Registration Number Recognizer

BIAI PSI sem. 6

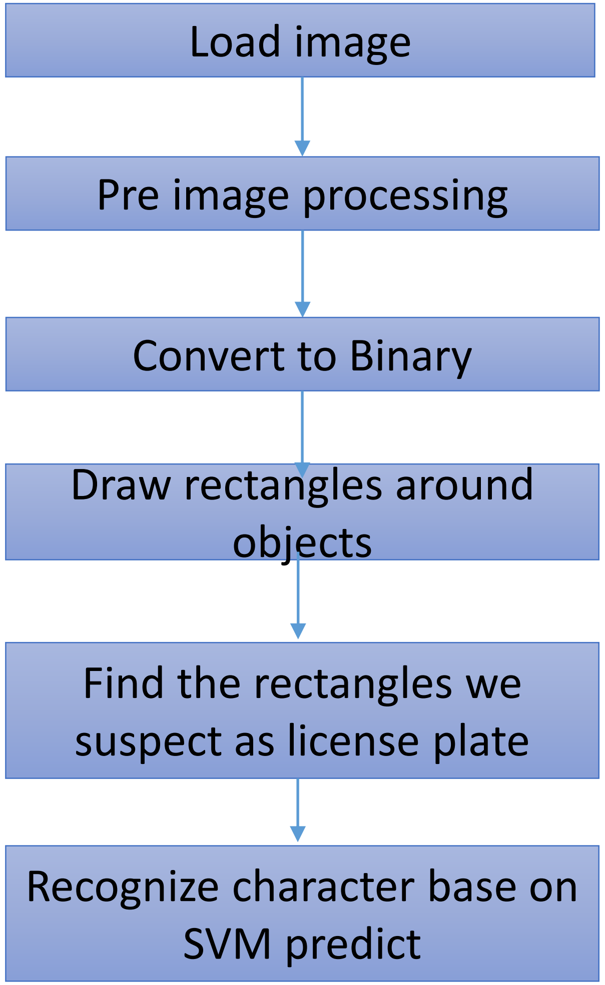
Jakub Czachor

Project assumptions

- A program that searches for a car licence plate on a photo and then recognises its registration number (at the end, the program generates a textbox from where you can copy the registration number as a string)
- Machine learning – SVM – Support Vector Machine (OpenCV)
- C++ program (GUI, machine learning, algorithm, any calculations)
- GUI – Windows Forms
- Photo editing – OpenCV library

