

# Computer Vision, Part 2

---

CST 205

# Machine learning

---

- Another important aspect of `OpenCV`
- Ability of computers to learn without being explicitly programmed.
- **Pattern Recognition** is a branch of machine learning focused on the recognition of patterns and regularities in data.
  - “Teach” the computer to look for patterns based on labeled training data (supervised learning)

# OpenCV and Machine Learning

---

- A classifier tells OpenCV what to look for in images.
- OpenCV includes a variety of example classifiers

**RUSSIAN  
LICENSE  
PLATES!**



# Face Detection

---

- Detection tells us where the face is, not of whom the face is.
- Paul Viola and Michael Jones invented a popular real-time face detection algorithm in 2001.
- Viola-Jones requires a full-view of front-facing, upright faces.
- Viola-Jones works best on low-resolution, grayscale images.

# Aside: Drawing rectangles

---

- OpenCV Drawing Functions
  - To mark areas of detection, we commonly use rectangles
    - `rectangle()` method takes 5 arguments:
      - the image
      - one corner of rectangle
      - opposite corner of rectangle
      - the color (BGR)
      - thickness of line

# Example

---

```
import cv2
img = cv2.imread('jeanne.png')

cv2.rectangle(
    img,
    (185, 254),
    (265, 334),
    (0,255,0),
    2
)

cv2.imshow("Rectangled", img)
cv2.waitKey()
```



# OpenCV drawing functions

---

- In addition to rectangles, OpenCV has functions for drawing circles, lines, arrows, and more.
- OpenCV can also place text on images.

```
cv2.putText(  
    img, 'Jeanne', (135, 500),  
    cv2.FONT_HERSHEY_SCRIPT_COMPLEX,  
    2, (0, 245, 245)  
)
```



# Back to face detection

---

- Viola-Jones uses a “Haar cascade” based on Haar wavelets
- Haar-like feature is a rectangular pattern of data
- Five Haar-like features commonly used in Viola-Jones for face detection:

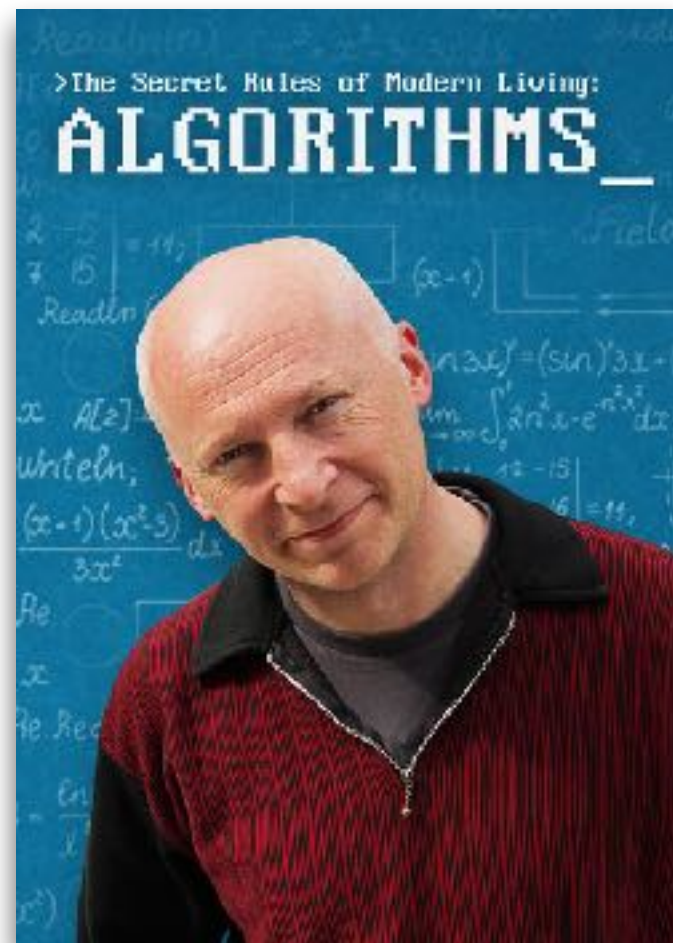




# Recommended viewing

---

- The Secret Rules of Modern Living: Algorithms



(available on Netflix)

# detectMultiScale() method

---

- `searchScaleFactor`
  - Amount by which to rescale the image (scale pyramid). For example, a value of 1.1 will gradually reduce the size of the input image by 10 percent, making it more likely for a face to be found than a larger value.
- `minNeighbors`
  - The number of neighbors each candidate rectangle should maintain. Typically, choose 3 or 5. Lower values result in more false positives.
- Returns an array of bounding boxes.