ML Session - 2

Jaswanth Reddy K

Coordinated by Dr. G Mamatha, SDIC

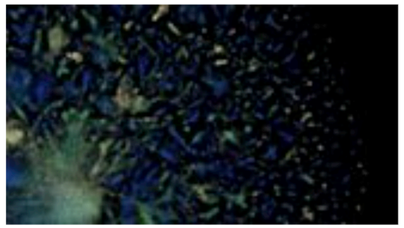
https://webbtelescope.org/contents/media/images/2022/053/01GFRYSFM89AFADVAA0W625BSB

Assignment 1 : Survey

- OpenAI
- Tesla
- Meta - Facebook
- Deepmind
- SpaceX
- Netflix
- Amazon Cloud Services, Google Cloud Platform, Microsoft Azure
- Disney AI/ML Research
- Alexa

Recent ML Advancements

- Machine Learning and Artificial Intelligence are daily headlines.

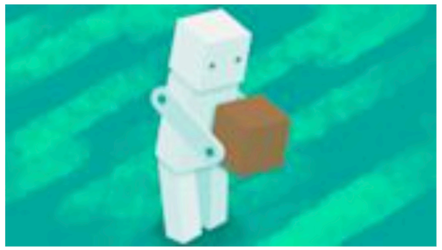


Perceptron: AI that sees with sound, learns to walk and predicts seismic physics

Research in the field of **machine learning** and AI, nc
a key technology in practically every industry...

Perceptron: AI saving whales, steadying gaits and banishing traffic

Research in the field of **machine learning** and AI, now
a key technology in practically every industry...



Will Robots Save The Future Of Work?

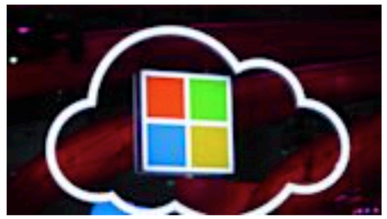
Alastair Bathgate is CEO of robotic process
a
a

B

Salesforce researchers are wor on an AI economist for more equitable tax policy

n an AI
cy policy is surely a complex beast, and depen
on your political leanings, you probably have...

By Ron Miller | April 29, 2020



Microsoft and Nvidia team up to build new Azure-hosted AI supercomputer

Roughly two years ago, Microsoft announced a partnership with OpenAI, the **AI** lab with which it has a...

By Kyle Wiggers | November 16, 2022

Recent ML Advancements :

