



Неделя 1, Урок 1  
15 января 2018





## Welcome speech





CVT



KBTU



KBTU



SDU

Нургабылов Ернияз,  
окончил бакалавр  
computer science в НУ.  
Работаю в CVT.  
Участвовал в разработке  
“Сергек”. Олимпиадчик по  
математике, призовые  
места на IMO.  
Полуфиналист ACM.

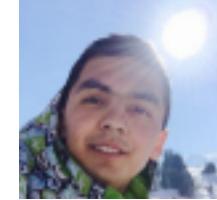
Бобур Мухсимбаев,  
Магистратура  
интеллектуальные системы  
в КБТУ. Курс по DL в  
University of Southampton.  
Работал над проектами в  
CV. Исследования по  
Альтернативному скорингу.  
tensorflow, torch, sklearn,  
scipy.

Зарина Улықпанова,  
окончила магистратуру по  
специализации  
интеллектуальные системы  
в КБТУ, прошла курс по DL  
в University of Southampton.  
Работала над задачами NP-  
сложности, используя  
эвристические алгоритмы.  
Используемый стек  
tensorflow, theano, sklearn,  
hadoop.

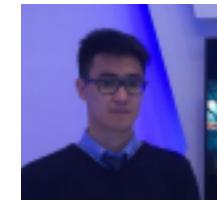
Калмахан Артықбаев, 4  
курс SDU. Исследования в  
DL и ML. Проходил  
практику research intern в  
UMD, USA. Исследование в  
computer vision, cnn и rnn.  
Есть хороший опыт работы  
с Tensorflow, Caffe, theano,  
Sklearn, scipy, django.  
Интересно работать с  
хардкорными задачами в  
ML)



Live stream

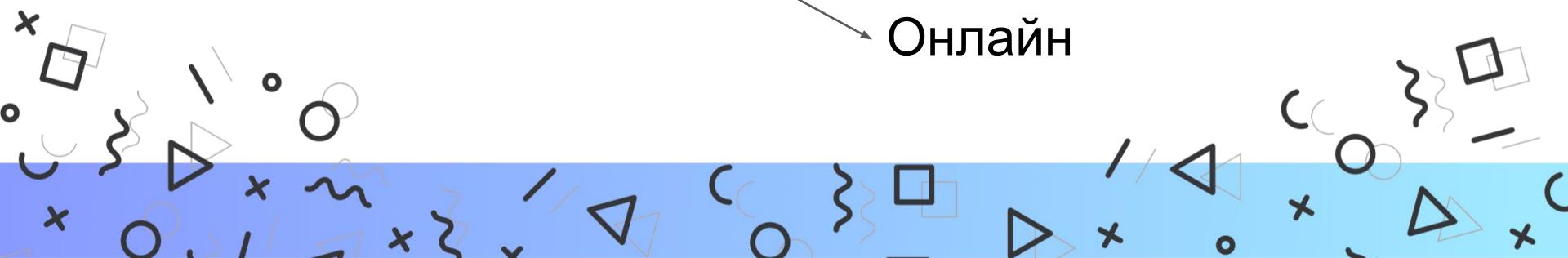


Live stream



Live stream

Онлайн



# Что такое “Computer Vision”?

## И что оно изучает?



- Computer Vision
- Image processing
- Computational Photography



# Каждый рисунок рассказывает историю



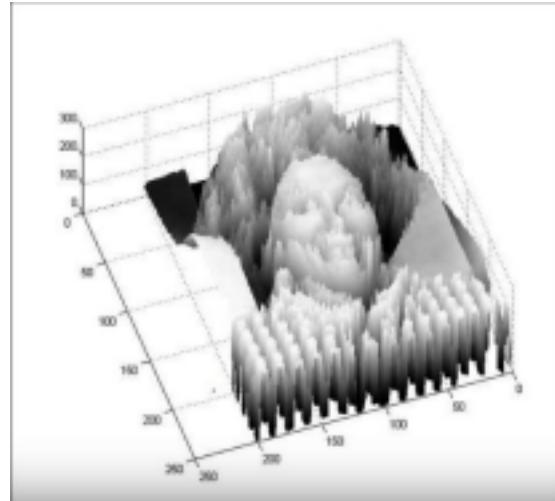
И задача CV создать программу способную интерпретировать изображение

