_n + w_0 < 0$ for all y_n. Show that if their convex hulls intersect, the two sets of points cannot be linearly separable, and conversely that if they are linearly separable, their convex hulls do not intersect.

$x = \sum_{\mathrm{n}} a_n x_n$

For linear discriminant,

$F(x) = w^T x_n + w_0$

then we get $F(x) = w^T(\sum_{\mathrm{n}} a_n x_n) + +w_0 = \sum_{\mathrm{n}} a_n(w^T x_n + w_0)$ due to $\sum_{\mathrm{n}} a_n = 1$

If convex hulls intersect, then there must be a point between {x} and {y} hull, called xy

$$F(y) = \sum_{\mathrm{m}} b_m(w^T y_m + w_0)$$

$$F(xy) = \sum_{\mathrm{n}} a_n(w^T x_n + w_0) = \sum_{\mathrm{m}} b_m(w^T y_m + w_0)$$

For linear separable,

$F(x) = w^T x_n + w_0 > 0$

$F(y) = w^T y_n + w_0 < 0$

This means that discriminant bas to be larger than 0 and also smaller than 0, which is impossible. Therefore, Intersection cannot be linearly separable.

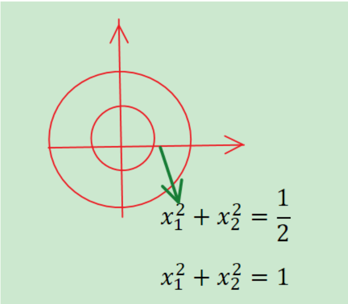
Exercise 3. Consider a binary classification problem. The data lies in the two-dimensional plane $\mathbf{x} \in \mathbf{R}^2$. The only positive example ($t = +1$) is at the origin $(0, 0)$. The are 8 negative examples ($t = -1$) equally spaced on a circle centered on the origin with radius R.

(a) Choose a three-dimensional feature vector $\phi(\mathbf{x})$ which enables the positive and negative examples to be separated by a linear hyperplane (in feature space). Note that since we are in two dimensions therefore $\mathbf{x} = (x_1, x_2)$. You will find a feature vector of the type $\phi(\mathbf{x}) = \phi(x_1, x_2) = (\phi_1, \phi_2, \phi_3)$. Use these feature vectors to calculate a kernel $K(\mathbf{x}_n, \mathbf{x}_m)$ for this problem.

(b) Show that the classifier in feature space can be expresses using kernels as $\text{sign}(g(\mathbf{x}))$ where

$$g(x) = \sum_{n=1}^{8} \alpha_n t_n K(\mathbf{x}, \mathbf{x}_n) + \alpha_0 t_0 K(\mathbf{x}, \mathbf{0}) + b,$$

where $\{\mathbf{x}_n, n = 1, \ldots, 8\}$ are the data points on the circle. What are the support vectors? (The answer will depend on your choice of feature vectors).



There are 8 points on the circle $({x_1}^u, {x_2}^u) = (\cos\left(pi * \frac{u}{4}\right), \sin\left(pi * \frac{u}{4}\right))$

u=0,1,2,..8

We can use surface $x_1^2 + x_2^2 = \frac{1}{2}$ to separate the central point and the outer circle.

So tge classifier will be sign {(0,0,1)* $\vec{\phi}(x_1, x_2, x_3)$-1/2}

And then the kerel will be K= $\vec{\phi}(x_1, x_2) * \vec{\phi}(x_1', x_2')$=

$$x_1 x_1' + x_2 x_2' + (x_1^2 + x_2^2)(x_1'^2 + x_2'^2)$$

Therefore, our classifier in feature space will be

sign((1/8)$\sum_{n=1}^{8} K \quad - 1/2)$

(using $\sum_{n=1}^{8} {x_n}^u = 0$)

And the support vector is on the 8 points $({x_1}^u, {x_2}^u)$ and central point

EX4.