Program architecture

Detect License Plate



Load image

Pre image processing

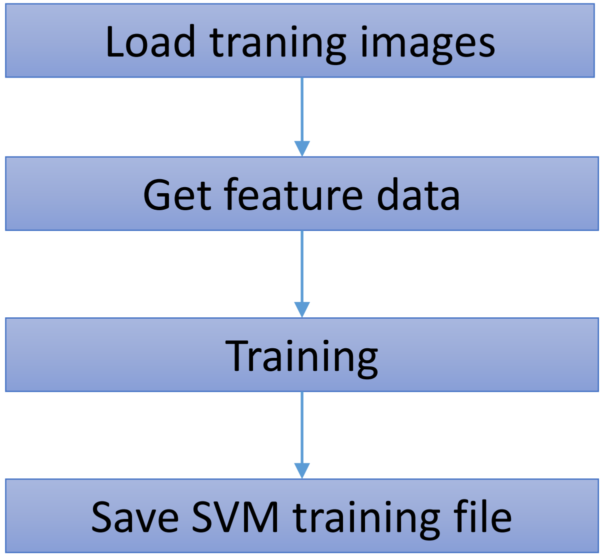
Convert to Binary

Draw rectangles around objects

Find the rectangles we suspect as license plate

Recognize character base on SVM predict

Character training



Load traning images

Get feature data

Training

Save SVM training file

Photo editing – pre-image process

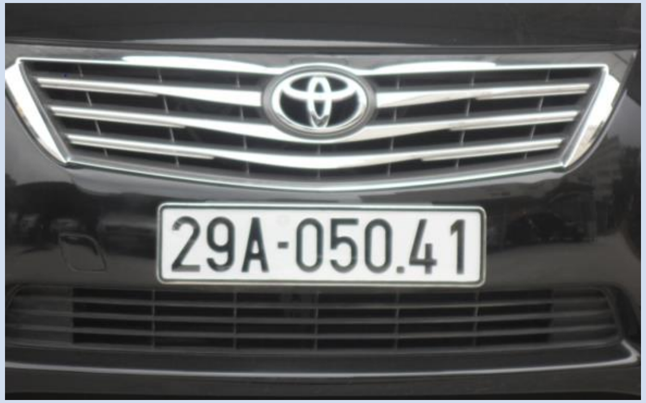
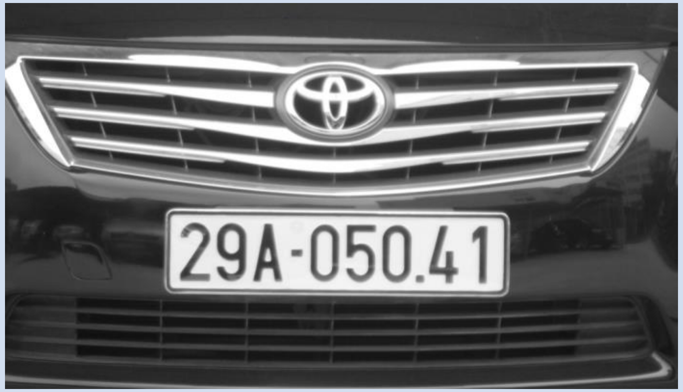
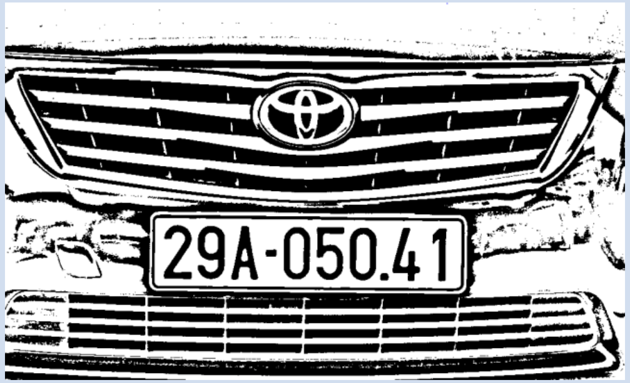
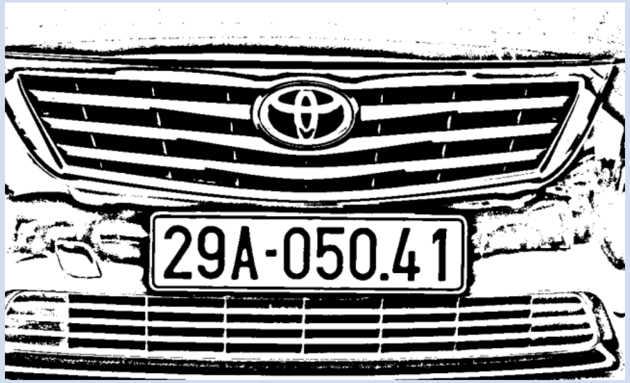
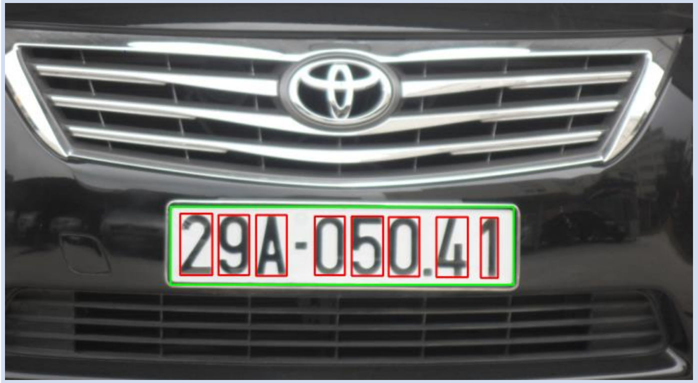
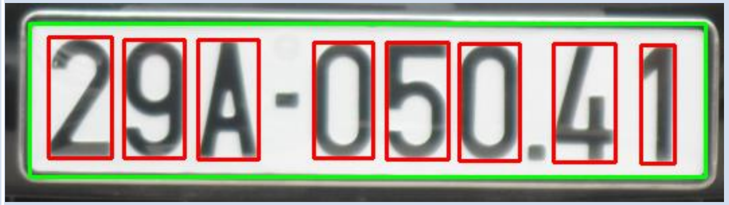
image	Convert to gray	Convert to binary
		

