EECE 5644 Assignment 4 Jingcheng Wang

Repository: https://github.com/tomwang777/2025-Fall-EECE-5644-Machine-Learning/tree/main/Assignment%204

Question 1

For this question, I first need to generate data using the metrics given by the document, then I need to run SVM and MLP model separately to utilize K-fold cross validation (uses fitcsvm with Gaussian kernel) and single hidden layer (uses patternnet with tansig activation) to do the classification. Finally I show the result of the classification from booth models by showing various plots through visualization.

Derivation of the class conditional density p(x|r)

Given a certain angle, the Gaussian density is:

$$p(\mathbf{x} \mid r, \theta) = \frac{1}{2\pi\sigma^2} \exp\Big(- \frac{\|\mathbf{x} - r\mathbf{u}(\theta)\|^2}{2\sigma^2} \Big), \quad \mathbf{u}(\theta) = (\cos\theta, \sin\theta)^T$$

Use identity:

$$\int_0^{2\pi} e^{a\cos\theta}\, d\theta = 2\pi I_0(a)$$

We get:

$$p(\mathbf{x} \mid r) = \frac{1}{2\pi\sigma^2} \exp\Big(- \frac{\rho^2 + r^2}{2\sigma^2} \Big)\, I_0\Big(\frac{r\rho}{\sigma^2} \Big)$$

Bayesian Judgment and Decision Boundary

Assume prior knowledge is equal, P(l=+1)=P(l=−1)=1/2, then the decision boundary is:

$$\exp\Big(- \frac{r_{+1}^2 - r_{-1}^2}{2\sigma^2} \Big) \frac{I_0\left(\frac{r_{+1}\rho^*}{\sigma^2} \right)}{I_0\left(\frac{r_{-1}\rho^*}{\sigma^2} \right)} = 1$$

Use $r_{-1} = 2$, $r_{+1} = 4$, $\sigma = 1$ we can calculate that $\rho^* \approx 3.36$.

The expression for Min-P(error)

P(error)

$$P_e = \tfrac{1}{2} \int_{\text{decide } +1} p(\mathbf{x} \mid r_{-1})\, d\mathbf{x} + \tfrac{1}{2} \int_{\text{decide } -1} p(\mathbf{x} \mid r_{+1})\, d\mathbf{x}.$$

Min-P(error)

$$P_e = \frac{1}{2}\left[\int_{\rho^*}^{\infty} \frac{\rho}{\sigma^2} e^{-\frac{\rho^2+r_{-1}^2}{2\sigma^2}} I_0\left(\frac{r_{-1}\rho}{\sigma^2}\right) d\rho + \int_0^{\rho^*} \frac{\rho}{\sigma^2} e^{-\frac{\rho^2+r_{+1}^2}{2\sigma^2}} I_0\left(\frac{r_{+1}\rho}{\sigma^2}\right) d\rho\right]$$

Results

Data Generation
Class -1: r = 2.0, Class +1: r = 4.0, sigma = 1.0
Training samples: 1000, Test samples: 10000

SVM Training
K-fold cross-validation: 10 folds

Best SVM Parameters (via CV):
  Kernel Width: 5.00
  Box Constraint: 0.10
  CV Error: 0.1900 (19.00%)

Training final SVM with optimized hyperparameters...
SVM Test Error: 0.1683 (16.83%)

MLP Training
Testing 10 perceptron configurations...
  Tested 3/10 configurations...
  Tested 6/10 configurations...
  Tested 9/10 configurations...