(See readme.md for the result table)

Results,

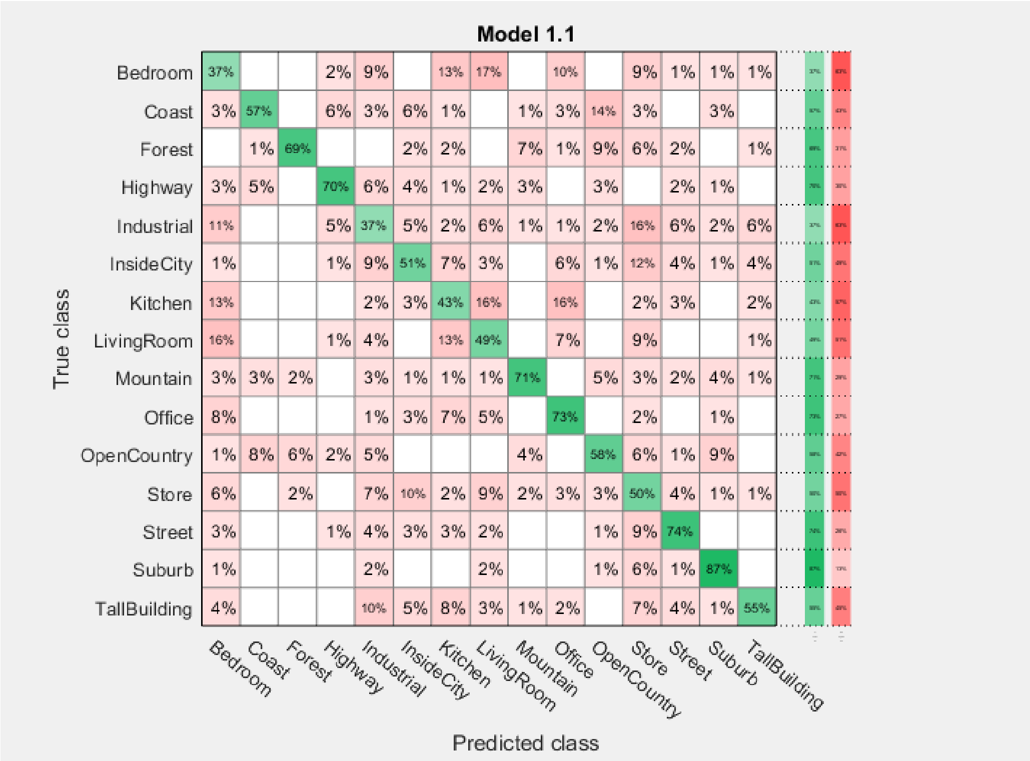
Tiny images representation and nearest neighbor classifier (Accuracy (mean of diagonal of confusion matrix) is 0.199)

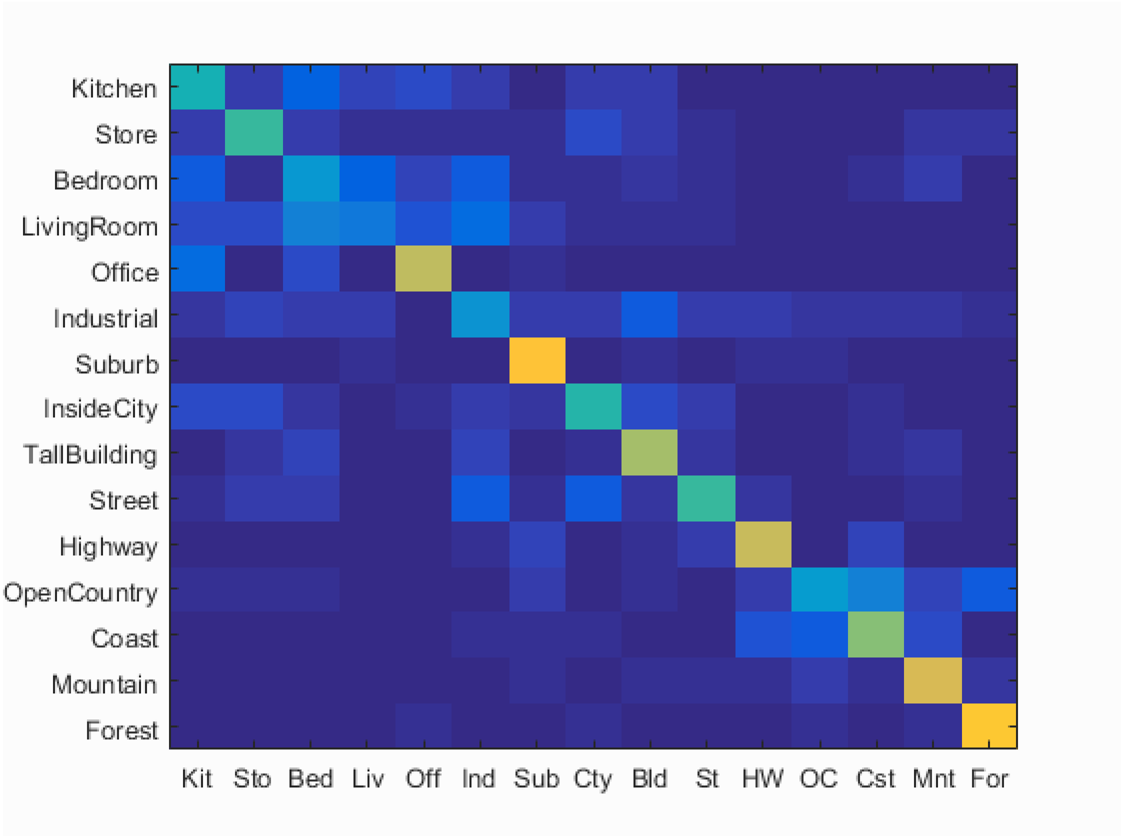
Bag of SIFT representation and nearest neighbor classifier (when K=1, Accuracy is 41.5%)

