

Telegram bot

# Цель

- Сервис для поиска людей по фотографии

# Задачи

- Сбор и подготовка данных
- Детектирование и локализация лица
- Извлечение дескрипторов изображения
- Обучение модели
- Доступ через интерфейс бота

# Python

- Прост
- Удобен
- Разнообразие библиотек

# Github

- Все-таки Github, потому что он родной и удобный

# Название

- Telegram обязует добавлять 'bot' в конец имени бота
- Мы выбрали FindyBot

# Вконтакте

- Доступный API
- Мы ориентированы пока на Россию
- Большинство жителей России использует VK

# Machine learning

- ML - раздел математики и статистики, исследующий проблемы анализа данных
- Изучение основных алгоритмов машинного обучения
- Поиск библиотек с реализацией этих алгоритмов



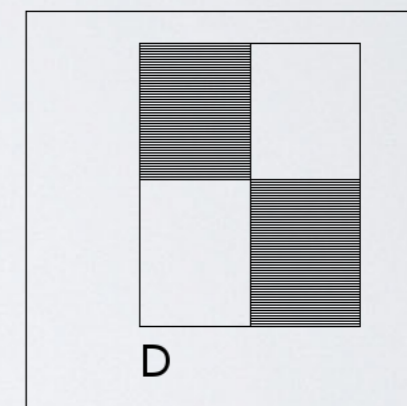
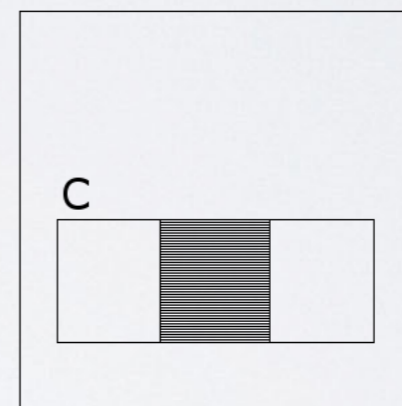
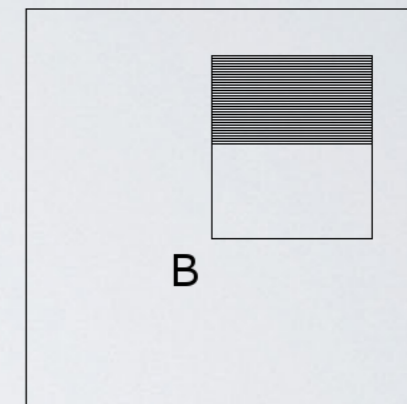
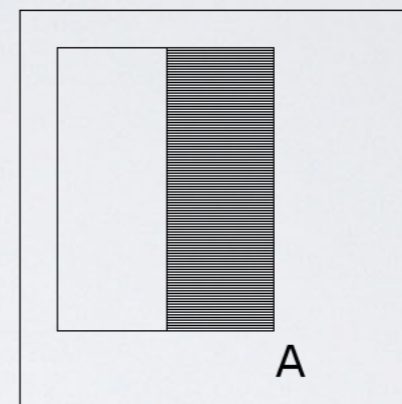
# Подготовка данных

- Чьих данных?
- У нас всего 250 ГБ
- Как распознавать?
- Где лицо?
- Че дальше?
- Мало!
- Всех пользователей!
- Друзей... их друзей и т.д.
- По лицу!
- Детектор лиц найдет
- Extend, crop, resize
- Размножим! (Random move, rotate)

# Детектирование лица

## Viola-Jones Cascade Classifier

- Использование простейших классификаторов для построения сложного
- Фильтры Хоара
- Скользящее окно



# Извлечение дескрипторов

Histogram of oriented gradients (HOG)

1. Image normalization (power-law compression)
2. First-order gradient detection
3. Local details encoding
4. Normalization across blocks
5. Conversion into a feature vector

# Глаза и нос

- Blob detection для локализации зрачков
- Эвристика расположения носа

# Распознавание лица

- Идеи:
- Подготовить базу лиц и кластеризовать
- Обучить классификатор на каждом профиле
- Использовать нейронную сеть для выделения индивидуальных черт
- DAISY, ORB?

# Классификатор

- SVM (Support Vector Machine)
- Linear, RBG kernels
- Grid search with 5-fold cross-validation