Text generation

Text Generation demo

using gpt2

Text Generation

Examples

My name is Lewis and I am

Compute ⌘+Enter

0.3

Computation time on cpu: 0.099 s

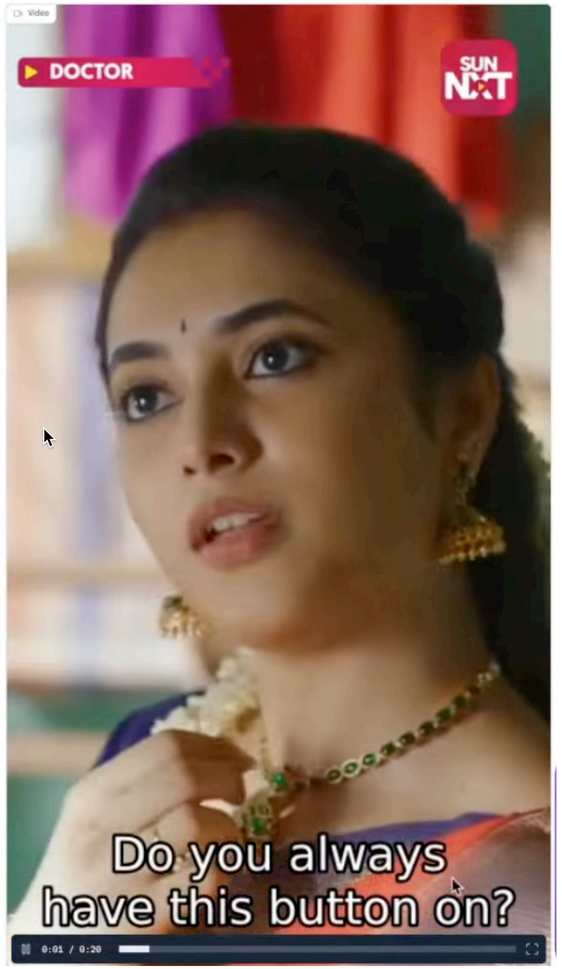
</> JSON Output

Maximize

Image generation

Prompt : John Snow sitting on the iron throne

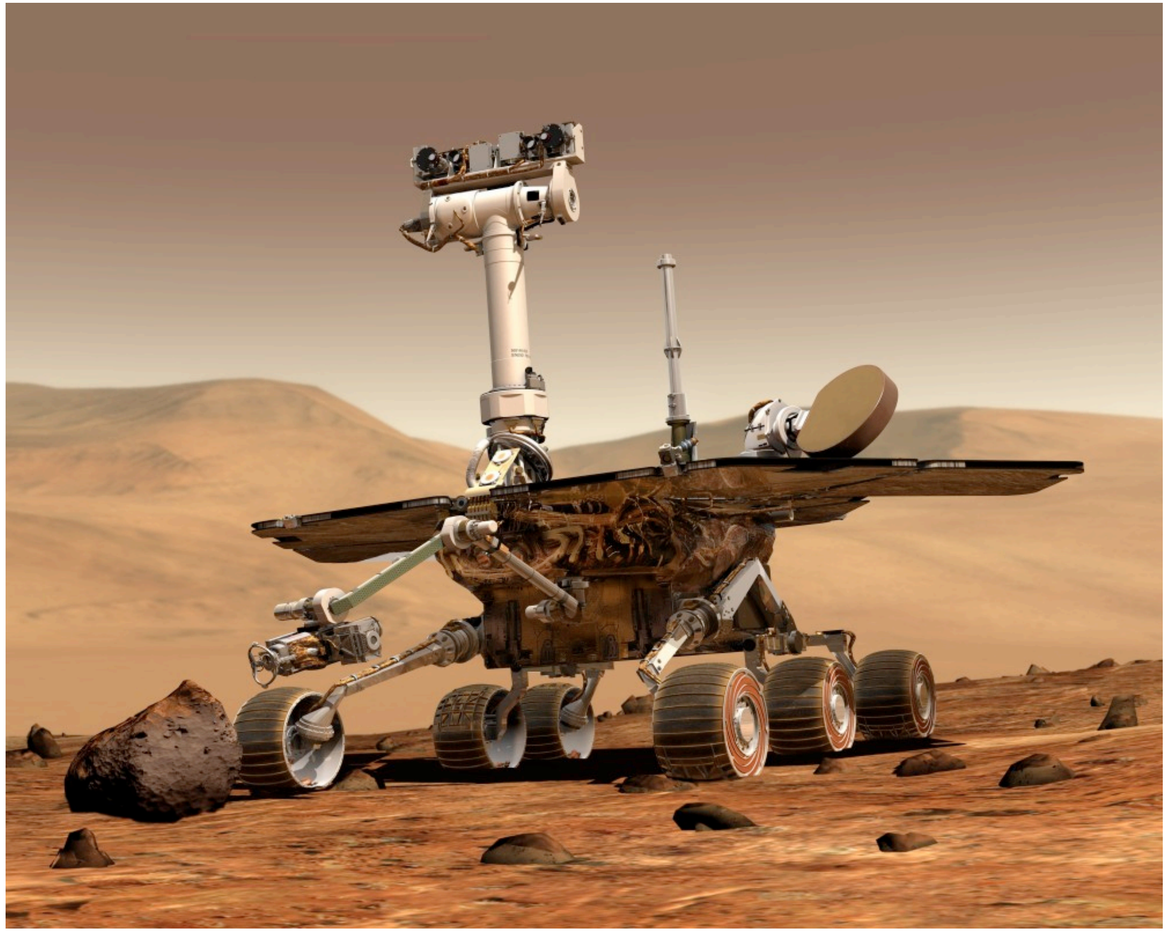
Audio Transcripts



Source : https://github.com/amrrs/subtitle-embedded-video-generator

Why are these important to you?

- All these advancements affect the market
- They affect the jobs available
- This is very similar to industrial revolution
- The academic curriculum is getting outdated
- And, most importantly
- To maintain your curiosity
- And use it in a beneficial direction



Practical

- Scripting language
- Object Oriented Programming
- Python
- Machine Learning libraries
- Deep Learning libraries
- Jupyter Notebooks for EDA
- Kaggle competitions
- Research projects

Theory

- Maths
  - Linear Algebra
  - Optimization Theory
- Probability
  - Random Variables
  - Moments
- Signal Processing
- Machine Learning Algorithms
- Research papers

Signals and Systems

- Signal is a disturbance in a medium (looks like “wave” definition).
- Ex : Images, Audio, Text, Video, Taste, smell
- Systems process signals, and produce new signals.

Signals and Systems

- Signal is a disturbance in a medium (looks like “wave” definition).
- Ex : Images, Audio, Text, Video, Taste, smell
- Systems process signals, and produce new signals.

Signals and Systems

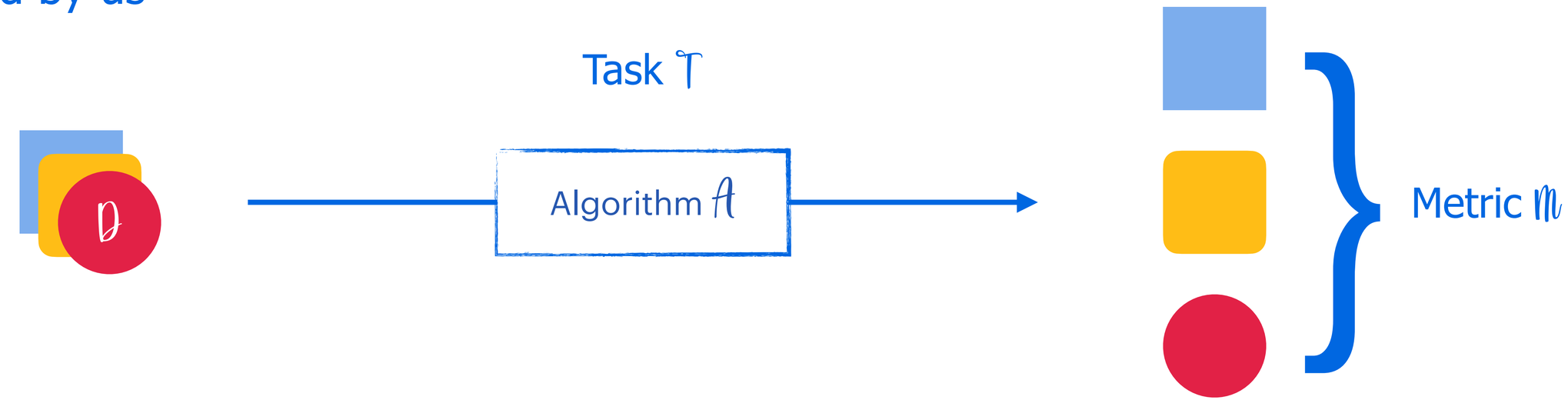
- Signal is a disturbance in a medium (looks like “wave” definition).
- Ex : Images, Audio, Text, Video, Taste, smell
- Systems process signals, and produce new signals.

Signals and Systems

- Signal is a disturbance in a medium (looks like “wave” definition).
- Ex : Images, Audio, Text, Video, Taste, smell
- Systems process signals, and produce new signals.