Bag of SIFT representation and linear SVM classifier (accuracy of 58.7%)

1.1	SVM	Last change: Linear SVM	Accuracy: 58.7%	128/128 features
1.2	SVM	Last change: Quadratic SVM	Accuracy: 58.2%	128/128 features
1.3	SVM	Last change: Cubic SVM	Accuracy: 57.5%	128/128 features
1.4	SVM	Last change: Fine Gaussian SVM	Accuracy: 10.3%	128/128 features
1.5	SVM	Last change: Medium Gaussian SVM	Accuracy: 58.6%	128/128 features
1.6	SVM	Last change: Coarse Gaussian SVM	Accuracy: 55.9%	128/128 features
2.1	KNN	Last change: Fine KNN	Accuracy: 41.5%	128/128 features
2.2	KNN	Last change: Medium KNN	Accuracy: 42.3%	128/128 features
2.3	KNN	Last change: Coarse KNN	Accuracy: 36.7%	128/128 features
2.4	KNN	Last change: Cosine KNN	Accuracy: 50.3%	128/128 features
2.5	KNN	Last change: Cubic KNN	Accuracy: 37.3%	128/128 features
2.6	KNN	Last change: Weighted KNN	Accuracy: 44.9%	128/128 features
3	SVM	Last change: 'MulticlassMethod' = 'OVA'	Accuracy: 54.4%	128/128 features

Bag of SIFT representation with Machine learning algorithm (last one is 1 vs all SVM) 5 fold cross-validation results





But the result is not satisfied, since there is better result on the paper, "**Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories**" with the method of SPM. And the paper shows that it can achieve almost 100 percent for the suburb. I try many codes on the website to see whether they can outperform higher accuracy. And it turns out that some of the code cannot run on my computer even it suggests that it has 80% accuracy. When the code is running, it exceed the matrix maximum limit of the system, and even cause my computer went black. I also try the SPM algorithm with the code provide on the github with some modification. Finally the result is surprisingly good.

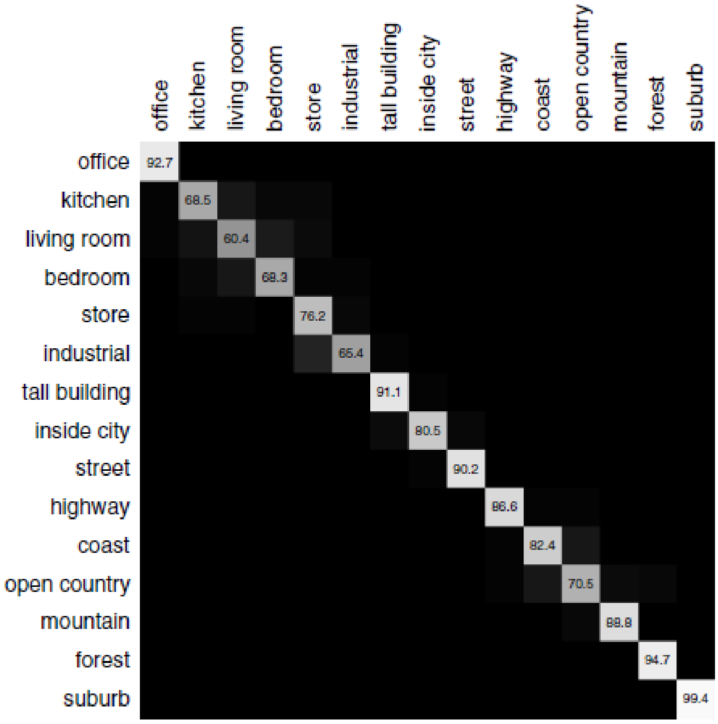
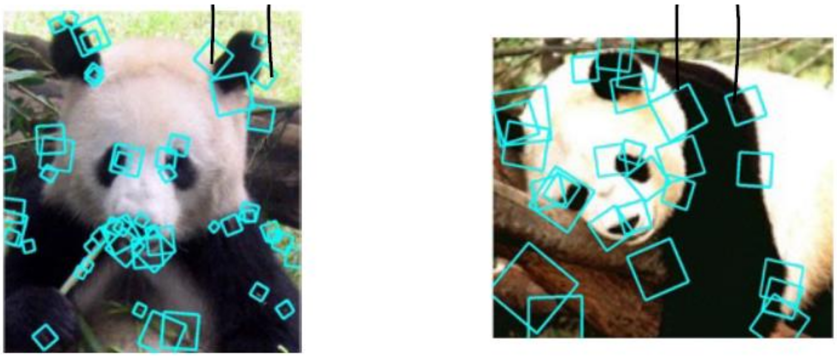