# Изображение как функция



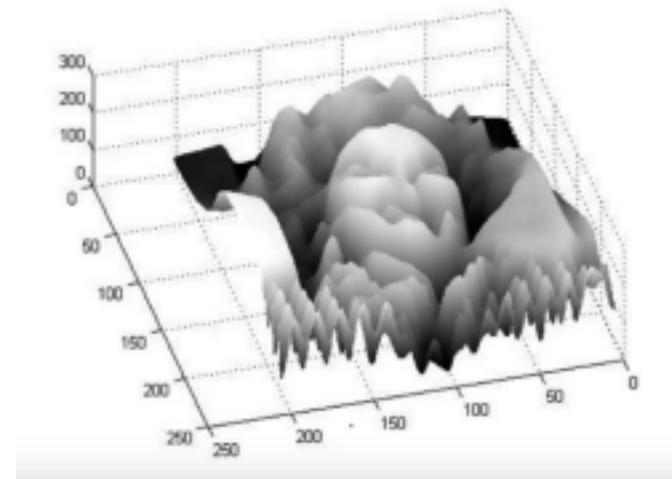
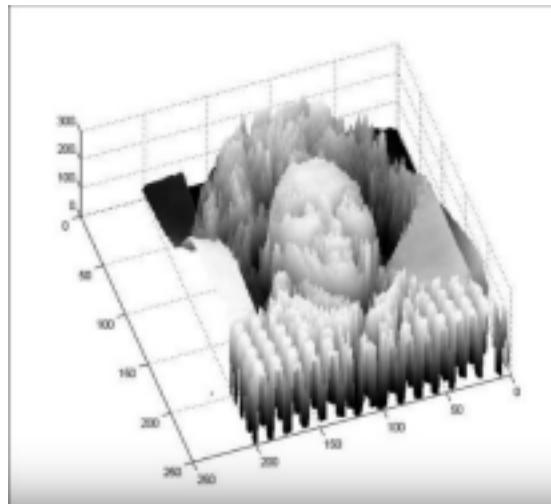
Introduction to Computer Vision on [udacity.com](https://udacity.com)

# Изображение как функция



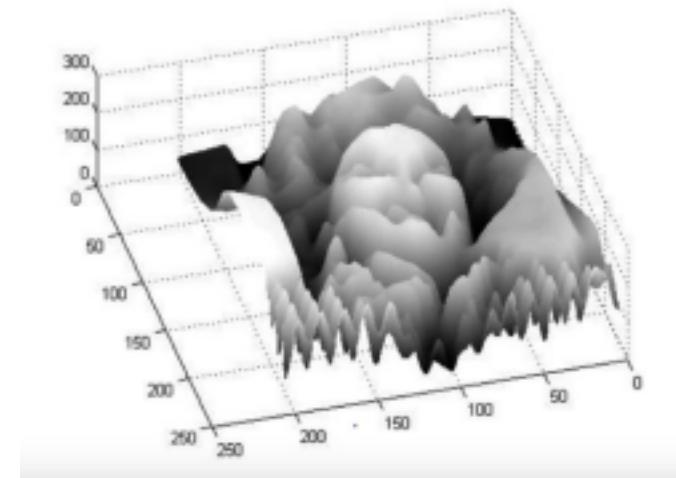
Introduction to Computer Vision on [udacity.com](https://udacity.com)

# Изображение как функция (сглаживание)



Introduction to Computer Vision on [udacity.com](https://udacity.com)

# Изображение как функция (сглаживание)



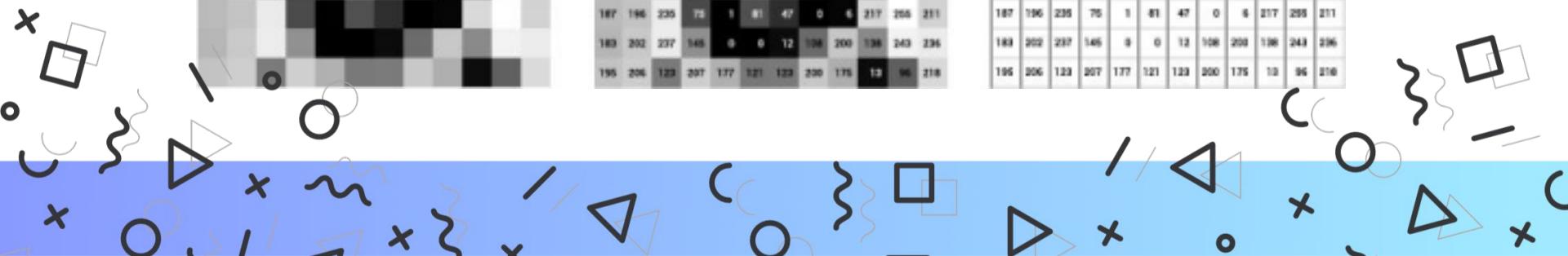
Introduction to Computer Vision on [udacity.com](https://udacity.com)