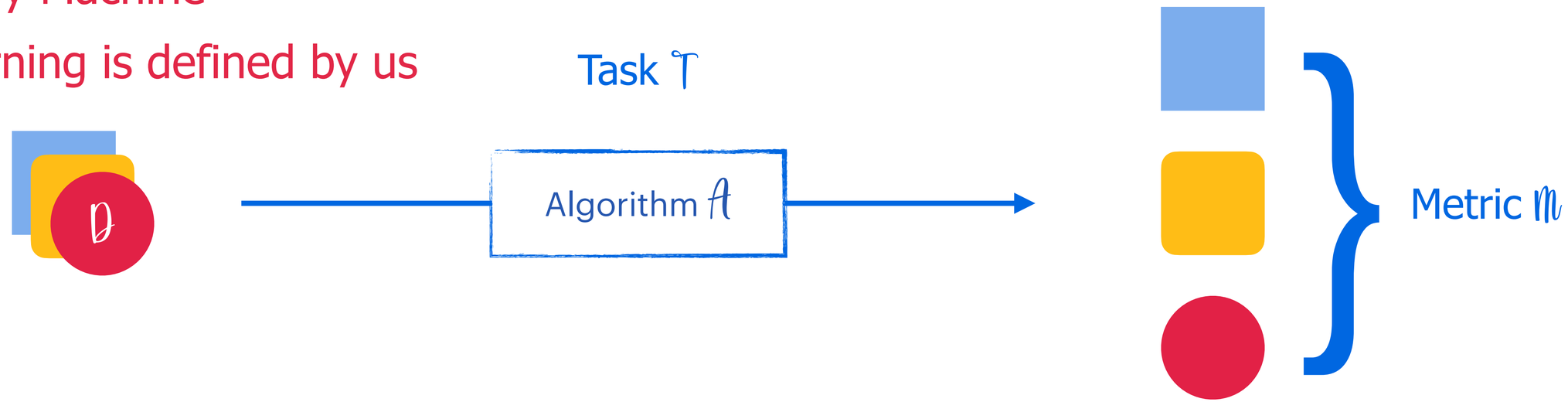
Signal Processing

- Data $\mathcal{D}$
- Task $\mathcal{T}$
- Metric $\mathcal{M}$
- Algorithm $\mathcal{A}$
  - Specified by us



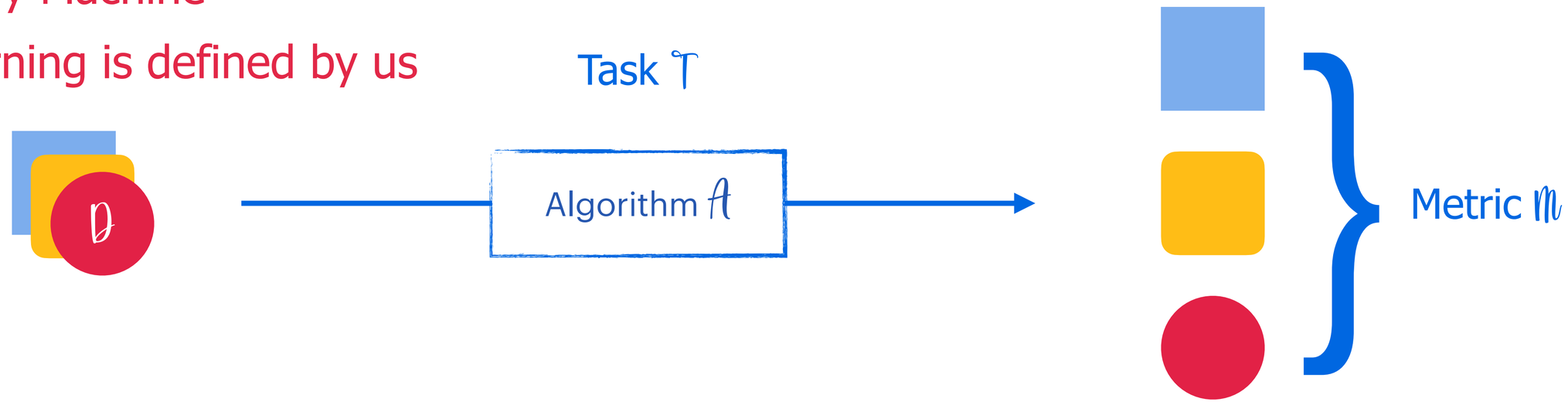
Signal Processing

- Data $\mathcal{D}$
- Task $\mathcal{T}$
- Metric $\mathcal{M}$
- Algorithm $\mathcal{A}$
  - Learnt by Machine
  - The learning is defined by us



Machine Learning

- Data $\mathcal{D}$
- Task $\mathcal{T}$
- Metric $\mathcal{M}$
- Algorithm $\mathcal{A}$
  - Learnt by Machine
  - The learning is defined by us

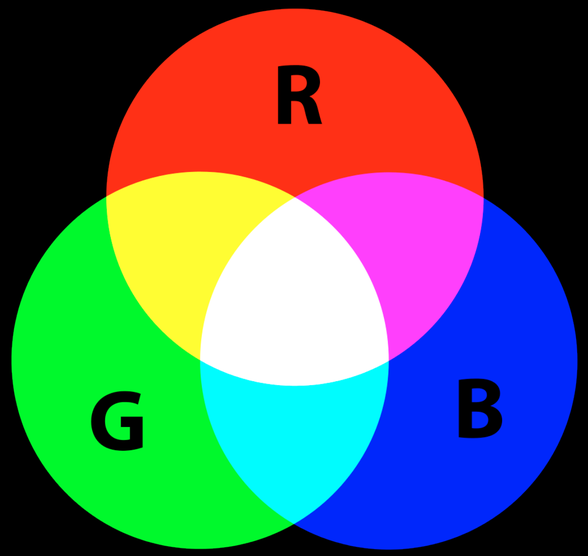


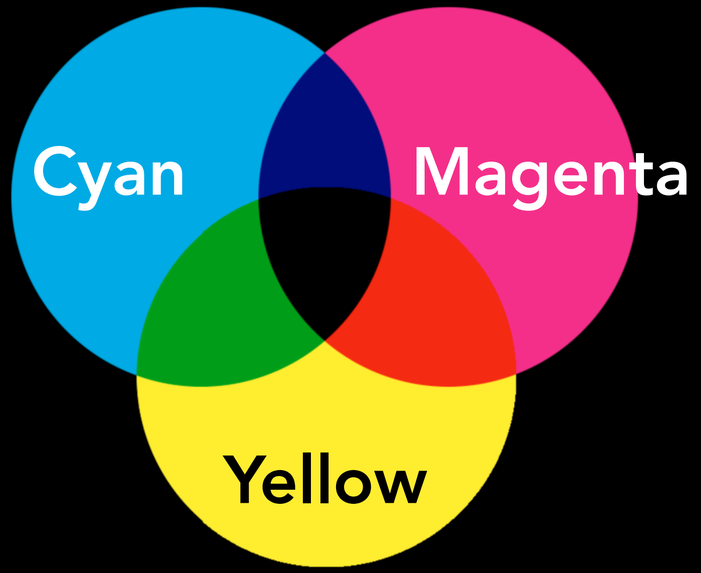
One Image System : Histogram

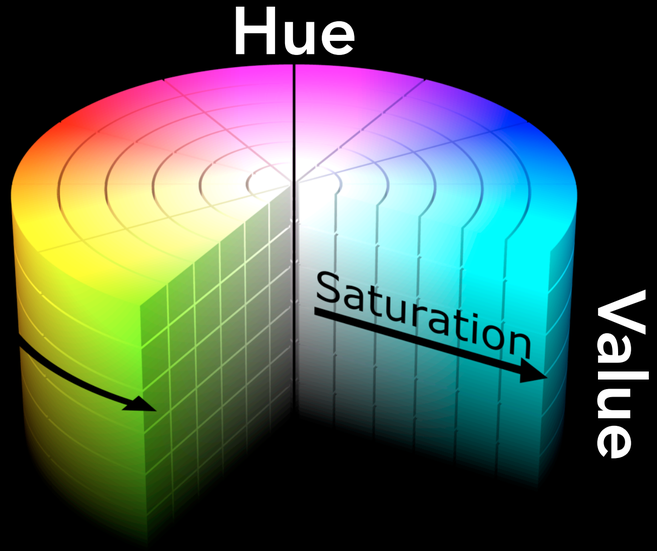
- Histogram gives us information about an image.
- It counts the occurrence of values.
- To understand it properly, we need to first define few things :
  - Image
  - Intensity values
  - occurrence
- Image :
  - Digital Image