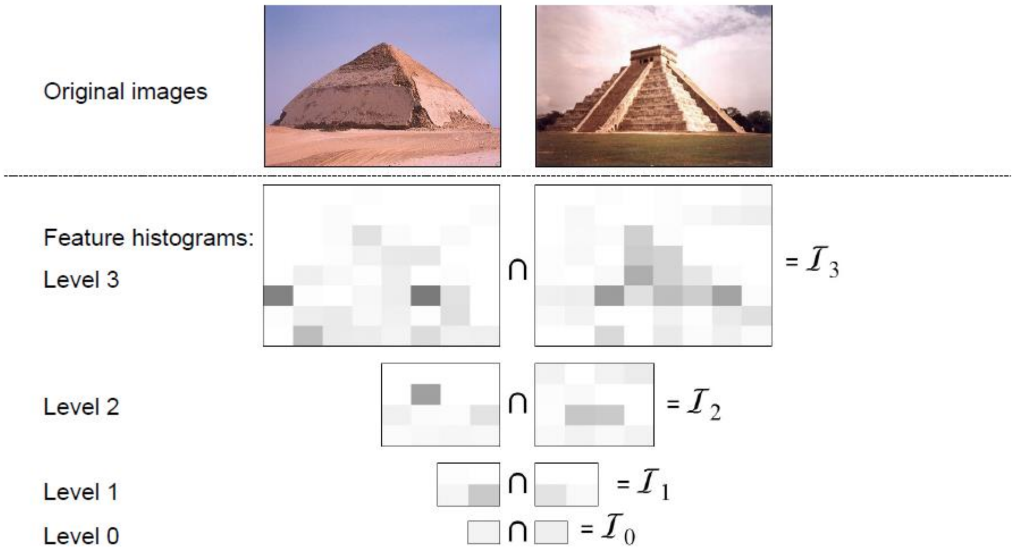
Figure 3. Confusion table for the scene category dataset. Average classification rates for individual classes are listed along the diagonal. The entry in the ith row and jth column is the percentage of images from class i that were misidentified as class j.

Paper result with SPM

The spatial pyramid matching can distinguish the picture with orderless bag of feature, and the numbers of feature can be different in training and testing data set. Basically, they use pyramid kernel in the BOW, which includes three steps. First, they subdivide the image at three different levels of resolution. And then they count the features which cluster into each spatial bin for each level of resolution. Finally, they weight each spatial histogram as the graph below.



$$\mathbf{X} = \{\vec{\mathbf{x}}_1, \ldots, \vec{\mathbf{x}}_m\} \qquad \mathbf{Y} = \{\vec{\mathbf{y}}_1, \ldots, \vec{\mathbf{y}}_n\}$$
$$\vec{\mathbf{x}}_i \in \Re^d \qquad \vec{\mathbf{y}}_i \in \Re^d$$



Total weight (value of *pyramid match kernel*): $\mathcal{I}_3 + \frac{1}{2}(\mathcal{I}_2 - \mathcal{I}_3) + \frac{1}{4}(\mathcal{I}_1 - \mathcal{I}_2) + \frac{1}{8}(\mathcal{I}_0 - \mathcal{I}_1)$

