

# «Анализ изображений на основе реализации алгоритмов и методов компьютерного зрения»

Подготовила:

студентка 2 курса 1 группы

факультета информационных технологий

Кашперко Василиса Сергеевна

Руководитель: Игнаткова Янина Алексеевна



# Цель:

ознакомление с компьютерным зрением, его задачами и практическим применением, провести анализ изображений на основе реализации алгоритмов и методов компьютерного зрения

# Задачи:

- Дать определение компьютерному зрению;
- Выяснить, какие задачи стоят перед компьютерным зрением;
- Изучить доступные Интернет-ресурсы;
- Изучить области применения компьютерного зрения;
- Рассмотреть примеры применения компьютерного зрения на практике.



Что такое компьютерное зрение?







# История развития компьютерного зрения

Компьютерное зрение сформировалось как независимая дисциплина еще к концу 60-х годов.

Это течение появилось в пределах искусственного интеллекта тогда, когда еще велись жаркие дискуссии о вероятности сотворения мыслящей машины. Оно появилось из трудов по распознаванию образов.

# Первые эксперименты

1958 г. - **Фрэнк Розенблатт**, психолог из Корнеллского университета, сотворил компьютерное воплощение перцептрона (от «*perception*» - *восприятие*) - приспособления, имитирующего схему толкования образов человеческим мозгом.