Photo editing – character recognition

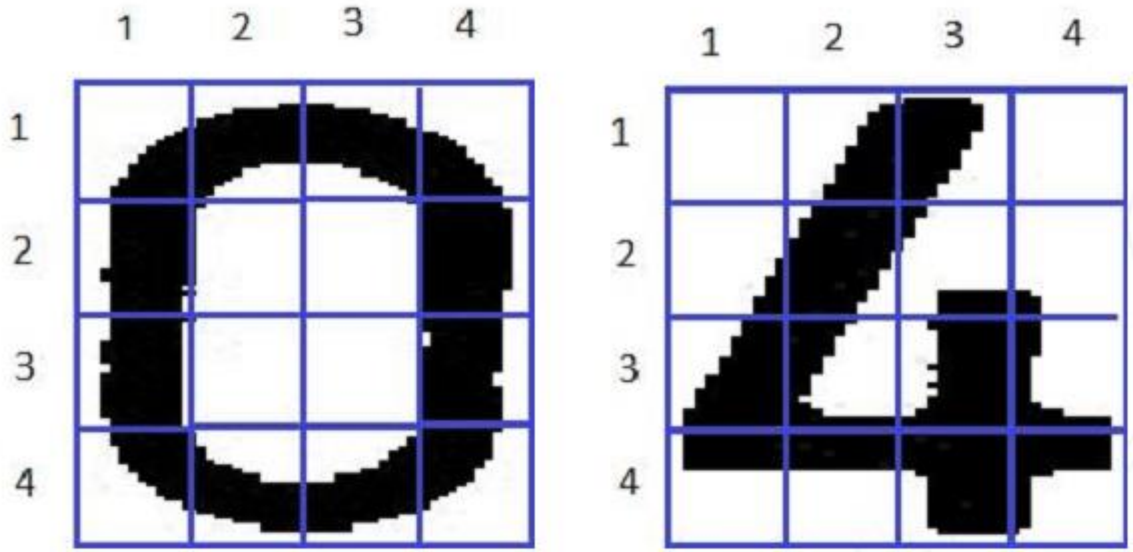
Binary image	Find contours	Detect the plate
		