Histogram : Digital Image

- Digital Formats
- File Formats : JPEG, PNG, TIFF, PDF, EPS
- Color spaces : RGB, CMYK, HSV





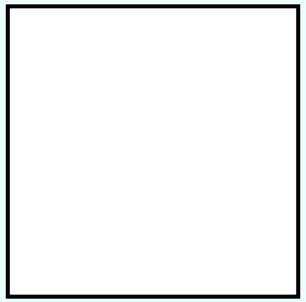


Digital conversion - Images

- Resolution is the scale we are working/measuring at.
- Similarly, for images.
- To what precision do we want to capture the colour information.
- That is called “bit-depth”.
- Number of bits we use to represent each pixel.

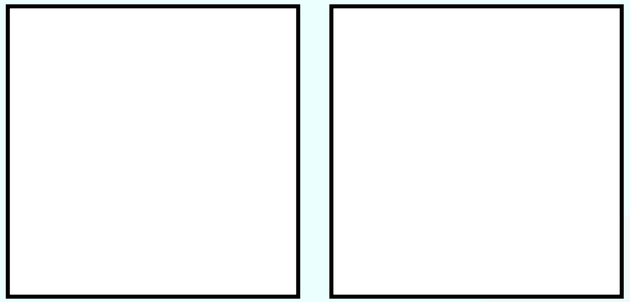
- Therefore while storing/capturing image, we need to think about :
  - The file format : JPG, PNG, TIFF
  - The color space : RGB, CMYK, HSV
  - The bit-depth : 8-bit, 10-bit, 12-bit, 14-bit