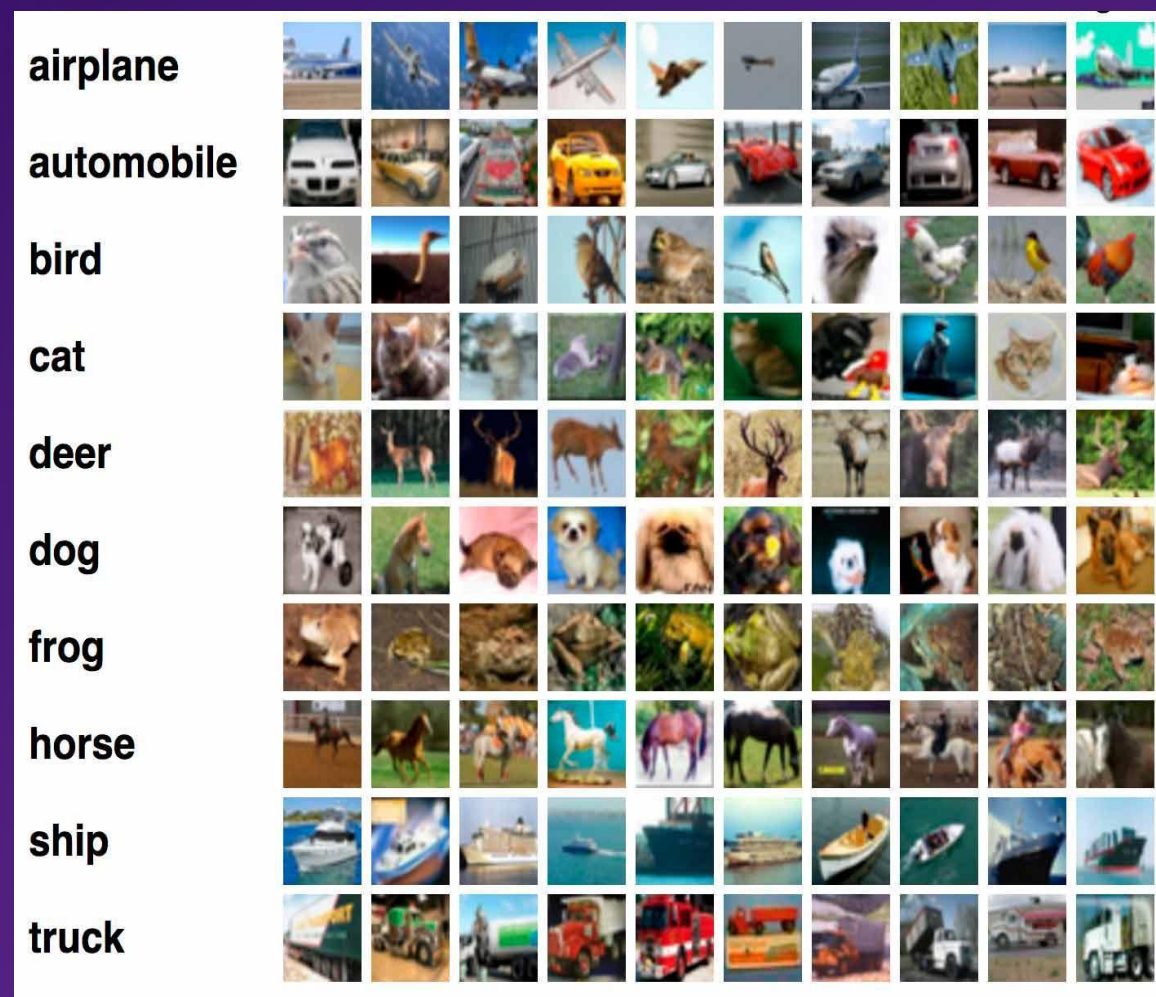
# Перцептрон

Перцептрон был впервые создан в 1958 году, вдобавок его подготовка занимала около получаса машинного времени на ЭВМ IBM-704. Аппаратный вариант - Mark I Perceptron – был сконструирован в 1960 г. и применялся для толкования зрительных образов.





В 2012 году на конкурсе **ImageNet** сверточная нейронная сеть **AlexNet** вошла в топ-5 алгоритмов с уровнем ошибок 15,3%. В 2015 году нейросеть победила в конкурсе. Именно это событие считается отправной точкой в современной истории компьютерного зрения.





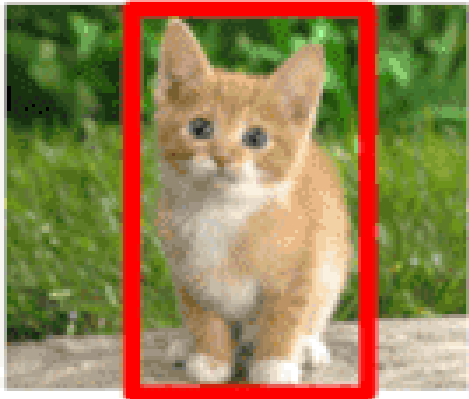
Цель компьютерного зрения - принятие решений о реальных физических объектах и сценах, основываясь на воспринимаемых изображениях.

**Классификация**



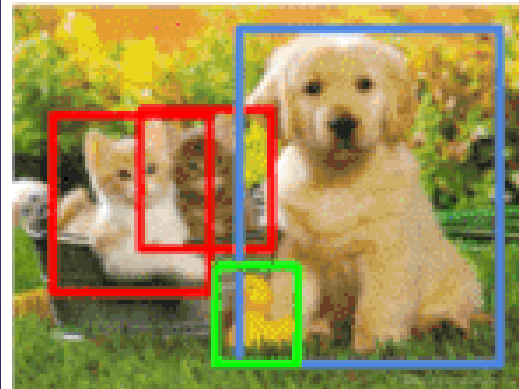
CAT

**Локализация**



CAT

**Детектирование**



CAT, DOG, DUCK

**Сегментация**



CAT, DOG, DUCK

# Источники изображений



Компьютерное зрение  
теснейшим образом  
взаимодействует с  
областью обработки  
изображений, часто трудно  
однозначно отнести  
возникающие задачи и  
применяемые методы  
решения к одной из этих  
областей.