Best MLP Parameters (via CV):
  Number of Perceptrons: 3
  CV Error: 0.1940 (19.40%)

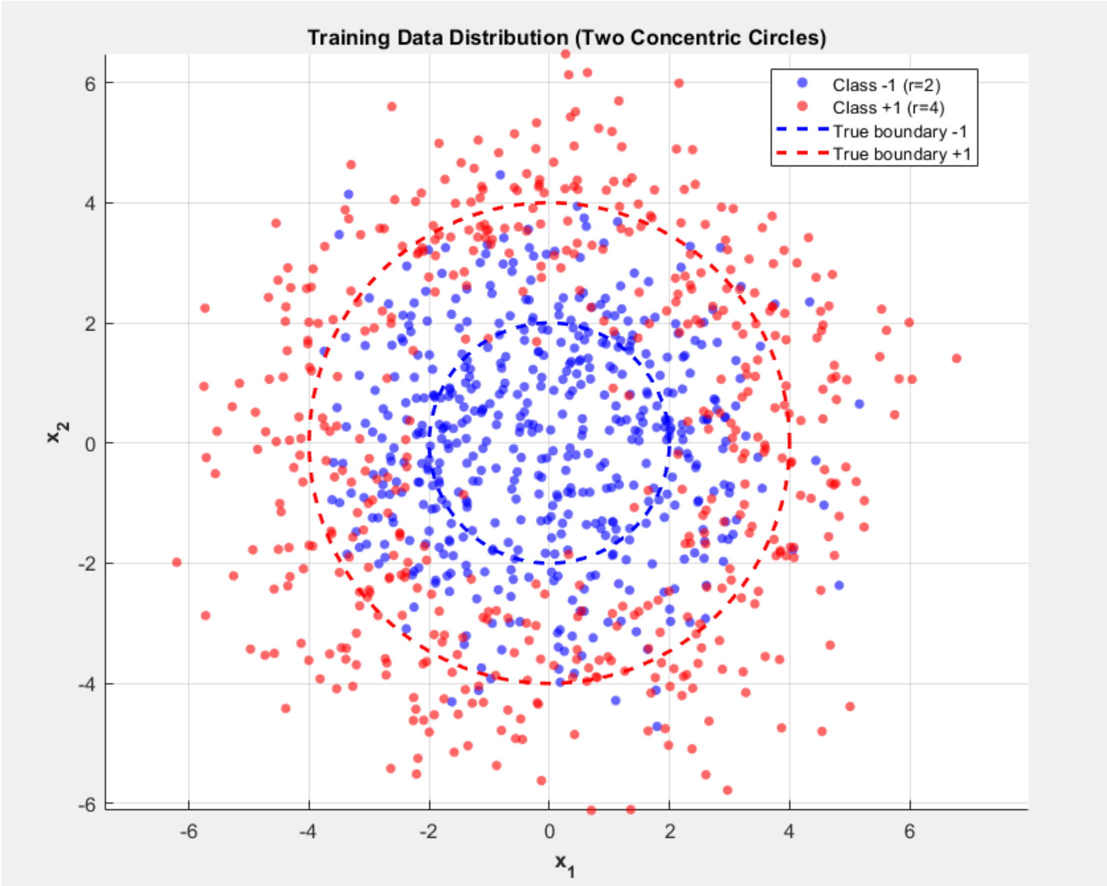
Training final MLP with optimized hyperparameters...
MLP Test Error: 0.1710 (17.10%)