SVM Training/Feature calculation

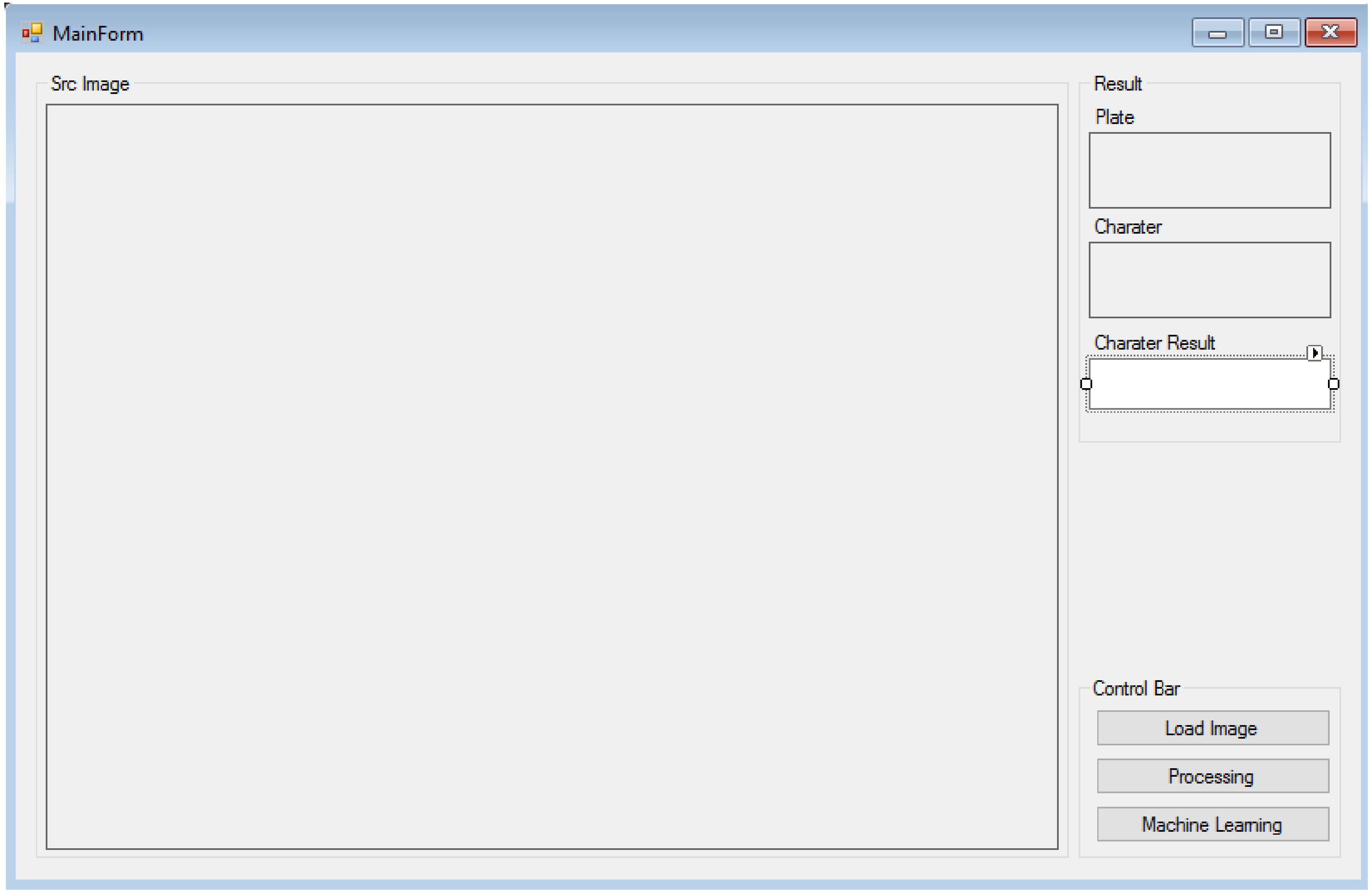
Example we have 2 character image 0 and 4

We need resize 2 image to the same size and divide as 16 small area as picture

As you can see we can find difference between 2 image on cell(1,1)(1,4)(2,2)... on those cell total back pixel are difference.



GUI


MainForm
Src Image
Result
Plate
Charater
Charater Result
Control Bar
Load Image
Processing
Machine Learning