Resolution & Bit-depth



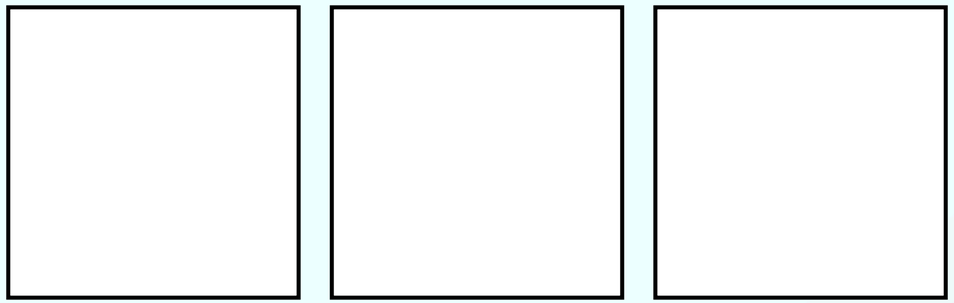
Digit

Resolution & Bit-depth



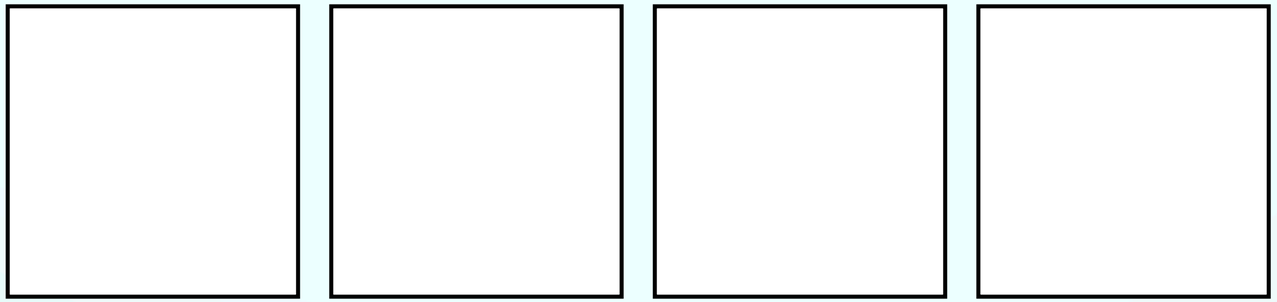
Digit

Resolution & Bit-depth

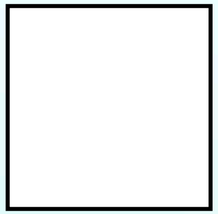


Digit

Resolution & Bit-depth

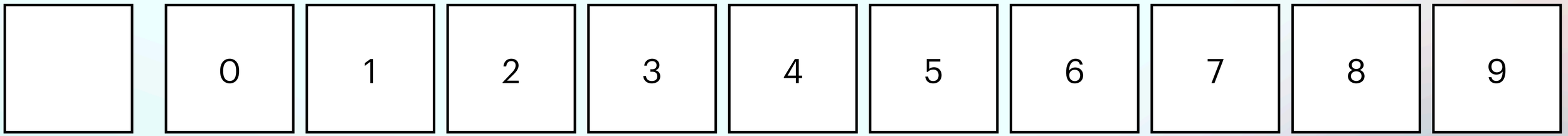


Resolution & Bit-depth



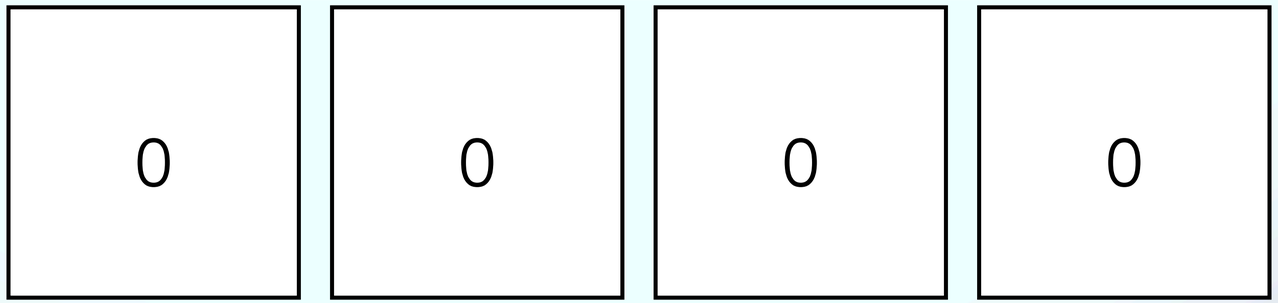
Digit

Resolution & Bit-depth



Resolution & Bit-depth

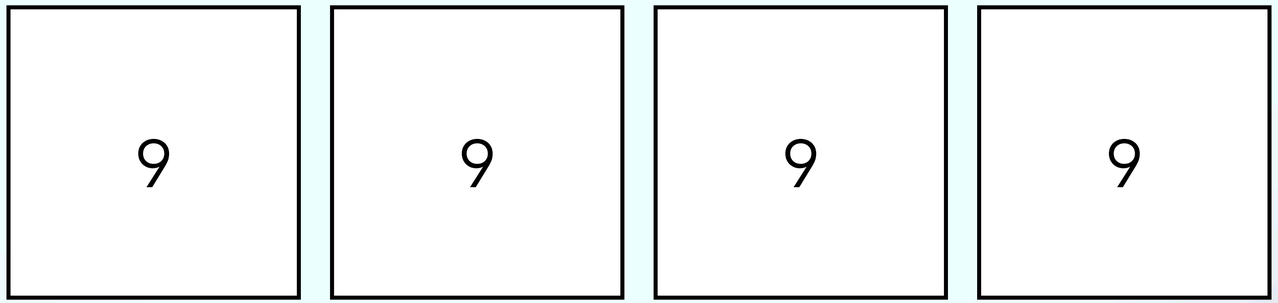
4 - Digit Number



Ranges from 0000 to 9999
that is
10000 values

Resolution & Bit-depth

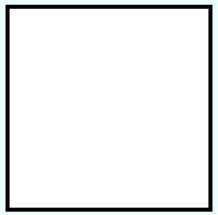
4 - Digit Number



Ranges from 0000 to 9999
that is
10000 values

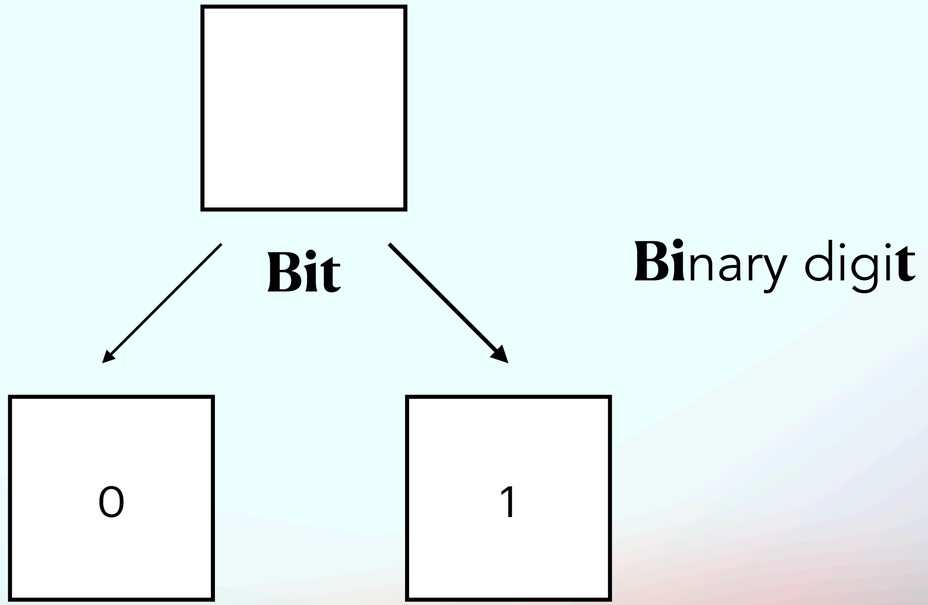
Resolution & Bit-depth

Binary digi**t**



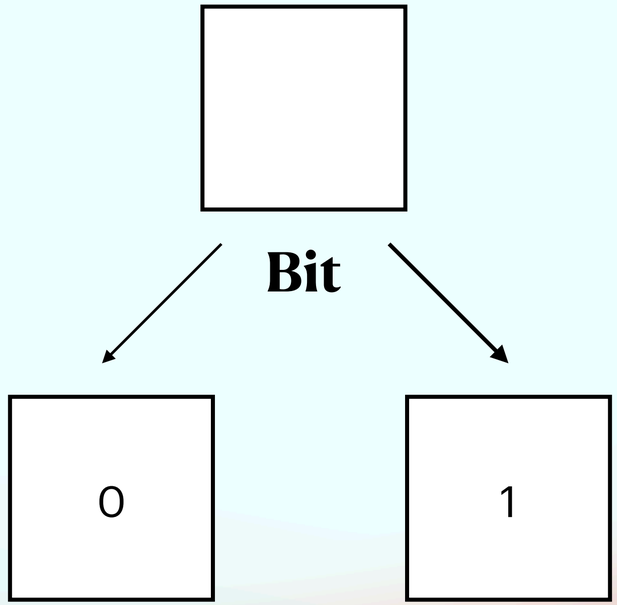
Bit

Resolution & Bit-depth



Resolution & Bit-depth

- Examples :
  - Light - ON/OFF
  - Yes/No
  - True/False
  - Exists/Not



Binary digi**t**

2^1

Resolution & Bit-depth

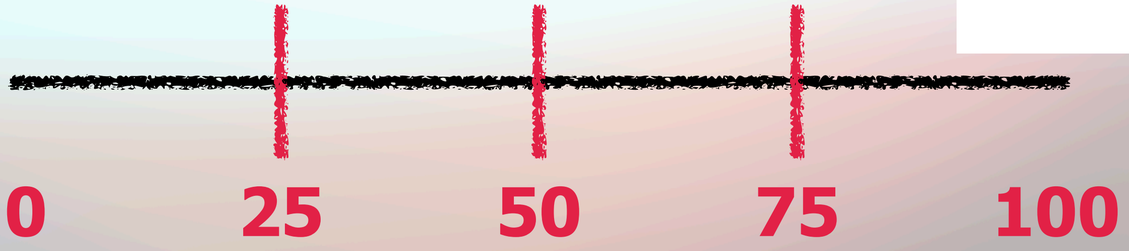
$$4 = 2^2$$

- Examples :
  - Season - Winter, Spring, Summer, Rainy
  - Part of day - Morning, noon, evening, night
  - 0, 25, 50, 75