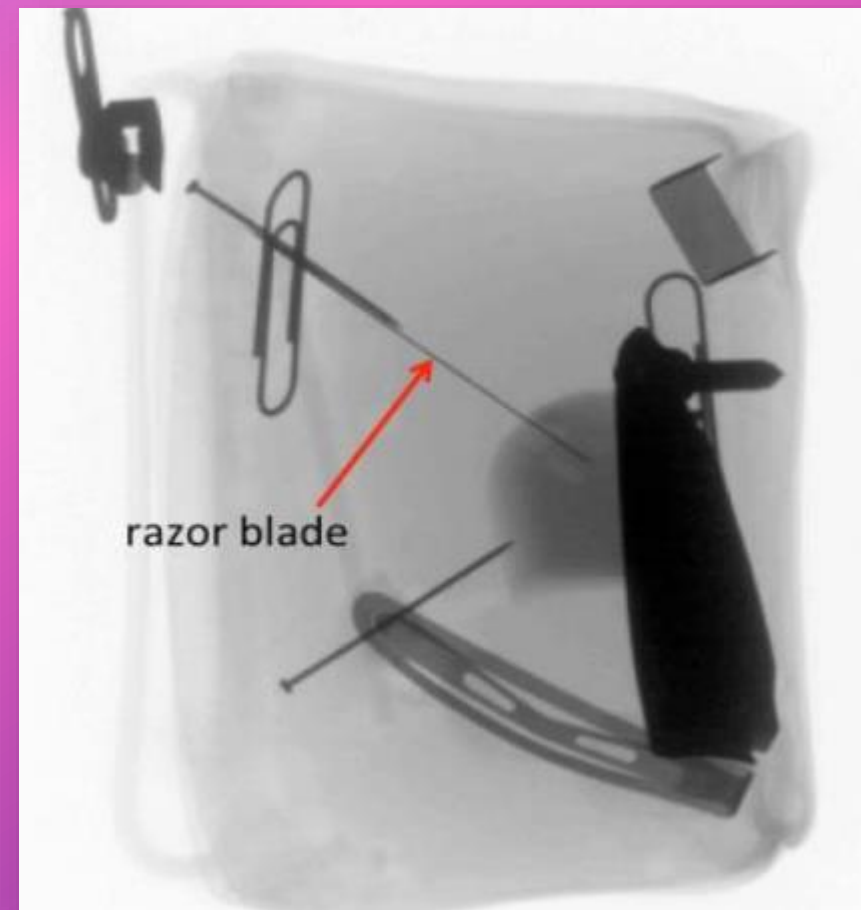




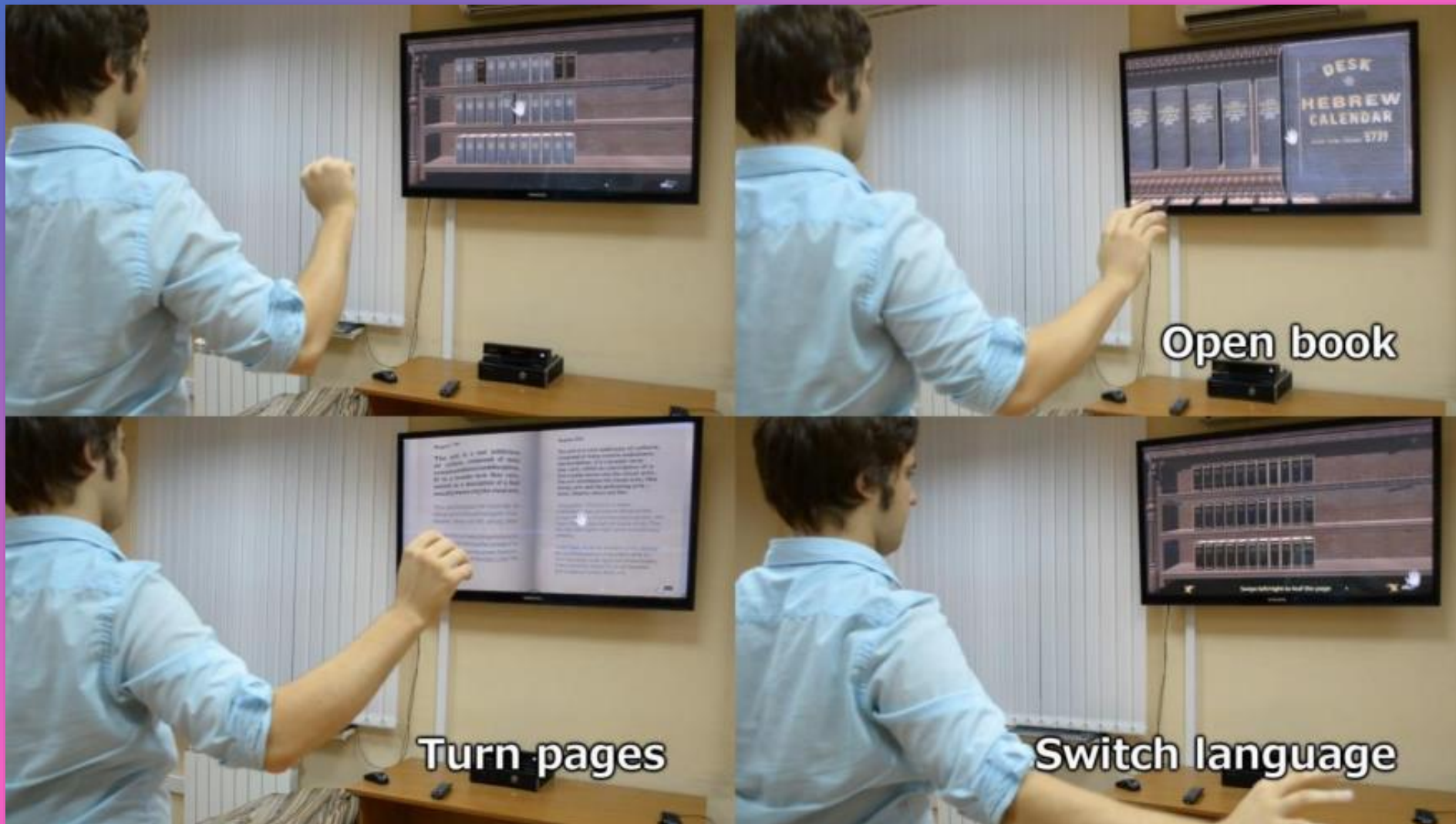
# Область применения CV в современном мире