Sample code – load image

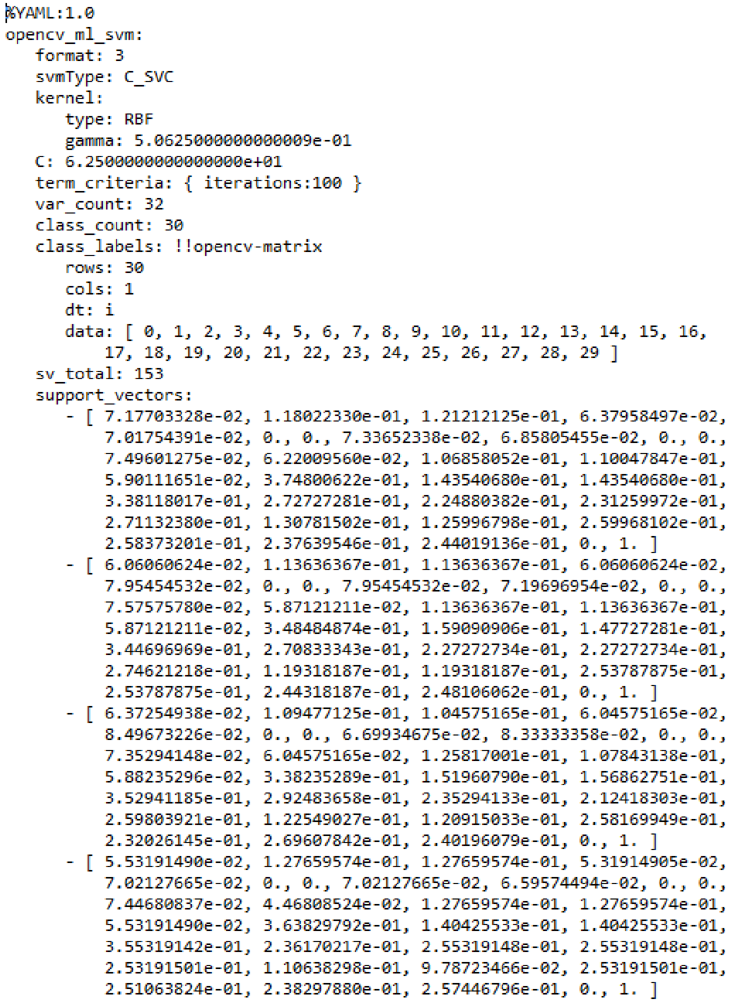
```
// Load image by dialog
OpenFileDialog^ dgOpen = gcnew OpenFileDialog();
dgOpen->Filter = "Image(*.bmp; *.jpg)|*.bmp;*.jpg|All files (*.*)|*.*||";

if (dgOpen->ShowDialog() == System::Windows::Forms::DialogResult::Cancel) {
       return;
}

// Assign image to the picture box & srcImage
Bitmap^ bmpSrc = gcnew Bitmap(dgOpen->FileName);
ptbSrc->Image = bmpSrc;
ptbSrc->Refresh();
srcImg = imread(stringTochar.ConvertString2Char(dgOpen->FileName));
