МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Костромской государственный университет»

(КГУ)

ИАСТ

Кафедра автоматизированных систем и технологий

09.03.02

Направление подготовки/Специальность Информационные системы и технологии

Дисциплина Технологии компьютерного зрения

# Лабораторная №7.

# Простые классификаторы.

Выполнили студенты Копосов Лев Владимирович

Копосов Владимир Владимирович

Группа 22-ИСбо-1б

Проверил \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Подпись преподавателя \_\_\_\_\_\_\_\_\_\_\_\_

Кострома

Вопросы:

1. **В методе опорных векторов (SVM), что называется опорными векторами?**

Опорными векторами в методе опорных векторов (SVM) называются ближайшие к гиперплоскости точки, которые определяют ширину коридора.

Ширина коридора - отступ, который разделяет положительные и отрицательные классифицированные объекты.

1. **Какими способами можно классифицировать объекты по нескольким классам с помощью SVM?**

Классифицировать объекты по нескольким классам с помощью SVM можно следующими способами:

1. «один против всех»

Для каждого класса (всего N) обучаем отдельный классификатор. Тот классификатор, который даст наиболее уверенный положительный результат, побеждает.

1. «один на один»

Для каждой пары классов (всего N\*(N-1)/2 пар) обучаем отдельный классификатор. Класс получает «голоса» от каждого классификатора, в пару классов которого он входит.

1. «решающее дерево»

Последовательно делим пространство на всё более мелкие области. В конечном итоге каждая область будет содержать только один класс.

1. **Для чего применяется повышение размерности при классификации с помощью SVM?**

Повышение размерности при классификации с помощью SVM применяется для устранения проблемы выбора гиперплоскости, то есть позволяет разделять классы, которые не могут быть разделены линейно.

Проблема повышения размерности заключается в быстром росте затрат на вычисление скалярных произведений.

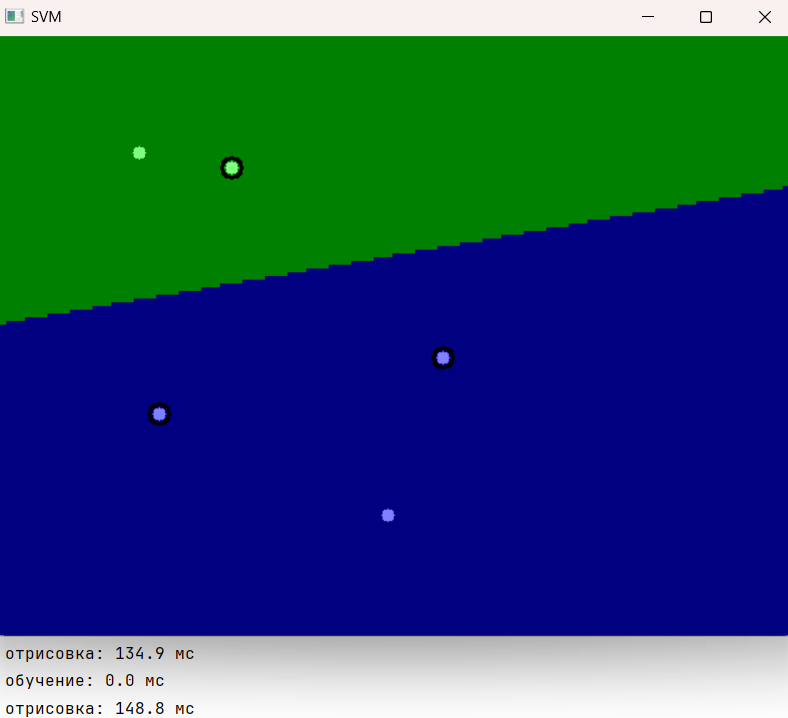
Вывод в консоль.

##### **Задание 1**

Создайте программу, выполняющую следующие действия.

* Программа должна вывести на экран окно размера 640х480.
* Щелчки мыши в окне должны регистрироваться, при этом должны регистрироваться сведения о том, был ли щелчок выполнен левой или правой кнопкой мыши. Нажатие на колёсико мыши должно очищать список.
* По нажатию Enter, если в списке есть хотя бы по две точки каждого типа, следует обучить на нём классификатор (метод **.train()**) по методу опорных векторов (функция **cv2.ml.SVM\_create()**). Используйте тип классификатора **cv2.ml.SVM\_C\_SVC** и ядро **cv2.ml.SVM\_LINEAR**. Критерий останова: **(cv2.TERM\_CRITERIA\_MAX\_ITER+cv2.TERM\_CRITERIA\_EPS, 100, 1.0)**.
* После каждого обучения создайте изображение размером 640х480 и классифицируйте каждый пиксель в нём. Окрасьте пиксели в зелёный или синий цвет в зависимости от результата классификации.
* Также отметьте на изображении поставленные ранее точки. Цвета точек должны соответствовать их классу. Выясните, какие из них являются опорными векторами с помощью метода **.getUncompressedSupportVectors()**, и отметьте их отдельной меткой (если эти сведения доступны). Выведите полученное изображение в окно программы.
* Программа должна выводить время, затраченное на обучение и на построение карты классов.
* По нажатию Esc приложение должно закрыть окно.