# Системы безопасности



# Распознавание жестов





# Распознавание текста

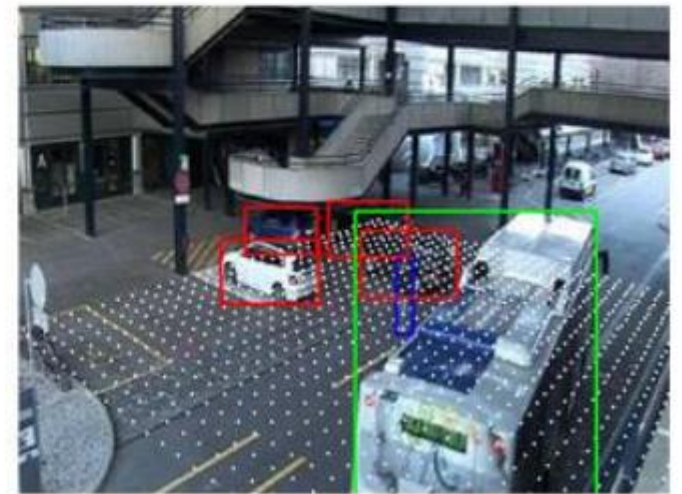
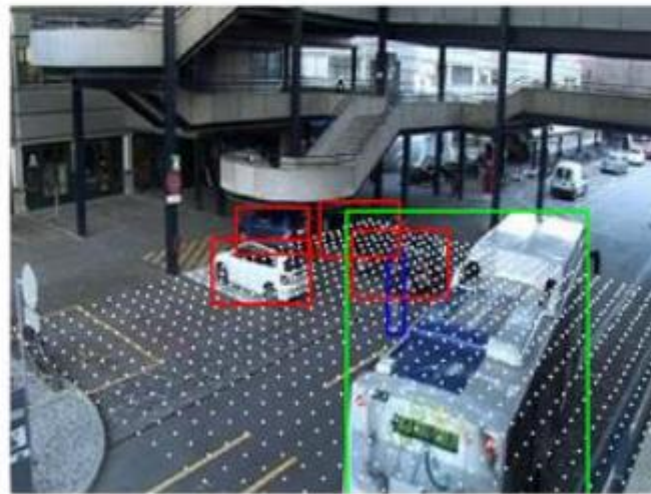
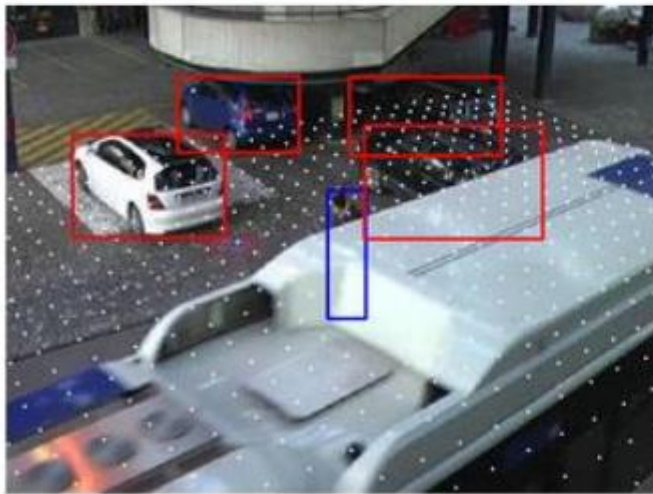
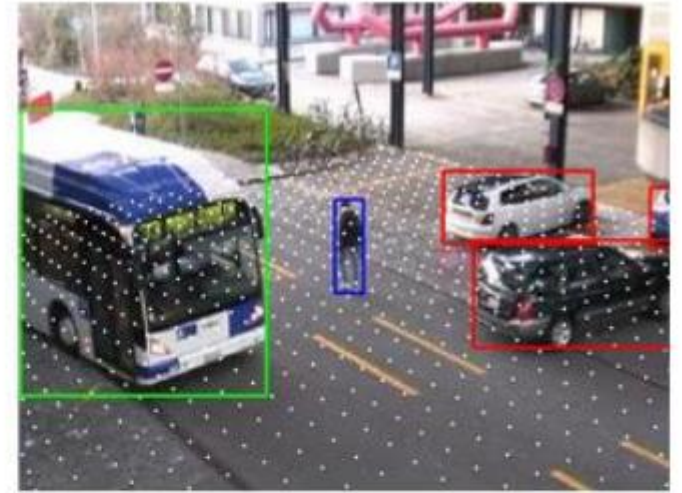
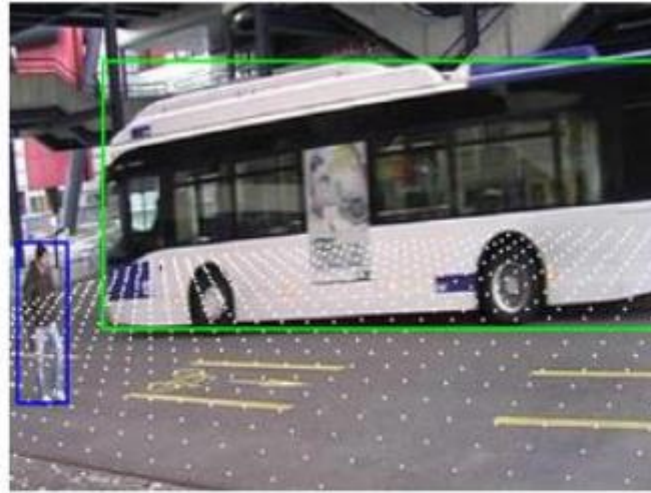
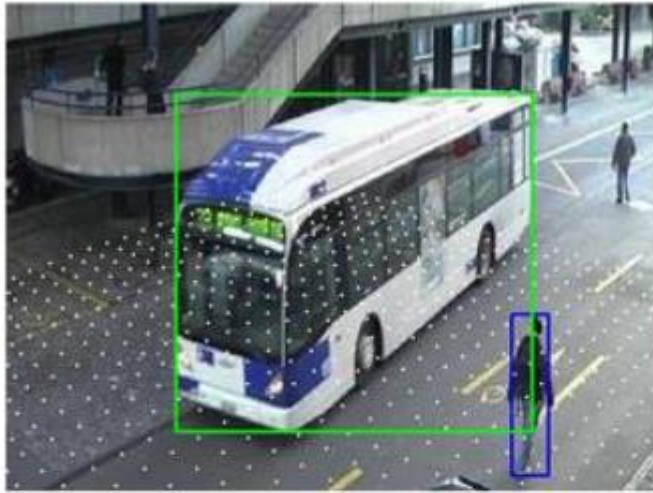
Hand writing recognition