Results

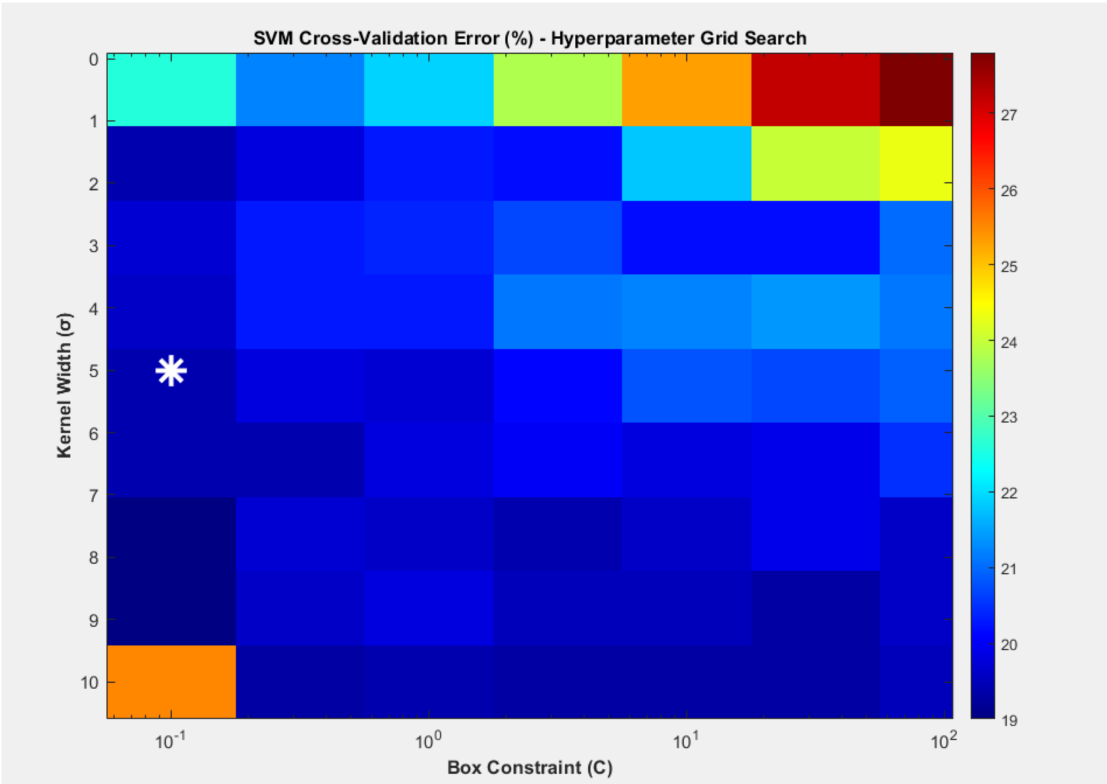
Classifier	Best Hyperparameters	Test Error (%)
SVM (Gaussian)	σ=5.00, C=0.10	0.1683 (16.83%)
MLP (1 hidden layer)	P=3	0.1710 (17.10%)

Data Visualization

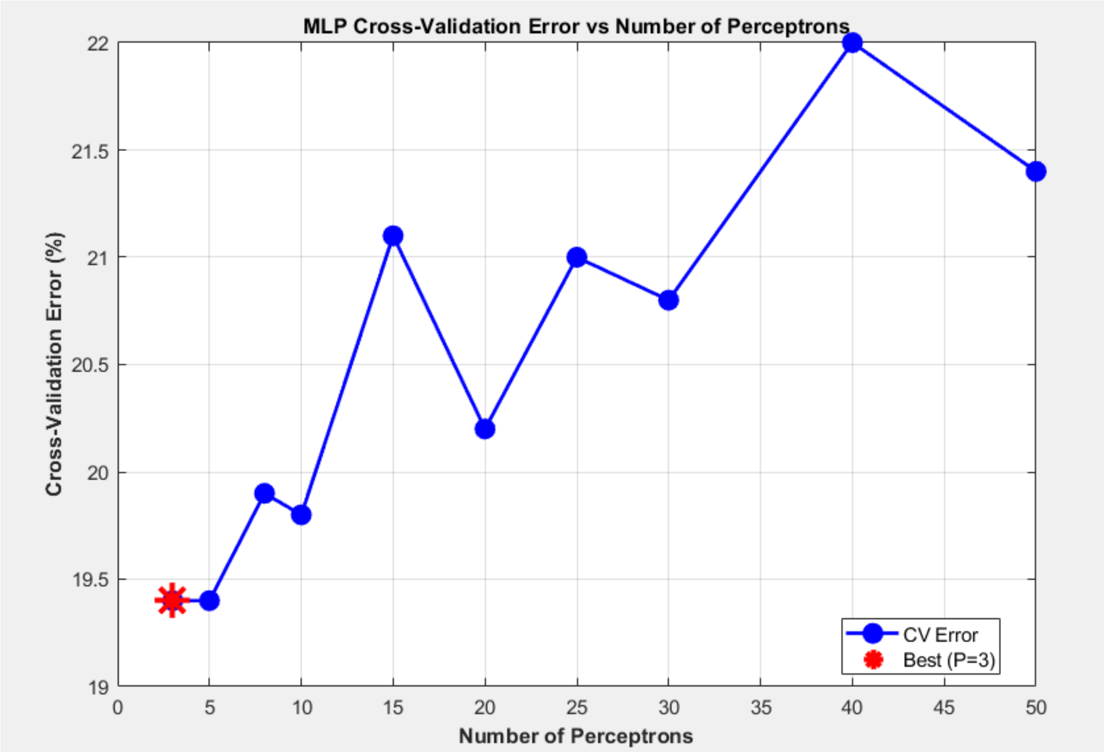
Plot of training data with true circular boundaries