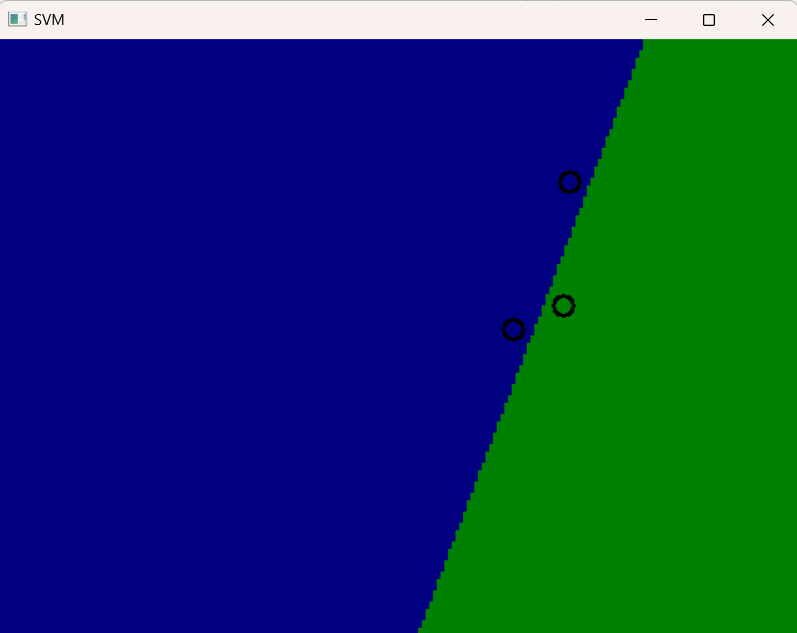
Рассмотрите поведение программы для случая линейно разделимых и линейно неразделимых данных. Поместите скриншоты для этих случаев в отчёт.



##### **Задание 2**

Модифицируйте программу из задания 1 следующим образом.

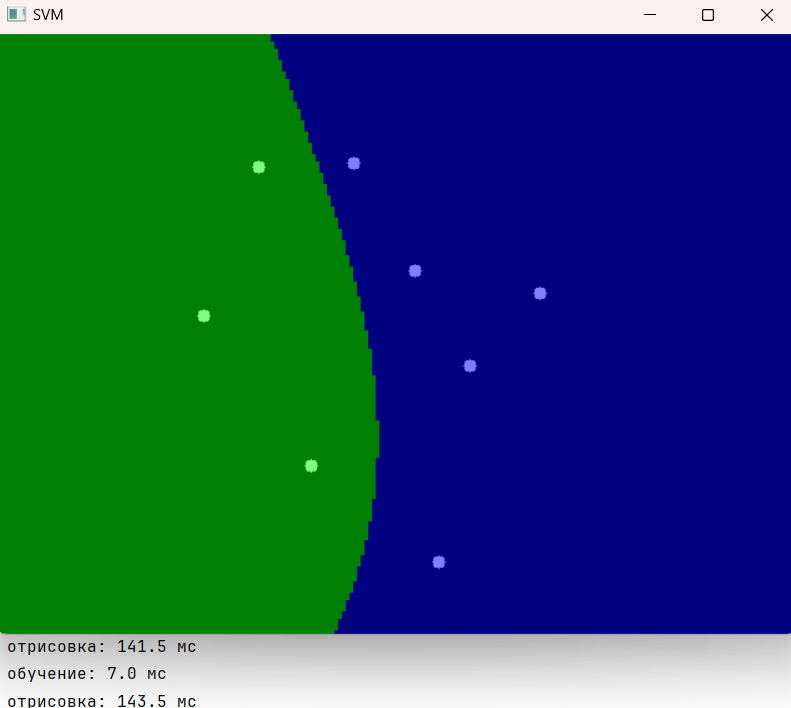
* При выходе программа должна сохранить результат обучения классификатора в файл **svm.xml**.
* При запуске программа должна пытаться загрузить созданный файл (функция **cv2.ml.SVM\_load()**) и вывести на экран карту классов и опорные вектора.



##### **Задание 3**

Модифицируйте программу из задания 1 следующим образом. Используйте ядро **cv2.ml.SVM\_RBF** и вызовите метод **.trainAuto()** вместо **.train()** для автоматического подбора параметров ядра.

Рассмотрите поведение программы для случая линейно разделимых и линейно неразделимых данных. Поместите скриншоты для этих случаев в отчёт.



##### **Дополнительное задание**

Модифицируйте программу из задания 3 следующим образом. Нажатия кнопок 1-5 должны выбирать класс точек (0-4). Левый клик должен ставить новую точку выбранного класса. Правый клик должен удалять последнюю поставленную точку. Цвета для отображения точек и их "зон влияния" выберите на ваше усмотрение.

Рекомендации по выполнению задания:

* Оберните вызов **self.svm.trainAuto()** в блок try-except. Наличие только одной точки в каком-либо классе приведёт к ошибке обучения.
* Храните отдельно список координат точек и список их классов. Это упростит подготовку данных для SVM.
* Простой способ определить новый номер класса - вычесть ord('1') из кода клавиши, полученного от **cv2.waitKey()**. Для клавиш 1-5 разность будет в диапазоне 0-4.

