

# Установка для распознавания образов

На первой практической работе по теме "Распознавание образов" обычно нужно установить следующие компоненты и инструменты, которые помогут в выполнении лабораторных задач:

## 1. Python

Это основной язык программирования для работы с распознаванием образов. Убедитесь, что у вас установлена актуальная версия Python (рекомендуется 3.8 и выше).

## 2. Библиотеки для машинного обучения и распознавания образов

Установите следующие библиотеки с помощью менеджера пакетов `pip`:

- **NumPy** — для работы с многомерными массивами и научными вычислениями:

```
bash
```

```
pip install numpy
```

- **Matplotlib** — для визуализации изображений и графиков:

```
bash
```

```
pip install matplotlib
```

- **Pillow (PIL)** — библиотека для работы с изображениями:

```
bash
```

```
pip install pillow
```

- **OpenCV** — популярная библиотека для компьютерного зрения и обработки изображений:

```
bash
```

```
pip install opencv-python
```

- **Scikit-learn** — для выполнения базовых задач машинного обучения:

```
bash
```

```
pip install scikit-learn
```

- **TensorFlow** или **PyTorch** — для создания нейронных сетей и глубокого обучения (можно выбрать одну из этих библиотек):
  - TensorFlow:

```
bash
```

```
pip install tensorflow
```

- PyTorch:

```
bash
```

```
pip install torch torchvision
```

### 3. Jupyter Notebook

Для удобного выполнения практической работы рекомендуется использовать Jupyter Notebook. Это интерактивная среда, где можно запускать код по частям, анализировать результаты и строить графики:

```
bash
```

```
pip install notebook
```

### 4. Среда разработки

Вы можете использовать любой удобный текстовый редактор или IDE, такие как:

- **PyCharm**
- **VSCode**
- **Jupyter Notebook** (как указано выше)

### 5. Датасет для практики

Для распознавания образов нужен набор изображений. Можно использовать заранее подготовленный датасет или загрузить его с одного из популярных ресурсов:

- **MNIST** — набор рукописных цифр для обучения моделей.
- **CIFAR-10** — набор изображений 10 классов для задач классификации.
- **ImageNet** — для более сложных задач.

## Пример команды для загрузки датасета (например, MNIST):

python

```
from tensorflow.keras.datasets import mnist
(x_train, y_train), (x_test, y_test) = mnist.load_data()
```