```

Sample code – SVM Training

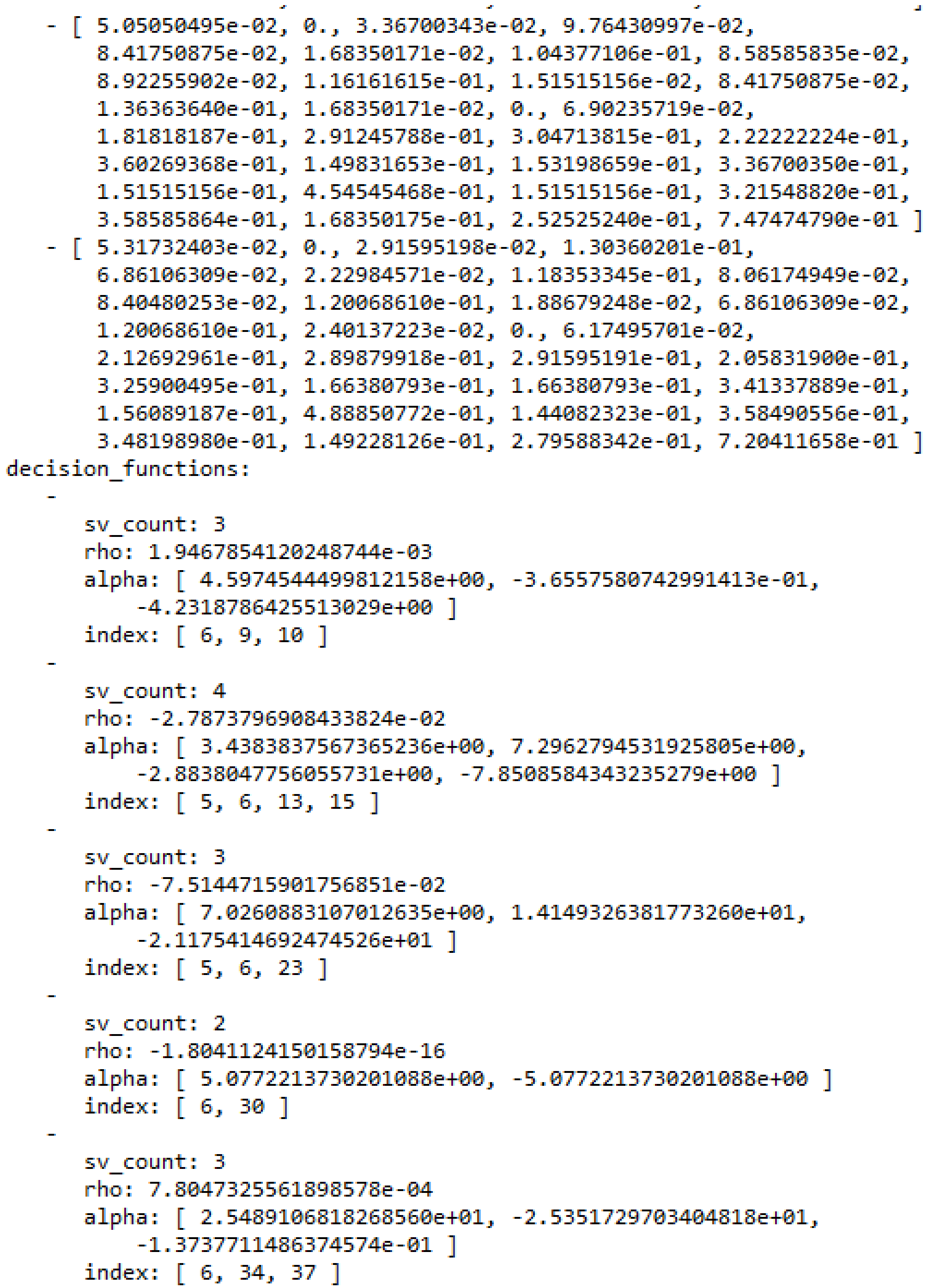
```
//Train SVM OpenCV 3.1 definition
Ptr<SVM> svm = SVM::create();
svm->setType(SVM::C_SVC);
svm->setKernel(SVM::RBF);
svm->setGamma(0.5);
svm->setC(16);
svm->setTermCriteria(TermCriteria(TermCriteria::MAX_ITER, 100, 1e-6));
vector<string> folders = list_folder(trainImgpath);
Mat src;
Mat data = Mat( number_of_sample * number_of_class, number_of_feature, CV_32FC1);
Mat label = Mat(number_of_sample * number_of_class, 1, CV_32SC1);
```