2 bit number

0	0

Ranges from 0 to 3



Resolution & Bit-depth

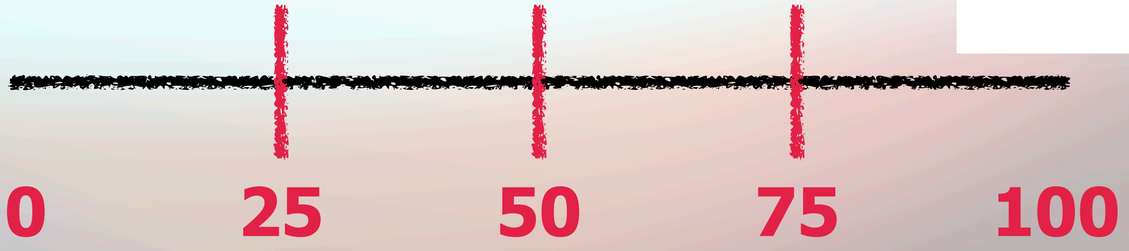
$$4 = 2^2$$

- Examples :
  - Season - Winter, Spring, Summer, Rainy
  - Part of day - Morning, noon, evening, night
  - 0, 25, 50, 75

2 bit number

1	1

Ranges from 0 to 3



Resolution & Bit-depth

$$4 = 2^2$$

- Examples :
  - Season - Winter, Spring, Summer, Rainy
  - Part of day - Morning, noon, evening, night
  - 0, 25, 50, 75

0 0 - 0+

0 1 - 25+

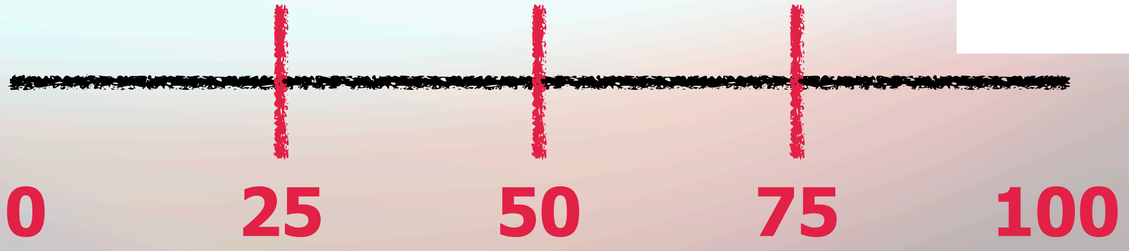
1 0 - 50+

1 1 - 75+

2 bit number

1	1

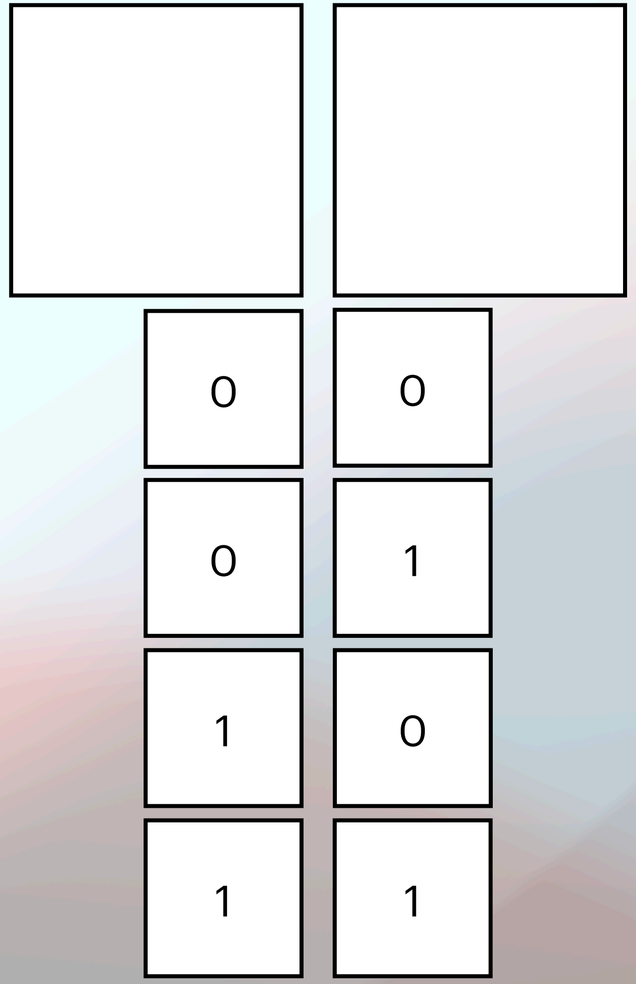
Ranges from 0 to 3



Resolution & Bit-depth

$4 = 2^2$

- Examples :
  - Season - Winter, Spring, Summer, Rainy
  - Part of day - Morning, noon, evening, night
  - 0, 25, 50, 75