$$E = mc^2$$

Compute with  WolframAlpha

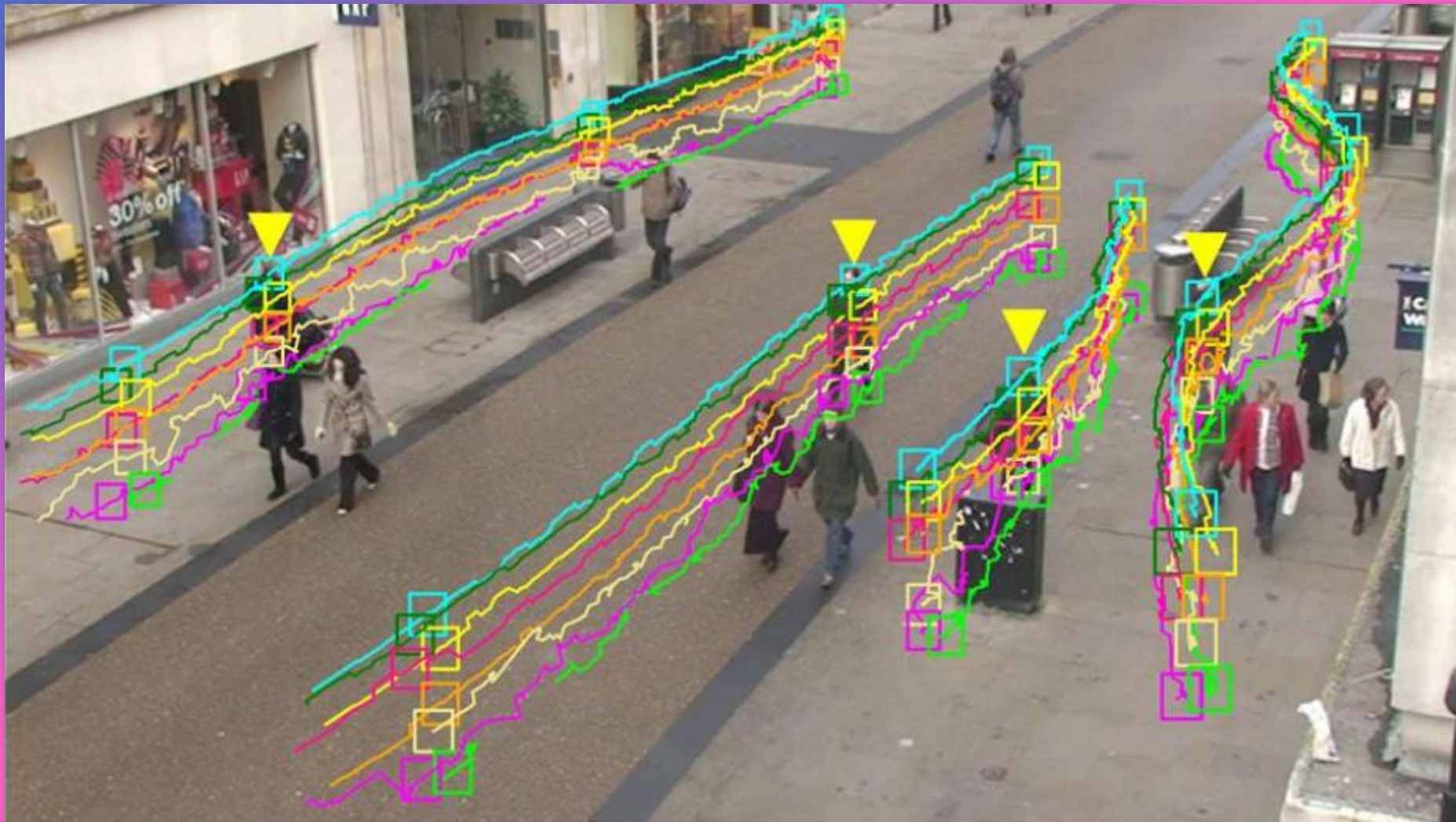
$$E = mc^2$$

# Отслеживание перемещения человека





# Отслеживание перемещения человека





# Рендеринг на основе изображения



Super-resolution



High dynamic range



Enhanced exposure



Object touchup



Object removal

# Дополненная реальность



Headtracking

# **Webcam Headtracking**



# Пример разработки программы

```
Командная строка
Microsoft Windows [Version 10.0.19041.1415]
(c) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

C:\Users\Vasilisa>python --version
Python 3.10.4

C:\Users\Vasilisa>pip -V
pip 22.0.4 from C:\Users\Vasilisa\AppData\Local\Programs\Python\Python310\lib\site-packages\pip (python 3.10)