Sample result file

After SVM learning



```
%YAML:1.0
opencv_ml_svm:
   format: 3
   svmType: C_SVC
   kernel:
      type: RBF
      gamma: 5.0625000000000009e-01
   C: 6.2500000000000000e+01
   term_criteria: { iterations:100 }
   var_count: 32
   class_count: 30
   class_labels: !!opencv-matrix
      rows: 30
      cols: 1
      dt: i
      data: [ 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16,
          17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29 ]
   sv_total: 153
   support_vectors:
      - [ 7.17703328e-02, 1.18022330e-01, 1.21212125e-01, 6.37958497e-02,
          7.01754391e-02, 0., 0., 7.33652338e-02, 6.85805455e-02, 0., 0.,
          7.49601275e-02, 6.22009560e-02, 1.06858052e-01, 1.10047847e-01,
          5.90111651e-02, 3.74800622e-01, 1.43540680e-01, 1.43540680e-01,
          3.38118017e-01, 2.72727281e-01, 2.24880382e-01, 2.31259972e-01,
          2.71132380e-01, 1.30781502e-01, 1.25996798e-01, 2.59968102e-01,
          2.58373201e-01, 2.37639546e-01, 2.44019136e-01, 0., 1. ]
      - [ 6.06060624e-02, 1.13636367e-01, 1.13636367e-01, 6.06060624e-02,
          7.95454532e-02, 0., 0., 7.95454532e-02, 7.19696954e-02, 0., 0.,
          7.57575780e-02, 5.87121211e-02, 1.13636367e-01, 1.13636367e-01,
          5.87121211e-02, 3.48484874e-01, 1.59090906e-01, 1.47727281e-01,
          3.44696969e-01, 2.70833343e-01, 2.27272734e-01, 2.27272734e-01,
          2.74621218e-01, 1.19318187e-01, 1.19318187e-01, 2.53787875e-01,
          2.53787875e-01, 2.44318187e-01, 2.48106062e-01, 0., 1. ]
      - [ 6.37254938e-02, 1.09477125e-01, 1.04575165e-01, 6.04575165e-02,
          8.49673226e-02, 0., 0., 6.69934675e-02, 8.33333358e-02, 0., 0.,
          7.35294148e-02, 6.04575165e-02, 1.25817001e-01, 1.07843138e-01,
          5.88235296e-02, 3.38235289e-01, 1.51960790e-01, 1.56862751e-01,
          3.52941185e-01, 2.92483658e-01, 2.35294133e-01, 2.12418303e-01,
          2.59803921e-01, 1.22549027e-01, 1.20915033e-01, 2.58169949e-01,
          2.32026145e-01, 2.69607842e-01, 2.40196079e-01, 0., 1. ]
      - [ 5.53191490e-02, 1.27659574e-01, 1.27659574e-01, 5.31914905e-02,
          7.02127665e-02, 0., 0., 7.02127665e-02, 6.59574494e-02, 0., 0.,
          7.44680837e-02, 4.46808524e-02, 1.27659574e-01, 1.27659574e-01,
          5.53191490e-02, 3.63829792e-01, 1.40425533e-01, 1.40425533e-01,
          3.55319142e-01, 2.36170217e-01, 2.55319148e-01, 2.55319148e-01,
          2.53191501e-01, 1.10638298e-01, 9.78723466e-02, 2.53191501e-01,
          2.51063824e-01, 2.38297880e-01, 2.57446796e-01, 0., 1. ]
```



```
      - [ 5.05050495e-02, 0., 3.36700343e-02, 9.76430997e-02,
          8.41750875e-02, 1.68350171e-02, 1.04377106e-01, 8.58585835e-02,
          8.92255902e-02, 1.16161615e-01, 1.51515156e-02, 8.41750875e-02,
          1.36363640e-01, 1.68350171e-02, 0., 6.90235719e-02,
          1.81818187e-01, 2.91245788e-01, 3.04713815e-01, 2.22222224e-01,
          3.60269368e-01, 1.49831653e-01, 1.53198659e-01, 3.36700350e-01,
          1.51515156e-01, 4.54545468e-01, 1.51515156e-01, 3.21548820e-01,
          3.58585864e-01, 1.68350175e-01, 2.52525240e-01, 7.47474790e-01 ]
      - [ 5.31732403e-02, 0., 2.91595198e-02, 1.30360201e-01,
          6.86106309e-02, 2.22984571e-02, 1.18353345e-01, 8.06174949e-02,
          8.40480253e-02, 1.20068610e-01, 1.88679248e-02, 6.86106309e-02,
          1.20068610e-01, 2.40137223e-02, 0., 6.17495701e-02,
          2.12692961e-01, 2.89879918e-01, 2.91595191e-01, 2.05831900e-01,
          3.25900495e-01, 1.66380793e-01, 1.66380793e-01, 3.41337889e-01,
          1.56089187e-01, 4.88850772e-01, 1.44082323e-01, 3.58490556e-01,
          3.48198980e-01, 1.49228126e-01, 2.79588342e-01, 7.20411658e-01 ]
   decision_functions:
      -
         sv_count: 3
         rho: 1.9467854120248744e-03
         alpha: [ 4.5974544499812158e+00, -3.6557580742991413e-01,
             -4.2318786425513029e+00 ]
         index: [ 6, 9, 10 ]
      -
         sv_count: 4
         rho: -2.7873796908433824e-02
         alpha: [ 3.4383837567365236e+00, 7.2962794531925805e+00,
             -2.8838047756055731e+00, -7.8508584343235279e+00 ]
         index: [ 5, 6, 13, 15 ]
      -
         sv_count: 3
         rho: -7.5144715901756851e-02
         alpha: [ 7.0260883107012635e+00, 1.4149326381773260e+01,
             -2.1175414692474526e+01 ]
         index: [ 5, 6, 23 ]
      -
         sv_count: 2
         rho: -1.8041124150158794e-16
         alpha: [ 5.0772213730201088e+00, -5.0772213730201088e+00 ]
         index: [ 6, 30 ]
      -
         sv_count: 3
         rho: 7.8047325561898578e-04
         alpha: [ 2.5489106818268560e+01, -2.5351729703404818e+01,
             -1.3737711486374574e-01 ]
         index: [ 6, 34, 37 ]
```

Thank you for your attention