Introduction to Data Mining: 21191441
Zhejiang University
Professor Deng Cai

May 27, 2019
Homework 3

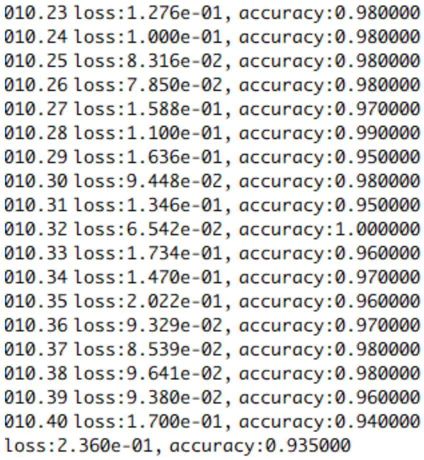
Homework 3

Collaborators:
Name: Shen Zijin
Student ID: 3160104734

Problem 3-1. Neural Networks

In this problem, we will implement the feedforward and backpropagation process of the neural networks.

(a) Answer: After training by such a 3-layer neural network, According to the output result in the console, the loss reaches about 0.236, and the accuracy reaches about 93.5%.



```
010.23 loss:1.276e-01, accuracy:0.980000
010.24 loss:1.000e-01, accuracy:0.980000
010.25 loss:8.316e-02, accuracy:0.980000
010.26 loss:7.850e-02, accuracy:0.980000
010.27 loss:1.588e-01, accuracy:0.970000
010.28 loss:1.100e-01, accuracy:0.990000
010.29 loss:1.636e-01, accuracy:0.950000
010.30 loss:9.448e-02, accuracy:0.980000
010.31 loss:1.346e-01, accuracy:0.950000
010.32 loss:6.542e-02, accuracy:1.000000
010.33 loss:1.734e-01, accuracy:0.960000
010.34 loss:1.470e-01, accuracy:0.970000
010.35 loss:2.022e-01, accuracy:0.960000
010.36 loss:9.329e-02, accuracy:0.970000
010.37 loss:8.539e-02, accuracy:0.980000
010.38 loss:9.641e-02, accuracy:0.980000
010.39 loss:9.380e-02, accuracy:0.960000
010.40 loss:1.700e-01, accuracy:0.940000
loss:2.360e-01, accuracy:0.935000
```

Figure 1: The result shown in the console

Problem 3-2. K-Nearest Neighbor

In this problem, we will play with K-Nearest Neighbor (KNN) algorithm and try it on real-world data. Implement KNN algorithm (in *knn.m/knn.py*), then answer the following questions.

(a) Try KNN with different K and plot the decision boundary.

Answer: Assume that $K = [1, 10, 100]$. The output figure are shown below:

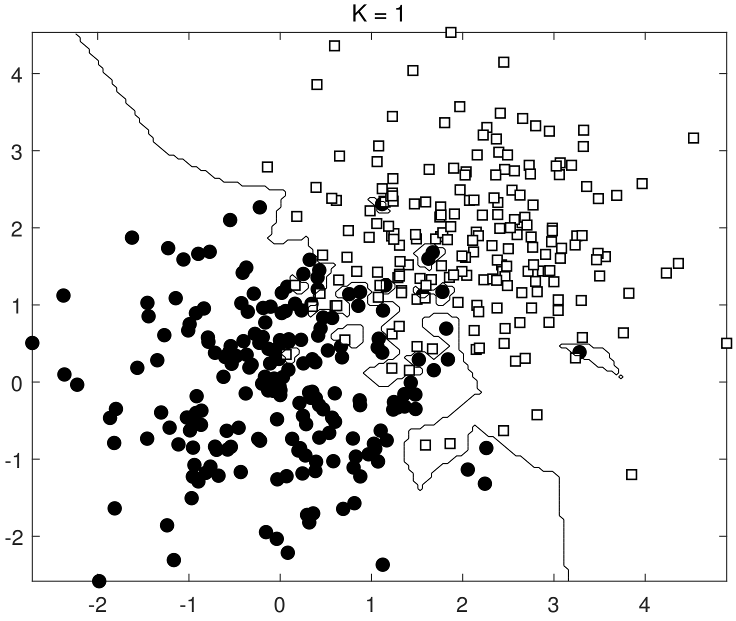


Figure 2: Figure when $K = 1$

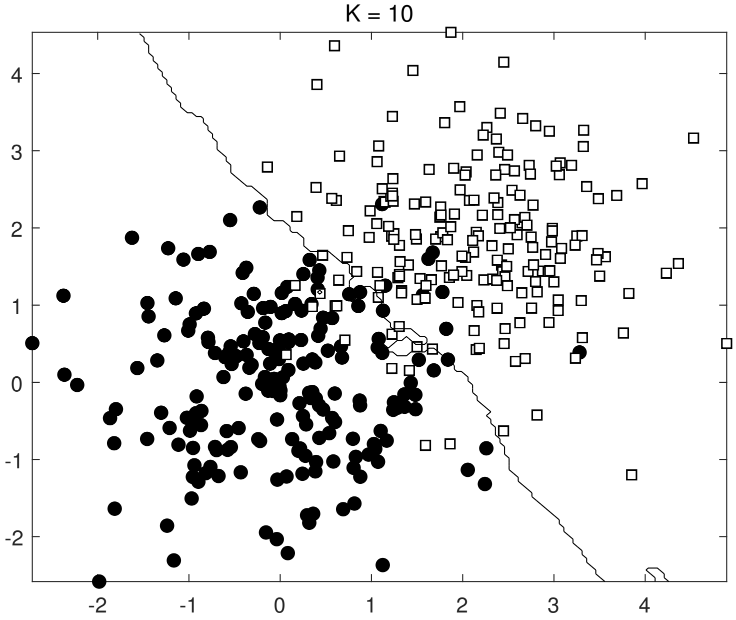


Figure 3: Figure when $K = 10$

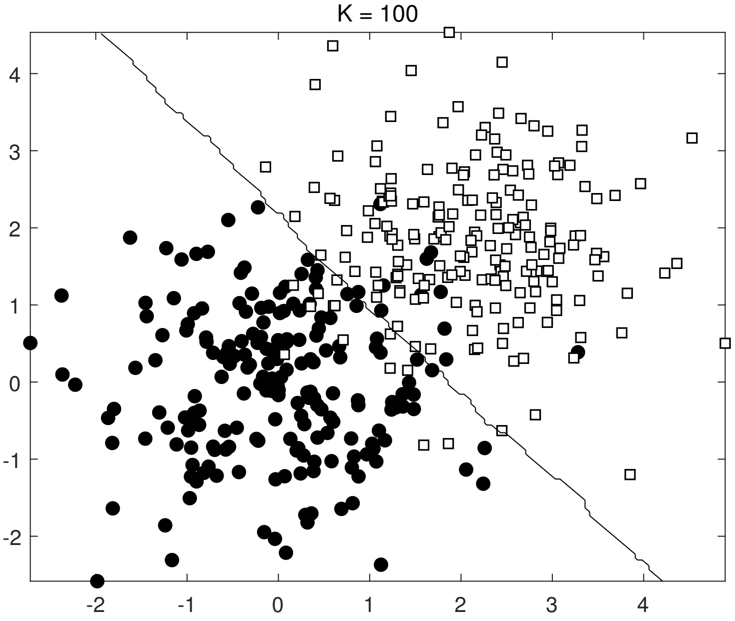


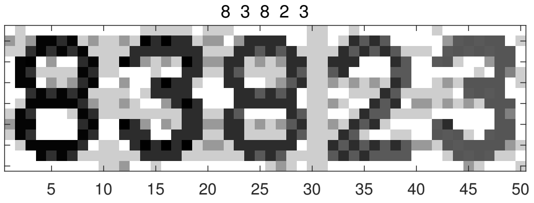
Figure 4: Figure when $K = 100$

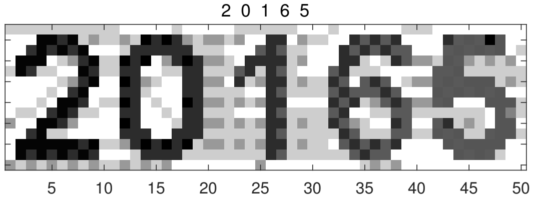
(b) We have seen the effects of different choices of K. How can you choose a proper K when dealing with real-world data ?