	Bedroom	Coast	Forest	Highway	Industrial	InsideCity	Kitchen	LivingRoom	Mountain	Office	OpenCountry	Store	Street	Suburb	TallBuilding
Bedroom	0.44	0.32	0.00	0.02	0.02	0.01	0.06	0.05	0.01	0.01	0.08	0.01	0.01	0.00	0.00
Coast	0.00	0.76	0.11	0.02	0.00	0.00	0.00	0.00	0.02	0.00	0.10	0.00	0.00	0.00	0.00
Forest	0.00	0.00	0.95	0.00	0.00	0.00	0.00	0.00	0.04	0.00	0.01	0.00	0.00	0.00	0.00
Highway	0.01	0.03	0.04	0.67	0.14	0.04	0.00	0.00	0.03	0.00	0.03	0.02	0.01	0.00	0.01
Industrial	0.03	0.01	0.00	0.00	0.61	0.12	0.04	0.02	0.01	0.02	0.00	0.13	0.01	0.02	0.02
InsideCity	0.01	0.01	0.00	0.03	0.00	0.81	0.06	0.00	0.00	0.00	0.00	0.02	0.04	0.00	0.04
Kitchen	0.12	0.00	0.00	0.00	0.01	0.00	0.39	0.36	0.00	0.03	0.00	0.09	0.00	0.00	0.00
LivingRoom	0.04	0.02	0.02	0.02	0.01	0.00	0.07	0.30	0.45	0.01	0.05	0.03	0.01	0.00	0.00
Mountain	0.00	0.00	0.02	0.02	0.00	0.00	0.01	0.00	0.74	0.17	0.04	0.01	0.01	0.00	0.00
Office	0.01	0.10	0.01	0.01	0.00	0.00	0.01	0.02	0.02	0.37	0.46	0.00	0.00	0.00	0.00
OpenCountry	0.00	0.09	0.05	0.04	0.01	0.00	0.00	0.00	0.05	0.00	0.74	0.04	0.01	0.00	0.00
Store	0.01	0.00	0.01	0.00	0.07	0.01	0.05	0.04	0.01	0.01	0.00	0.80	0.00	0.00	0.01
Street	0.00	0.00	0.01	0.03	0.00	0.08	0.00	0.01	0.01	0.00	0.02	0.03	0.81	0.00	0.02
Suburb	0.00	0.00	0.01	0.00	0.01	0.01	0.00	0.01	0.01	0.01	0.00	0.01	0.01	0.68	0.27
TallBuilding	0.00	0.00	0.00	0.00	0.00	0.05	0.00	0.00	0.01	0.00	0.00	0.01	0.00	0.00	0.93

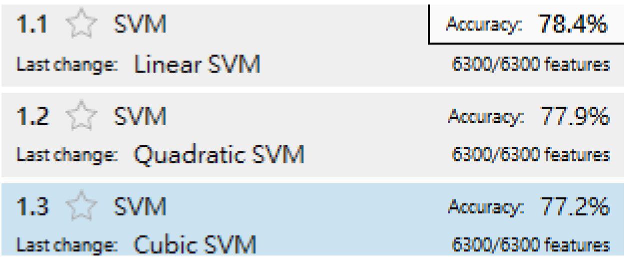
My result

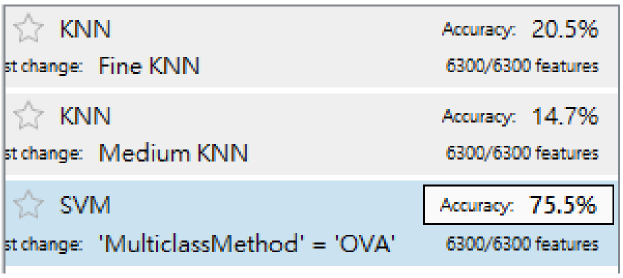
```
Classification using BOW rbf_svm
Accuracy = 74.1039% (2212/2985) (classification)

Classification using histogram intersection kernel svm
Accuracy = 77.1524% (2303/2985) (classification)

Classification using Pyramid BOW rbf_svm
Accuracy = 78.057% (2330/2985) (classification)

Classification using Pyramid BOW histogram intersection kernel svm
Accuracy = 82.7136% (2469/2985) (classification)
fx >>
```