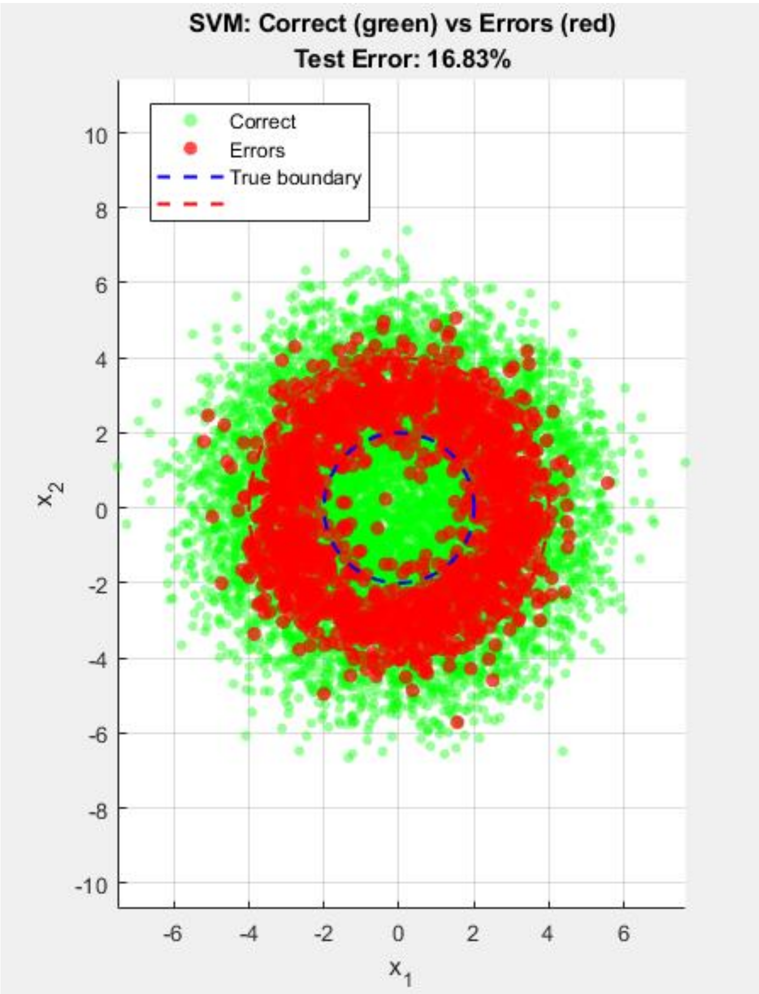
Plot of SVM CV heatmap



Plot of MLP CV Error curve



Plot of error distribution of SVM



As RBF (Gaussian) can approximate circles and MLP could only estimate the circular boundary with sufficient perceptrons, SVM model should have a better classification result, and the loss result shows the alignment as SVM got 0.1683 test loss and MLP got 0.1710. However, as we can see from the plots, the two models get a similar performance overall, both giving relatively correct classification.

Question 2

In this problem, for the selected image in the given dataset, a five-dimensional feature vector is generated for each pixel in the image and normalized. A Gaussian mixture model (GMM) is then fitted, and maximum likelihood estimation is used for K-fold verification during the fitting process. The model order is selected, and a label is assigned to each pixel to complete the image segmentation.