# Как компьютер видит изображение

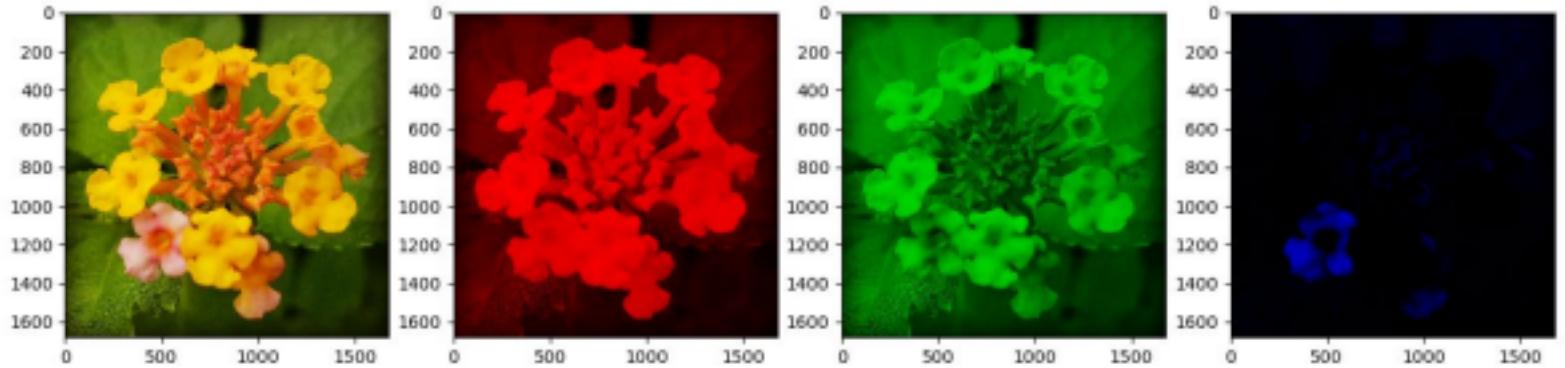


167	163	174	168	160	162	129	161	172	161	165	166
185	182	163	74	75	62	33	17	110	210	180	154
180	186	60	14	94	6	10	30	48	196	169	181
206	108	5	124	131	111	120	204	166	16	56	180
194	48	137	261	237	239	239	226	237	87	71	231
172	106	267	233	233	214	220	230	230	98	74	236
188	68	179	209	185	215	211	156	139	75	20	189
189	97	166	84	13	168	134	11	35	52	22	148
199	148	191	193	158	227	178	148	182	196	36	190
205	174	185	262	236	231	149	178	238	49	55	234
190	216	196	149	236	187	86	150	79	38	218	241
190	234	147	108	237	210	137	103	36	191	265	224
190	214	173	46	103	143	95	56	2	199	249	219
187	196	235	79	1	81	47	0	6	217	255	211
183	202	237	146	9	6	12	104	200	138	243	236
195	206	123	297	177	121	123	296	175	13	96	218

167	163	174	168	160	162	129	161	172	161	165	166
199	182	163	74	75	62	33	17	110	210	180	154
189	180	60	14	94	6	10	30	48	196	169	181
206	109	5	124	131	111	120	204	166	16	56	180
194	48	137	261	237	239	239	226	237	87	71	231
172	106	267	233	233	214	220	230	230	98	74	236
188	68	179	209	185	215	211	156	139	75	20	189
189	97	166	84	13	168	134	11	35	52	22	148
199	148	191	193	158	227	178	148	182	196	36	190
205	174	185	262	236	231	149	178	238	49	55	234
190	216	196	149	236	187	86	150	79	38	218	241
190	234	147	108	237	210	137	103	36	191	265	224
190	214	173	46	103	143	95	56	2	199	249	216
187	196	235	79	1	81	47	0	6	217	255	211
183	202	237	146	9	6	12	104	200	138	243	236
195	206	123	297	177	121	123	296	175	13	96	218