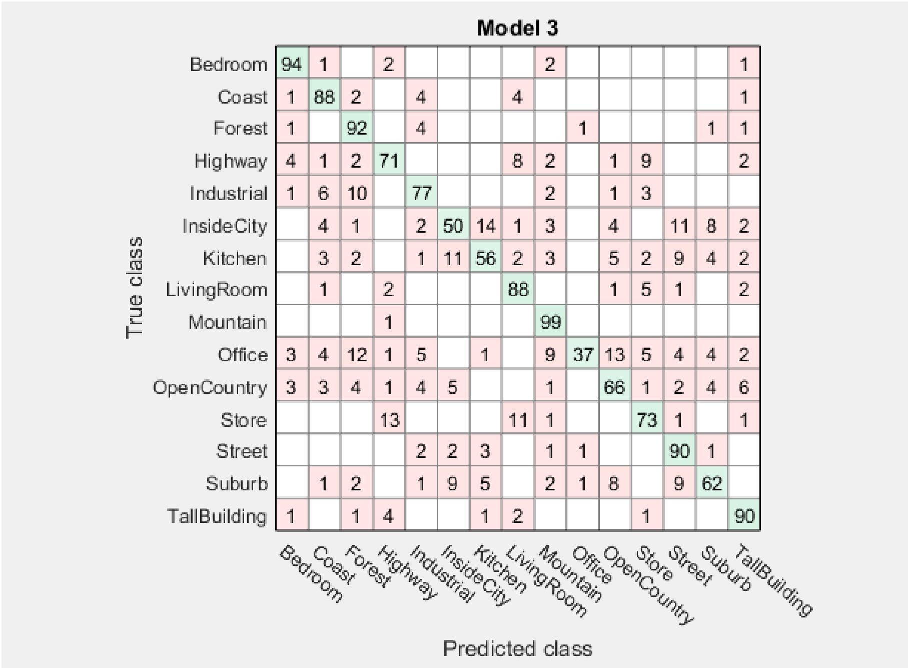




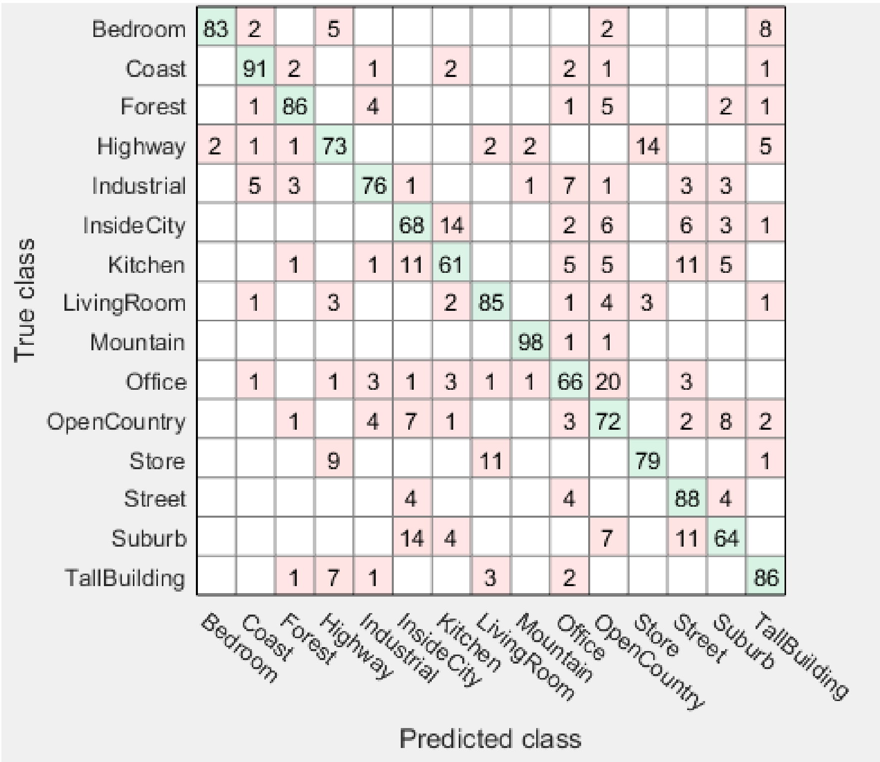
Pyramid bag of word with histogram intersection + different SVM (linear, quadratic, cubic)

5 fold cross-validation results

And the result shows that SVM is better than the KNN in clustering at the end of the image processing pipeline. But the most important part for high accuracy is the feature engineering, which means that better features (e.g. BOW SPM) will decide the overall accuracy.



SVM 75.5% confusion matrix (one vs all)



SVM 78.4% confusion matrix (one vs one)