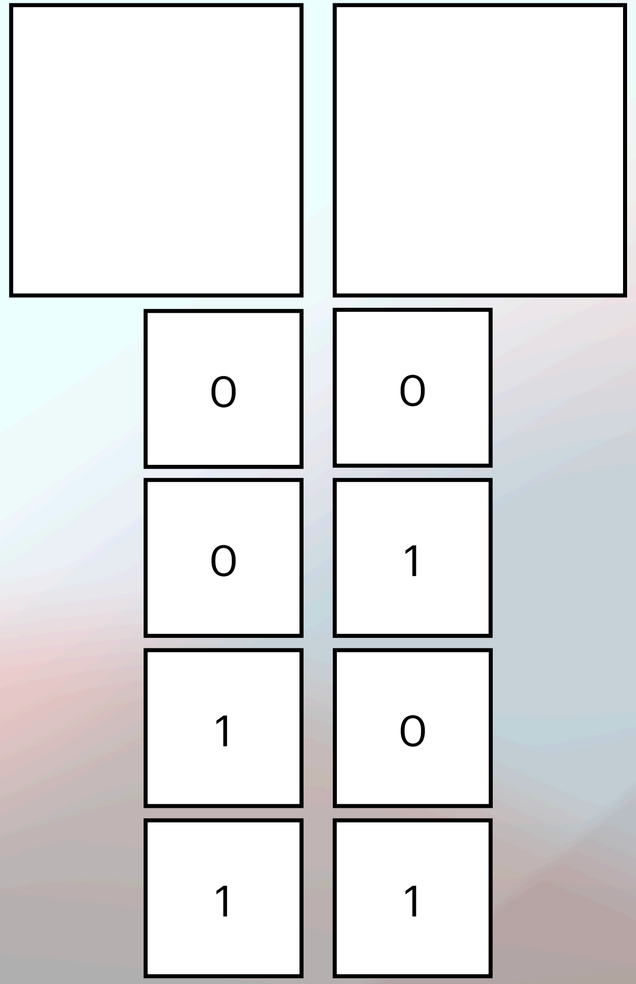


Resolution & Bit-depth

$4 = 2^2$

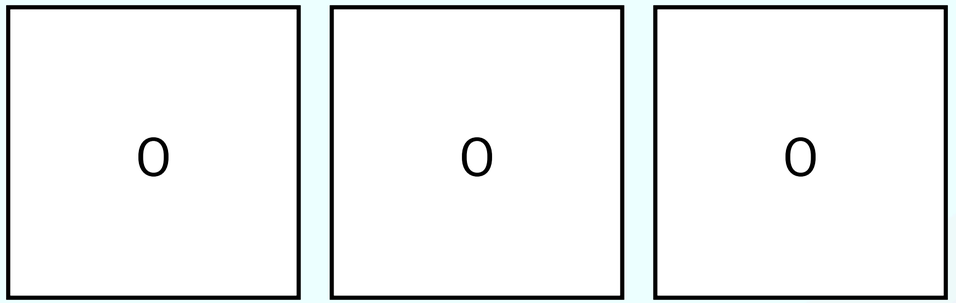
- Examples :
  - Season - Winter, Spring, Summer, Rainy
  - Part of day - Morning, noon, evening, night
  - 0, 25, 50, 75

```
0 0 - 0
0 1 - 1
1 0 - 2
1 1 - 3
```