Likelihood function and objective

Joint log-likelihood function of GMM (maximization goal)

$$\ln p(\mathcal{D}|\boldsymbol{\Theta}) = \sum_{n=1}^{N} \ln \left(\sum_{k=1}^{K'} \pi_k g(\mathbf{z}_n|\mu_k, \boldsymbol{\Sigma}_k) \right)$$

GMM Algorithm (Using EM)

EM Algorithm: E-step (Estimated Step)

The posterior probability that data point zn comes from the k-th Gaussian component

$$\gamma(z_n, k) = P(k|\mathbf{z}_n, \boldsymbol{\Theta}^{(\text{old})}) = \frac{\pi_k^{(\text{old})} g(\mathbf{z}_n|\mu_k^{(\text{old})}, \boldsymbol{\Sigma}_k^{(\text{old})})}{\sum_{j=1}^{K'} \pi_j^{(\text{old})} g(\mathbf{z}_n|\mu_j^{(\text{old})}, \boldsymbol{\Sigma}_j^{(\text{old})})}$$

EM Algorithm: M-step

Update component weights

$$\pi_k^{(\text{new})} = \frac{N_k}{N} = \frac{1}{N} \sum_{n=1}^{N} \gamma(z_n, k)$$

Update mean value

$$\mu_k^{(\text{new})} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_n, k) \mathbf{z}_n$$

Update covariance matrix

$$\boldsymbol{\Sigma}_k^{(\text{new})} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_n, k) (\mathbf{z}_n - \mu_k^{(\text{new})})(\mathbf{z}_n - \mu_k^{(\text{new})})^T$$

Results

Image loaded successfully.
Image size: 481 x 321 x 3
Image downsampled to: 200 x 134 (scale factor: 0.42)

Creating 5-dimensional feature vectors
Feature matrix created: 26800 pixels x 5 features
Normalizing features to [0, 1]...
Features normalized. All feature vectors fit in 5D unit hypercube.
K-fold cross-validation: 5 folds
Candidate model orders: 2 to 8 components
GMM fitting replicates: 3
Running cross-validation...
Testing M = 2 components...
CV log-likelihood: 11691.60
  Testing M = 3 components...
 CV log-likelihood: 12428.22
  Testing M = 4 components...
 CV log-likelihood: 12571.51
  Testing M = 5 components...
 CV log-likelihood: 12595.74
  Testing M = 6 components...
 CV log-likelihood: 12607.26
  Testing M = 7 components...
 CV log-likelihood: 12608.67
  Testing M = 8 components...
 CV log-likelihood: 12586.50

Best model order selected: M = 7 components
Best CV log-likelihood: 12608.67

Training GMM with M = 7 components on full dataset
Final GMM training complete.

GMM Component Weights:
  Component 1: 0.0017
  Component 2: 0.3207
  Component 3: 0.0754
  Component 4: 0.2411
  Component 5: 0.2571
  Component 6: 0.0017
  Component 7: 0.1023

Image Segmentation
Segmentation complete.
Segments identified: 5

Results
Image size: 200 x 134
Total pixels: 26800
Number of segments: 7

Segment sizes:
  Segment 1: 0 pixels (0.00%)
  Segment 2: 8488 pixels (31.67%)
  Segment 3: 2021 pixels (7.54%)
  Segment 4: 8589 pixels (32.05%)
  Segment 5: 7663 pixels (28.59%)
  Segment 6: 0 pixels (0.00%)
  Segment 7: 39 pixels (0.15%)