C:\Users\Vasilisa>python
Python 3.10.4 (tags/v3.10.4:9d38120, Mar 23 2022, 23:13:41) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import cv2
>>> print(cv2.__version__)
4.5.5
>>> ^Z




C:\Users\Vasilisa>
```

```
Командная строка
Microsoft Windows [Version 10.0.19041.1415]
(c) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

C:\Users\Vasilisa>pip install opencv-python
Requirement already satisfied: opencv-python in c:\users\vasilisa\appdata\local\programs\python\python310\lib\site-packages (4.5.5.64)
Requirement already satisfied: numpy>=1.21.2 in c:\users\vasilisa\appdata\local\programs\python\python310\lib\site-packages (from opencv-python) (1.22.3)

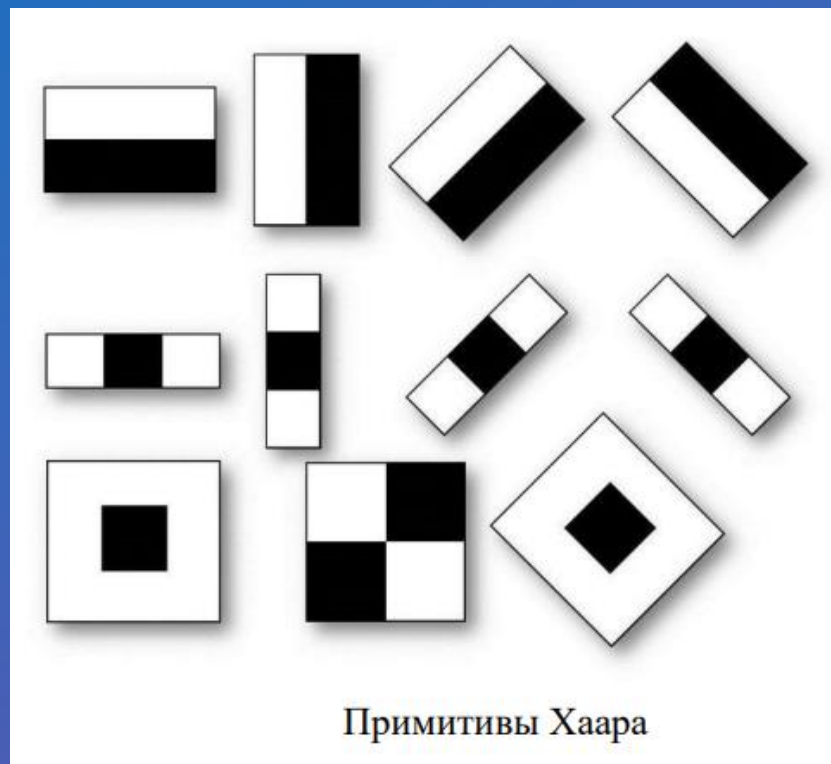
C:\Users\Vasilisa>
```