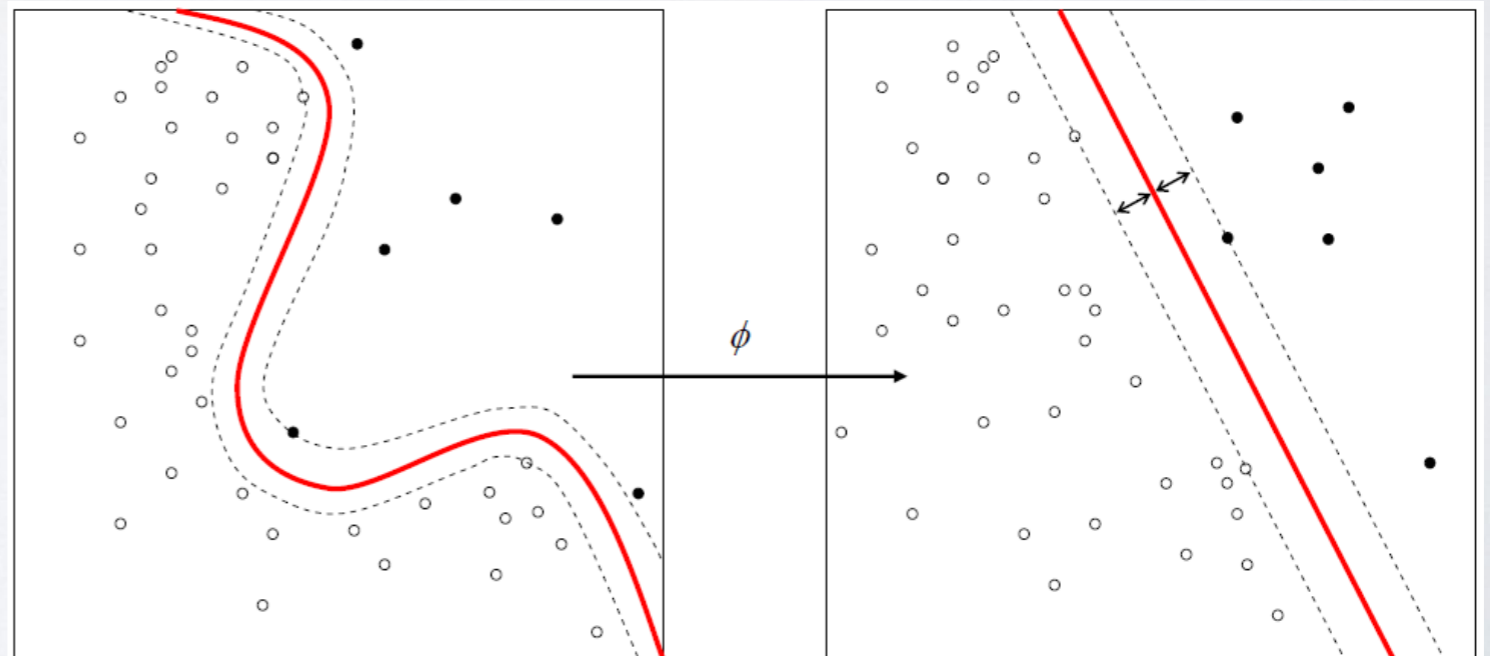
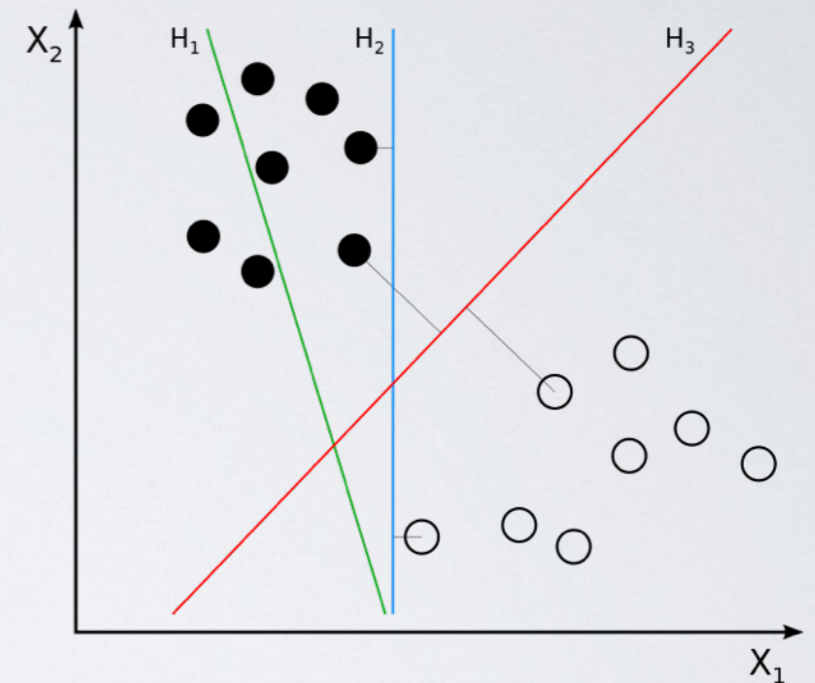
# SVM

Минимизация эмпирического риска

$$\hat{\varepsilon}(f) = \frac{1}{n} \sum_{k=1}^n \ell(y_k, f(X_k))$$

rbf (radial basis function)

$$k(\vec{x}_i, \vec{x}_j) = \exp(-\gamma \|\vec{x}_i - \vec{x}_j\|^2)$$



# Результаты

algorithm	#images	#classes	accuracy score
HOG+SVM	7K	25	77 %
DAISY+BoW +SVM	~500	25	52 %



# Перспективы

- Подключение Instagram, Facebook, Telegram
- Расширение хранилища и людей для поиска
- Расширение возможностей:
  - Распознавание животных
  - Распознавание авто