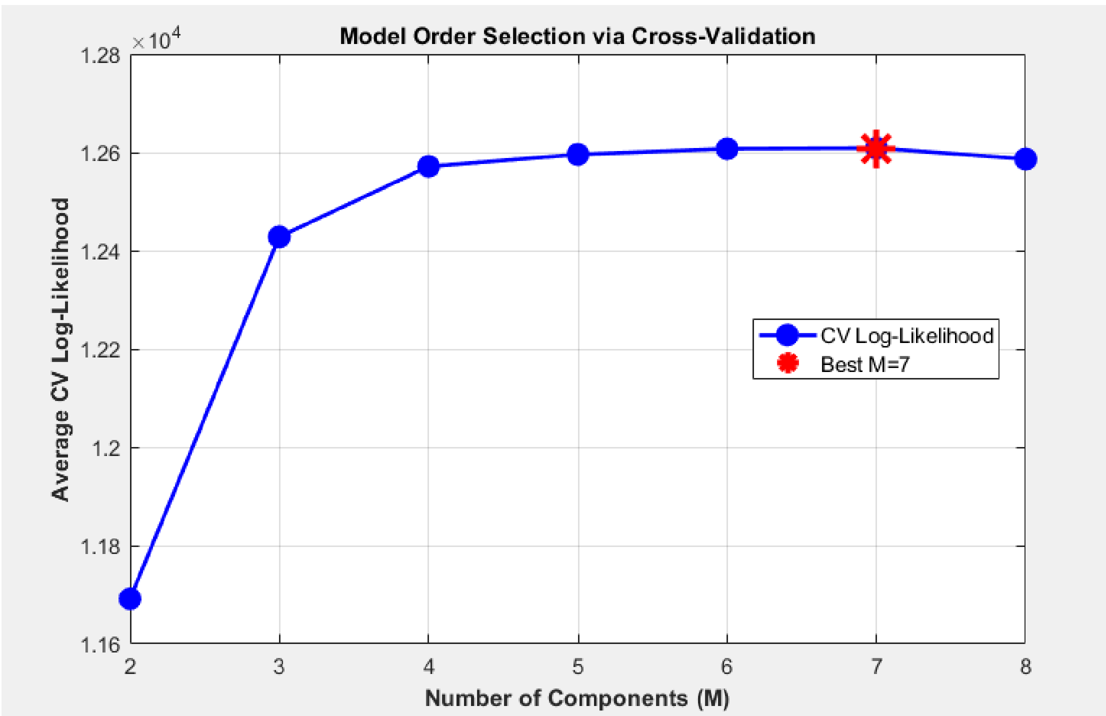
GMM Component Characteristics:
  Component 1:
    Weight: 0.0017
    Mean RGB (normalized): [0.349, 0.321, 0.250]
  Component 2:
    Weight: 0.3207
    Mean RGB (normalized): [0.219, 0.157, 0.111]
  Component 3:
    Weight: 0.0754
    Mean RGB (normalized): [0.508, 0.433, 0.436]
  Component 4:
    Weight: 0.2411
    Mean RGB (normalized): [0.345, 0.318, 0.246]
  Component 5:
    Weight: 0.2571
    Mean RGB (normalized): [0.200, 0.155, 0.129]
  Component 6:
    Weight: 0.0017
    Mean RGB (normalized): [0.348, 0.320, 0.249]
  Component 7:
    Weight: 0.1023
    Mean RGB (normalized): [0.331, 0.313, 0.237]