# Каналы изображения



# Черно-белое изображение



$$Y = 0.299 R + 0.587 G + 0.114 B$$



# Форматы изображения (пикселя)

	C1	C2	C3	C4
CV_8U	0	8	16	24
CV_8S	1	9	17	25
CV_16U	2	10	18	26
CV_16S	3	11	19	27
CV_32S	4	12	20	28
CV_32F	5	13	21	29
CV_64F	6	14	22	30

# Форматы изображения (пикселя)

	C1	C2	C3	C4
CV_8U	0	8	16	24
CV_8S	1		17	25
CV_16U	2		18	26
CV_16S	3		19	27
CV_32S	4		20	28
CV_32F	5	13	21	29
CV_64F	6	14	22	30

# Практика

1. Посмотреть маленькое изображение 25x25 пикселей
2. Показать BGR каналы изображения
3. Создать белое одноканальное изображение 50x50 пикселей
4. Закрасить центр изображения черным цветом 10x10 пикселей
5. Посчитать черно-белое изображение,  $Y=0.299R+0.587G+0.114B$  (вручную)
6. Посчитать черно-белое изображение (opencv function)



# Фильтры



# Самый простой фильтр

Входной массив

1	2	1	1	2	1
---	---	---	---	---	---

0.5	0.5
-----	-----

0.5	0.5
-----	-----

0.5	0.5
-----	-----

Фильтр

x	0.5	x	0.5
---	-----	---	-----

$$1 \times 0.5 + 2 \times 0.5 = 1.5$$

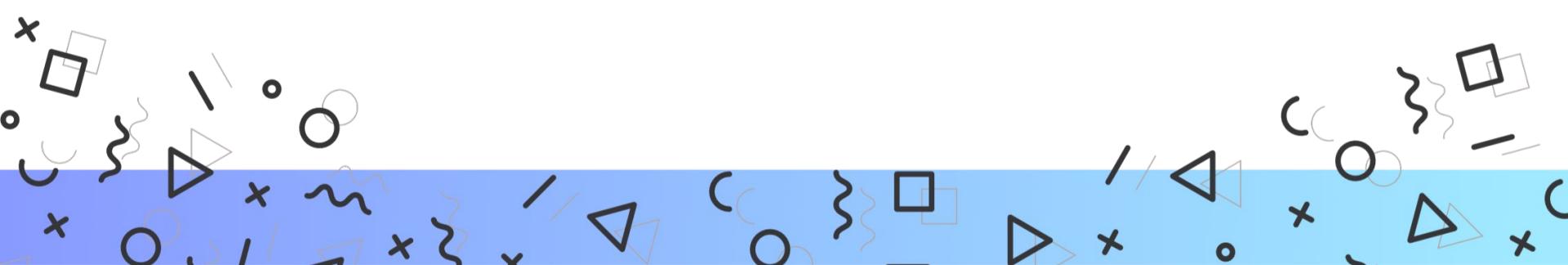
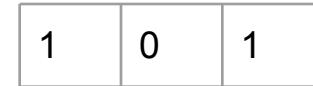
1.5	1.5	1	1.5	1.5	
-----	-----	---	-----	-----	--

# Симметричный (нечетный) фильтр

Входной массив



Фильтр



# Методы добавления рамок

Enumerator

BORDER_CONSTANT	iiiiii abcdefgh iiiiii with some specified i
BORDER_REPLICATE	aaaaaaaa abcdefgh hhhhhhh
BORDER_REFLECT	fedcba abcdefgh hgfedcb
BORDER_WRAP	cdefgh abcdefgh abcdefg
BORDER_REFLECT_101	gfedcb abcdefgh gfedcba
BORDER_TRANSPARENT	uvwxyz absdefgh ijklmno
BORDER_REFLECT101	same as BORDER_REFLECT_101
BORDER_DEFAULT	same as BORDER_REFLECT_101
BORDER_ISOLATED	do not look outside of ROI

[https://docs.opencv.org/3.2.0/d2/de8/group\\_\\_core\\_\\_array.html#ga209f2f4869e304c82d07739337eae7c5](https://docs.opencv.org/3.2.0/d2/de8/group__core__array.html#ga209f2f4869e304c82d07739337eae7c5)