# Пример разработки программы

 haarcascade_smile	xml
 haarcascade_frontalface_default	xml
 haarcascade_mcs_mouth	xml

```
- <opencv_storage>
- <cascade type_id="opencv-cascade-classifier">
  <stageType>BOOST</stageType>
  <featureType>HAAR</featureType>
  <height>24</height>
  <width>24</width>
  - <stageParams>
    <maxWeakCount>211</maxWeakCount>
  </stageParams>
  - <featureParams>
    <maxCatCount>0</maxCatCount>
  </featureParams>
  <stageNum>25</stageNum>
  - <stages>
    - <_>
      <maxWeakCount>9</maxWeakCount>
      <stageThreshold>-5.0425500869750977e+00</stageThreshold>
      - <weakClassifiers>
        - <_>
          <internalNodes> 0 -1 0 -3.1511999666690826e-02</internalNodes>
          <leafValues> 2.0875380039215088e+00 -2.2172100543975830e+00</leafValues>
        </_>
        - <_>
          <internalNodes> 0 -1 1 1.2396000325679779e-02</internalNodes>
          <leafValues> -1.8633940219879150e+00 1.3272049427032471e+00</leafValues>
        </_>
        - <_>
          <internalNodes> 0 -1 2 2.1927999332547188e-02</internalNodes>
          <leafValues> -1.5105249881744385e+00 1.0625729560852051e+00</leafValues>
        </_>
        - <_>
          <internalNodes> 0 -1 3 5.7529998011887074e-03</internalNodes>
          <leafValues> -8.7463897466659546e-01 1.1760339736938477e+00</leafValues>
        </_>
        - <_>
          <internalNodes> 0 -1 4 1.5014000236988068e-02</internalNodes>
          <leafValues> -7.7945697307586670e-01 1.2608419656753540e+00</leafValues>
        </_>
      </weakClassifiers>
    </_>
  </stages>
</cascade>
</opencv_storage>
```