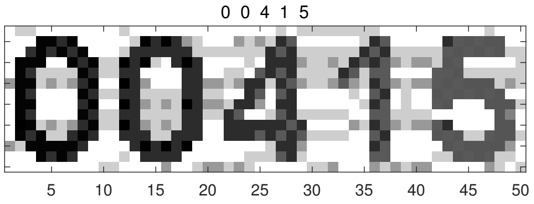
Answer: K ought to based on the No. of classed as well as train data. Assume that the No. of train data is $Count(train)$, and the No. of classed is $Count(class)$, then:$K \leq \left\lfloor \frac{Count(train)}{Count(class)} \right\rfloor$. The smaller the difference between K and the right-hand-side expression, the more linear the decision boundry is.

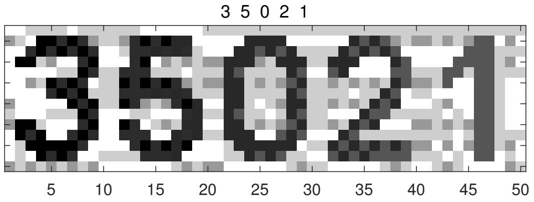
(c) Finish *hack.m*/*hack.py* to recognize the CAPTCHA image using KNN algorithm.

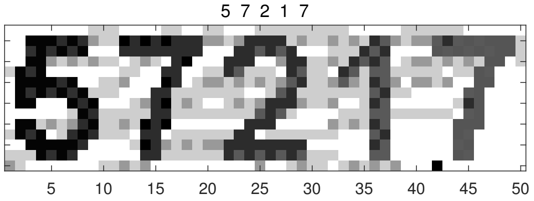
Answer: I used the python crawler to crawl 20 training sets and manually filter and mark each training set. Then I wrote a python script $fetch\_code.py$ that was designed to crawl or update the test set. The test set is updated each time the script is executed. Since the verification codes of the target websites are not always different, I set the test set size to 10, and I will select five of them to display.


8 3 8 2 3
2
4
6
8
10
12
14
5
10
15
20
25
30
35
40
45
50


2 0 1 6 5
2
4
6
8
10
12
14
5
10
15
20
25
30
35
40
45
50


0 0 4 1 5
2
4
6
8
10
12
14
5
10
15
20
25
30
35
40
45
50





Problem 3-3. Decision Tree and ID3

Consider the scholarship evaluation problem: selecting scholarship recipients based on gender and GPA. Given the following training data:

Answer:

$$Entropy(S) = -\sum_{i=1}^{c} P_i \log P_i = 0.991$$
$$Gain(S, GPA) = 0.4266$$
$$Gain(S, Gender) = 0.0112$$

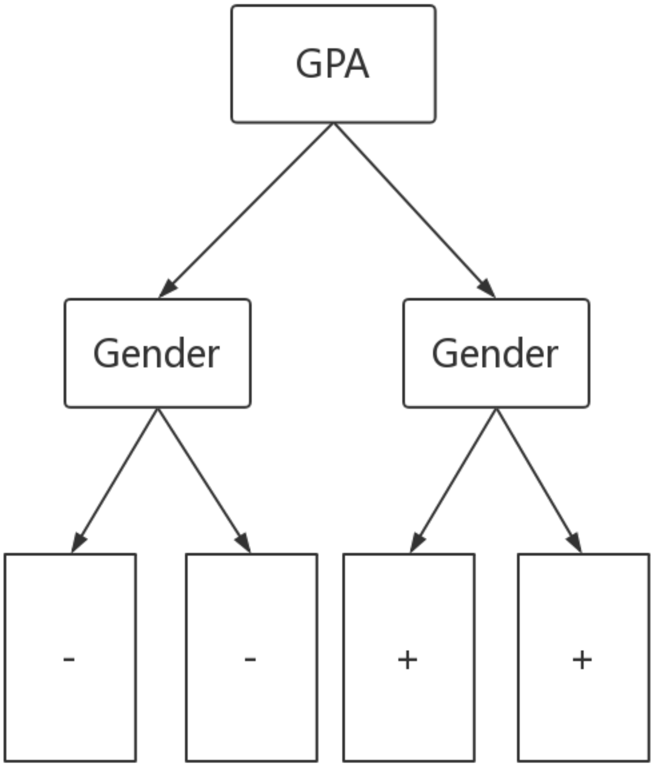


Figure 5: Decision Tree

Problem 3-4. K-Means Clustering

Finally, we will run our first unsupervised algorithm – k-means clustering.