Data Visualization

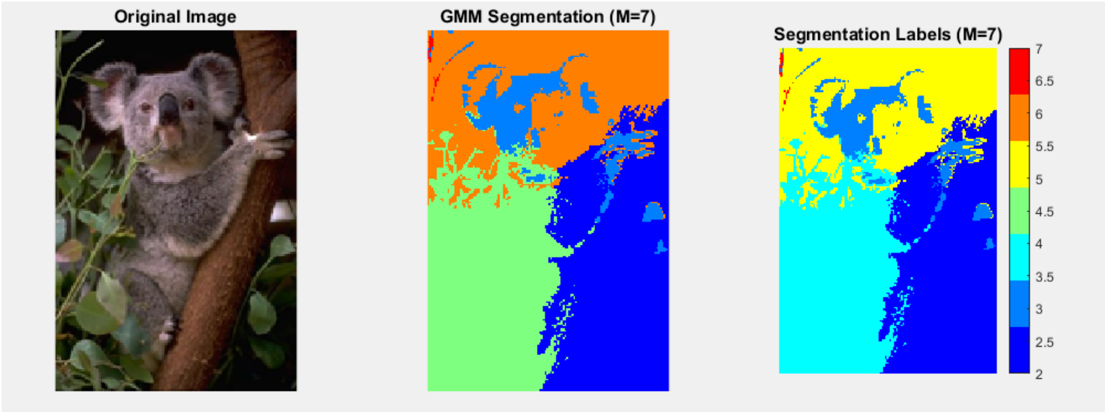
Original Image

Segmented Image

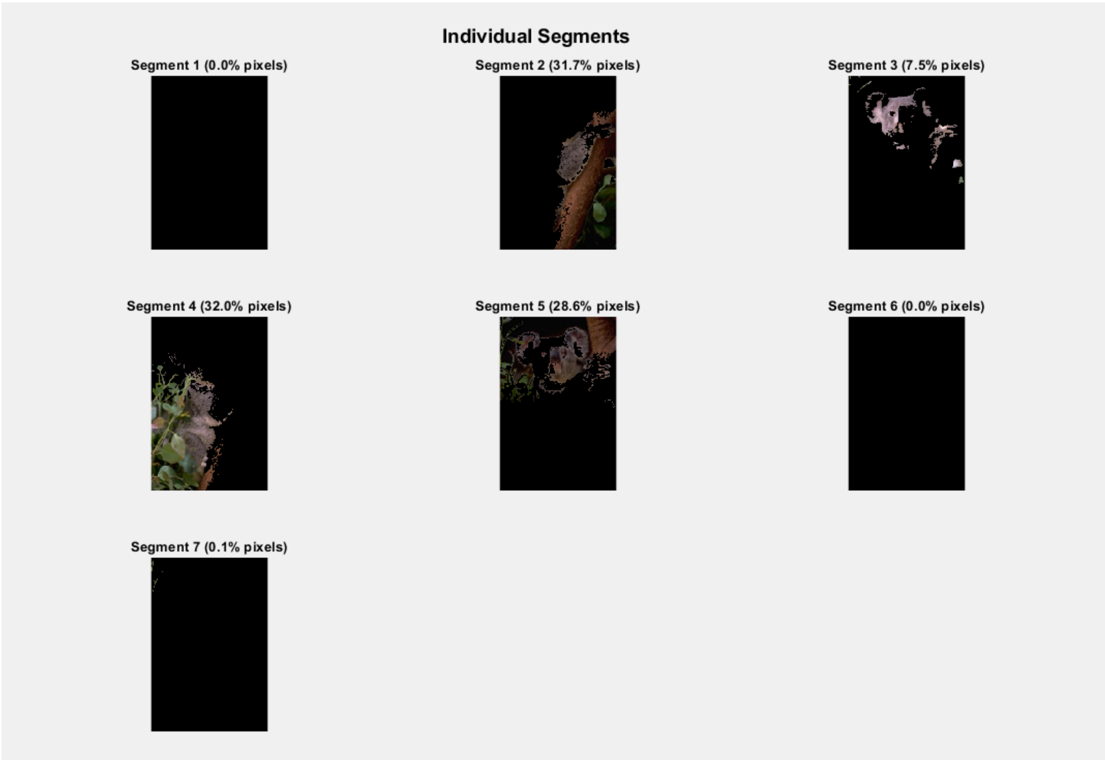
Plot of Model Order Selection Curve



Plot of RGB Feature Space



Plot of Individual Segments



By visualizing the results and comparing the original and segmented images, we can see that the GMM model initially selected the components with the highest log-likelihood of 7 for training in its successive training. After obtaining the weights of each component, the image was then segmented. This corresponds to the different RGB mean values, resulting in a good comparison effect.

Citation

1. Course recording
2. Course notes
3. Course codes provided on Canvas
4. Discussion with classmates
5. Generative AI models
6. Training tools from Matlab source