# 2D фильтр

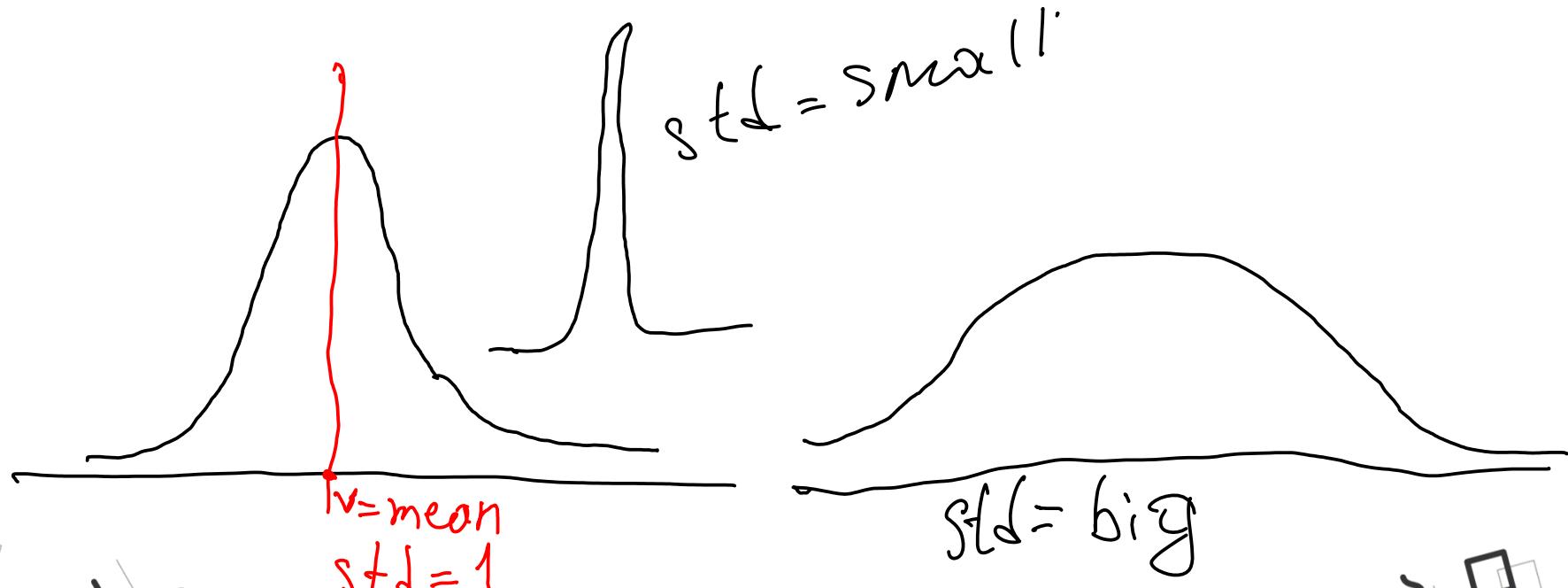
5	6	7	5	3
7	9	8	5	2
8	6	5	7	8
5	2	1	7	9
3	6	7	8	1

1	1	1
1	1	1
1	1	1

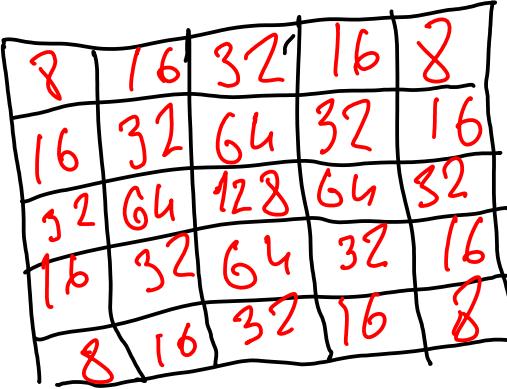
/ 9 =

0.11	0.11	0.11
0.11	0.11	0.11
0.11	0.11	0.11

# Gaussian (normal) distribution



# Gaussian filter



A 5x5 grid of numbers representing a Gaussian filter kernel. The values are:

8	16	32	16	8
16	32	64	32	16
32	64	128	64	32
16	32	64	32	16
8	16	32	16	8

The grid is labeled with a diagonal line and the word "SUM" indicating the values represent a probability distribution.