(a) Visualize the process of k-means algorithm for the two trials.

Answer:

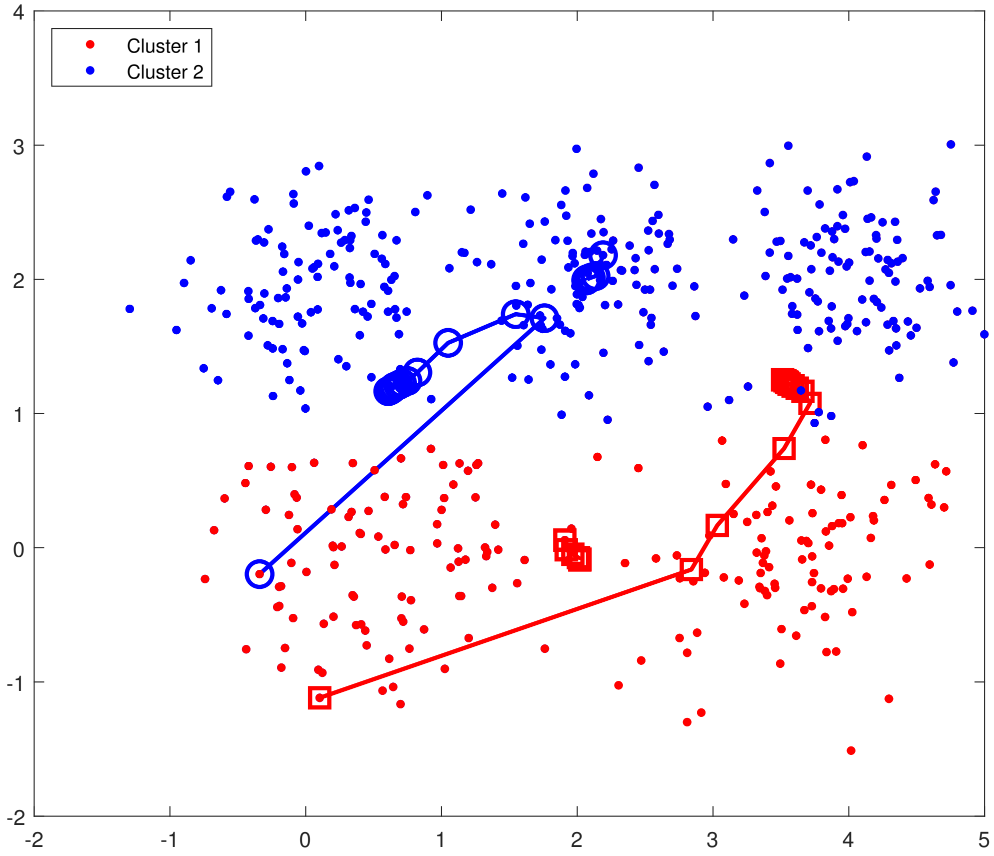


Figure 6: Two trails with the largest and the smallest SD.

(b) How can we get a stable result using k-means?

Answer: It will be better to run k-means for serval times and get average or maximum.

(c) Visualize the centroids.

Answer:

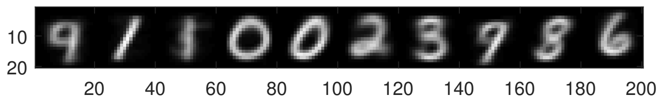


Figure 7: $K = 10$

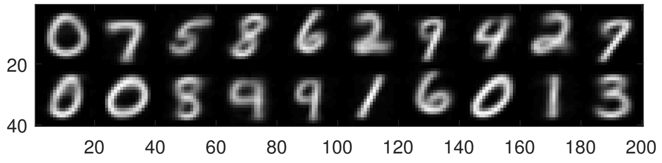


Figure 8: $K = 20$

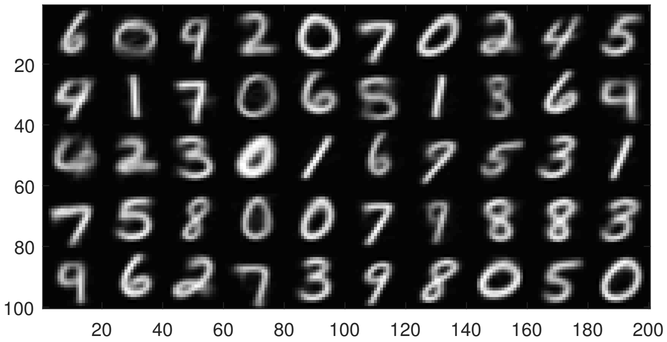


Figure 9: $K = 50$

(d) Vector quantization.

Answer:

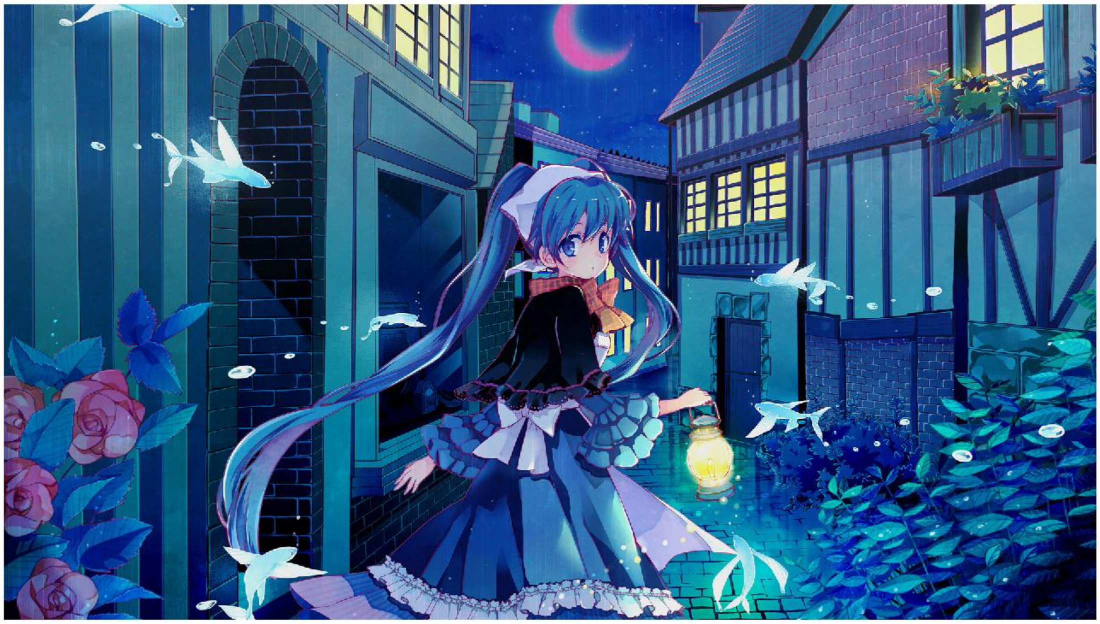


Figure 10: Original image

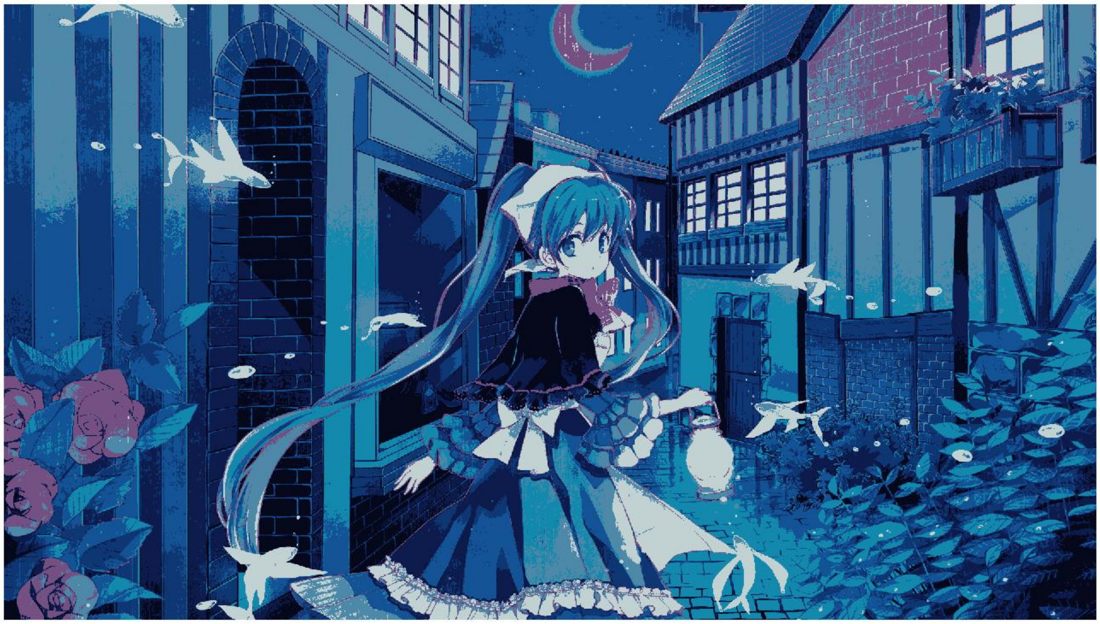


Figure 11: $K = 8$

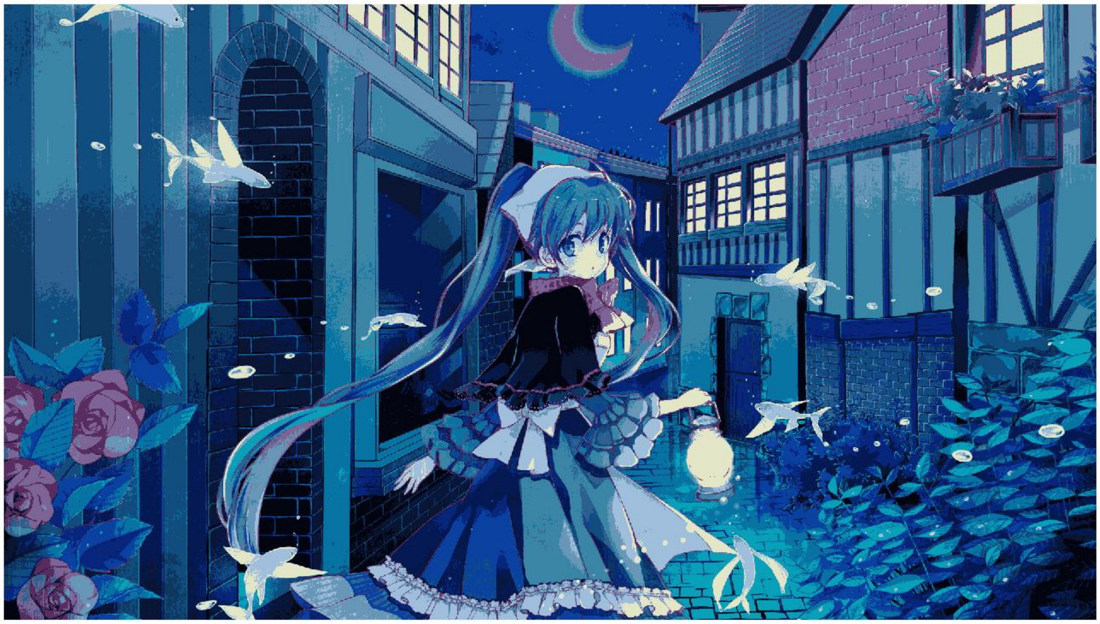


Figure 12: $K = 16$

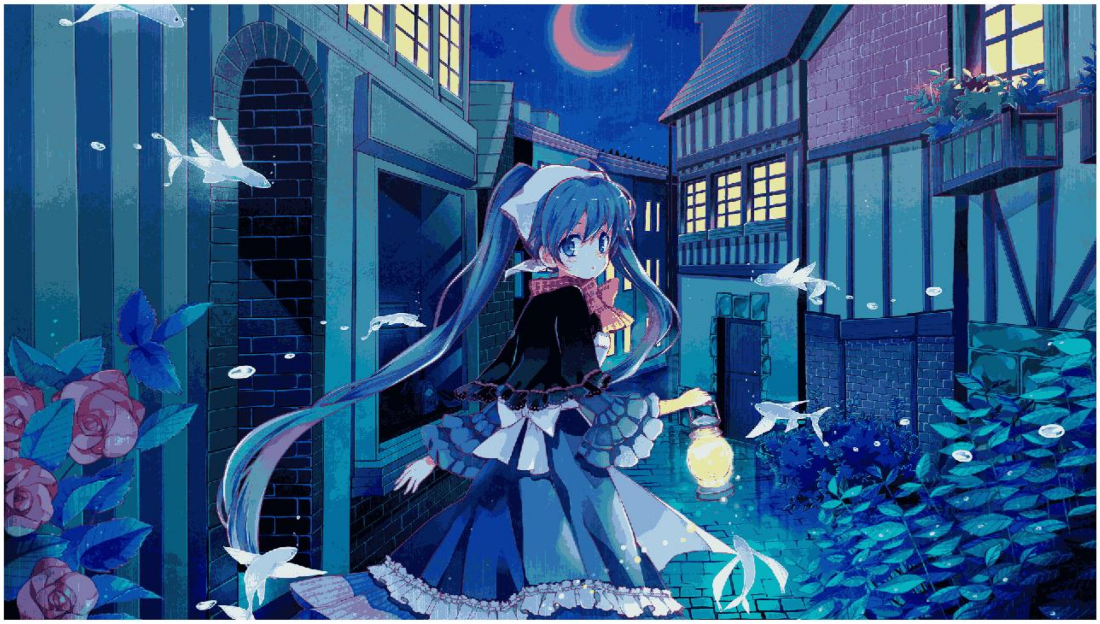


Figure 13: $K = 32$

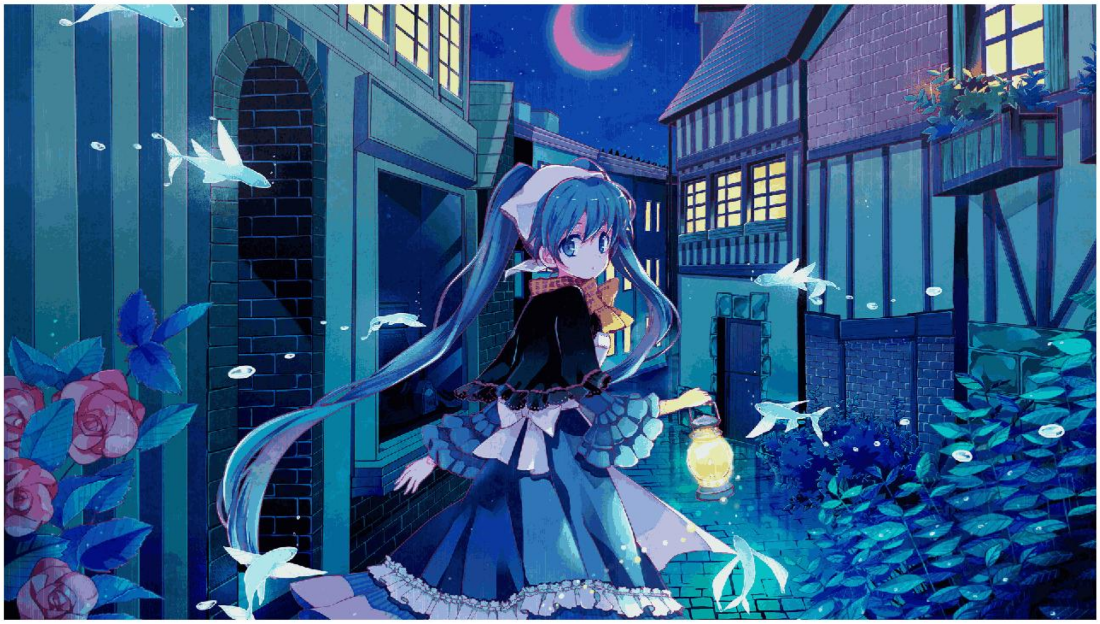


Figure 14: $K = 64$

If we use RGB, then the size of original image is $Np * 24$, where Np is the number of pixels in the original images; If we use kmeans, the size of the image is $Np * log_2K$; So the compression ratio is $\frac{Np*log_2K}{24Np} = \frac{log_2K}{24} = 25\%$