# Пример разработки программы





# Пример разработки программы

Original

5	2	3	4	1
1	5	4	2	3
2	2	1	3	4
3	5	6	4	5
4	1	3	2	6

Integral

5	7	10	14	15
6	13	20	26	30
8	17	25	34	42
11	25	39	52	65
15	30	47	62	81

$$5 + 2 + 3 + 1 + 5 + 4 = 20$$

Original

5	2	3	4	1
1	5	4	2	3
2	2	1	3	4
3	5	6	4	5
4	1	3	2	6

Integral

5	7	10	14	15
6	13	20	26	30
8	17	25	34	42
11	25	39	52	65
15	30	47	62	81

$$5 + 4 + 2 + 2 + 1 + 3 = 17$$

$$34 - 14 - 8 + 5 = 17$$

# Пример разработки программы

```
1 import numpy as np
2 import cv2
3 import time
4
5
6 def main():
7     face_cascade = cv2.CascadeClassifier('data\\xml\\haarcascade_frontalface_default.xml')
8     mouth_cascade = cv2.CascadeClassifier('data\\xml\\haarcascade_mcs_mouth.xml')
9     smile_cascade = cv2.CascadeClassifier('data\\xml\\haarcascade_smile.xml')
10
11     bw_threshold = 127
12     weared_mask = "WITH MASK"
13     not_weared_mask = "WITHOUT MASK"
14
15     cap = cv2.VideoCapture(0)
16
17     pTime = 0
18
19     while 1:
20         ret, img = cap.read()
21         img = cv2.flip(img, 1)
22         wn = "Mask detection"
23         cv2.namedWindow(wn, cv2.WINDOW_NORMAL)
24         gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
25         (thresh, black_and_white) = cv2.threshold(gray, bw_threshold, 255, cv2.THRESH_BINARY)
26         faces = face_cascade.detectMultiScale(gray, 1.1, 10)
27         faces_bw = face_cascade.detectMultiScale(black_and_white, 1.1, 10)
28
29         if(len(faces) == 0 and len(faces_bw) == 0):
30             cv2.putText(img, "No face found", (10, 30), cv2.FONT_HERSHEY_PLAIN, 1, (255, 255, 255), 1, cv2.LINE_AA)
31         else:
32             for (x, y, w, h) in faces:
33                 x1, y1 = x + w, y + h
34                 couter = 0
35                 colorLine = (0, 0, 0)
36                 color = (0, 0, 0)
37                 mouth = mouth_cascade.detectMultiScale(gray, 2, 10)
38                 for (mx, my, mw, mh) in mouth:
39                     if(x < mx and y < my and y < my + mh):
40                         cv2.rectangle(img, (mx, my), (mx + mw, my + mh), (255, 255, 0), 2)
41                         couter+=1
42                 smile = smile_cascade.detectMultiScale(gray, 2, 15)
43                 for (sx, sy, sw, sh) in smile:
44                     if(x < sx and y < sy):
45                         cv2.rectangle(img, (mx, my), (mx + mw, my + mh), (255, 255, 0), 2)
46                         couter+=1
```

```
47
48         if(couter > 0):
49             cv2.putText(img, not_weared_mask, (x, y - 10), cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 0, 255), 1, cv2.LINE_AA)
50             colorLine = (59, 59, 217)
51             color = (0, 0, 255)
52         else:
53             cv2.putText(img, weared_mask, (x, y - 10), cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 255, 0), 1, cv2.LINE_AA)
54             colorLine = (108, 181, 60)
55             color = (0, 255, 0)
56
57         cv2.rectangle(img, (x, y), (x1, y1), colorLine, 1)
58         cv2.line(img, (x, y), (x + 30, y), color, 3)
59         cv2.line(img, (x, y), (x, y + 30), color, 3)
60         cv2.line(img, (x1, y), (x1 - 30, y), color, 3)
61         cv2.line(img, (x1, y), (x1, y + 30), color, 3)
62         cv2.line(img, (x, y1), (x + 30, y1), color, 3)
63         cv2.line(img, (x, y1), (x, y1 - 30), color, 3)
64         cv2.line(img, (x1, y1), (x1 - 30, y1), color, 3)
65         cv2.line(img, (x1, y1), (x1, y1 - 30), color, 3)
66
67         cTime = time.time()
68         fps = 1 / (cTime - pTime)
69         pTime = cTime
70
71         #cv2.putText(img, f'FPS: {int(fps)}', (10, 15), cv2.FONT_HERSHEY_PLAIN, 1, (0, 255, 0), 2)
72         cv2.imshow(wn, img)
73         cv2.imshow("ddwa", black_and_white)
74
75         k = cv2.waitKey(1)
76         if k == 27:
77             break
78
79     cap.release()
80     cv2.destroyAllWindows()
81
82 if __name__ == "__main__":
83     main()
```

# Компьютерное зрение Vs Обработка изображений

**Обработка изображений** изучает преобразование изображения в изображение. Вход и выход обработки изображения – оба изображения.

**Компьютерное зрение** – это построение явных, значимых описаний физических объектов по их изображению. Результатом компьютерного зрения является описание или интерпретация структур в трехмерной сцене.



# Разница между компьютерным зрением (CV) и компьютерной графикой (CG)

Компьютерная графика (Computer Graphics) и Computer Vision (Computer Vision) - это два направления в одном процессе.

CG преобразует абстрактную семантическую информацию в изображения, а CV извлекает абстрактную семантическую информацию из изображений.

# Заключение

В заключение я постараюсь дать ответ на вопрос: «Зачем компьютеру зрение?».

Научить компьютер видеть мир — значит получить надежного помощника, который день и ночь будет занят решением важных задач. Некие опасения, конечно, присутствуют, но объективная польза от технологий и будущий прогресс, которые они могут дать, перевешивают доводы из теории «восстания машин». Компьютерное зрение — уникальный инструмент, и, рассмотренные примеры использования компьютерного зрения - это лишь малая доля того, на что способен «прозревший» компьютер!

Спасибо за внимание!