Resolution & Bit-depth

$8 = 2^3$



Resolution & Bit-depth

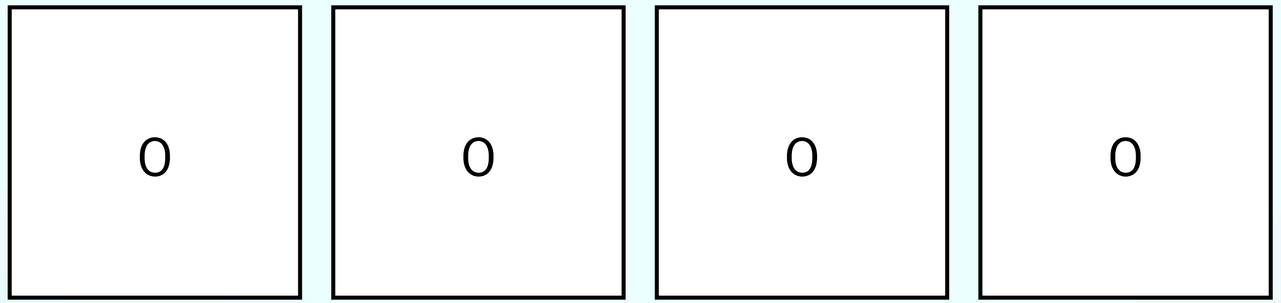
$8 = 2^3$

0	0	0	- 0
0	0	1	- 1
0	1	0	- 2
S	1	1	- 3
1	0	0	- 4
1	0	1	- 5
1	1	0	- 6
1	1	1	- 7

0	0	0	- 0
1	1	1	- 7

Resolution & Bit-depth

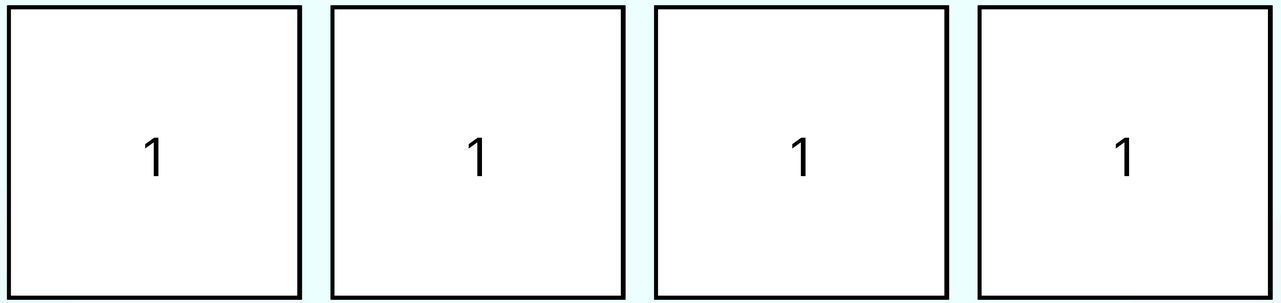
$16 = 2^4$



0

Resolution & Bit-depth

$16 = 2^4$



15

Resolution & Bit-depth

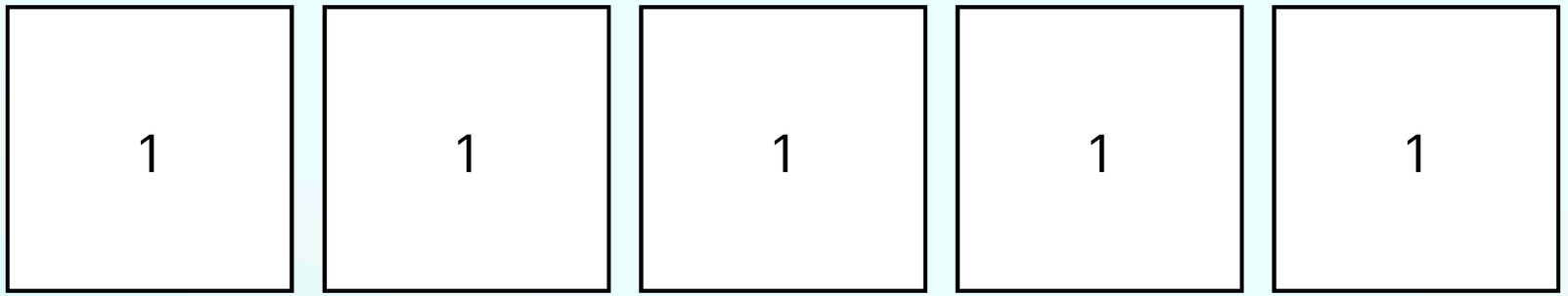
$32 = 2^5$

0	0	0	0	0

0

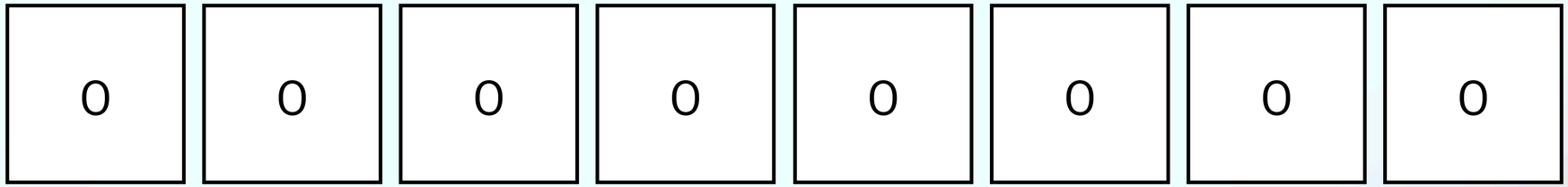
Resolution & Bit-depth

$32 = 2^5$

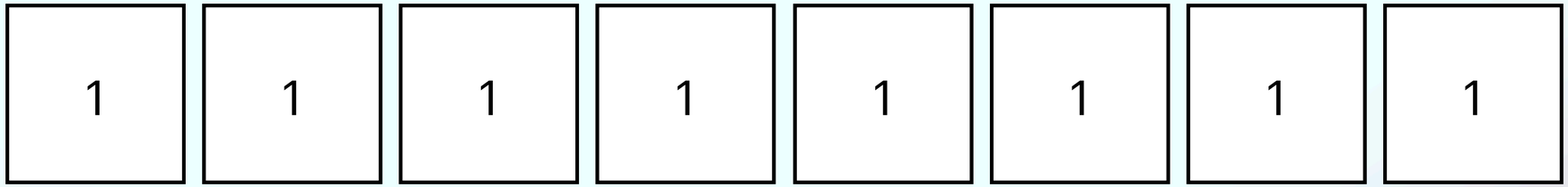


31

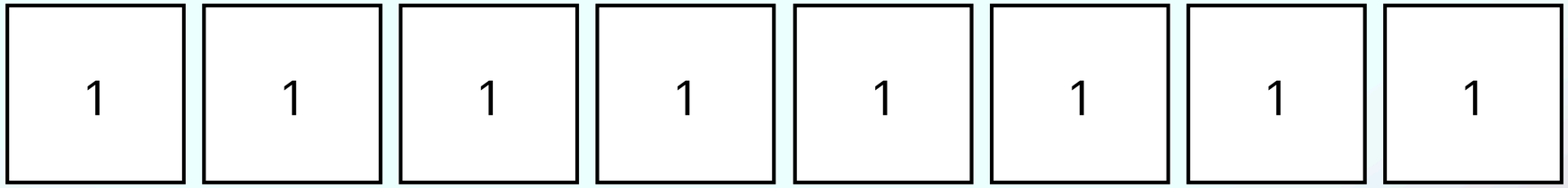
Resolution & Bit-depth



Resolution & Bit-depth

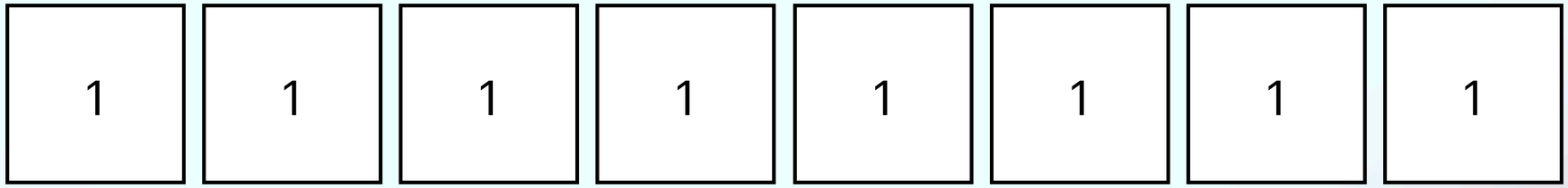


Resolution & Bit-depth



$$256 = 2^8$$

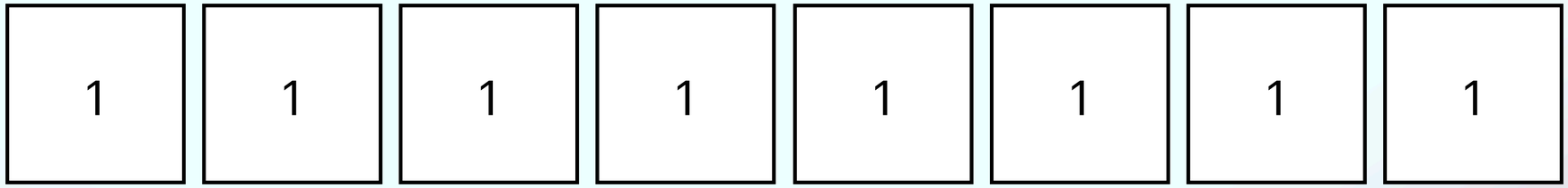
Resolution & Bit-depth



255

$256 = 2^8$

Resolution & Bit-depth



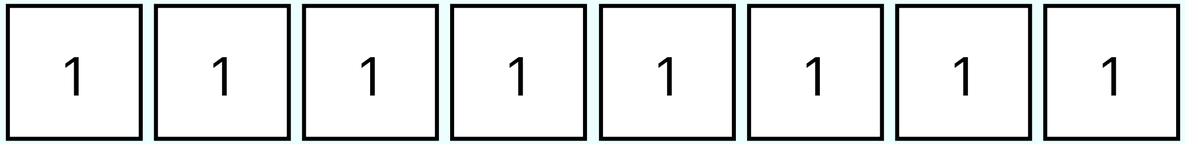
255

0

256 = 2^8

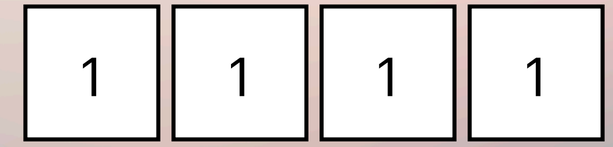
Resolution & Bit-depth

Byte



$256 = 2^8$

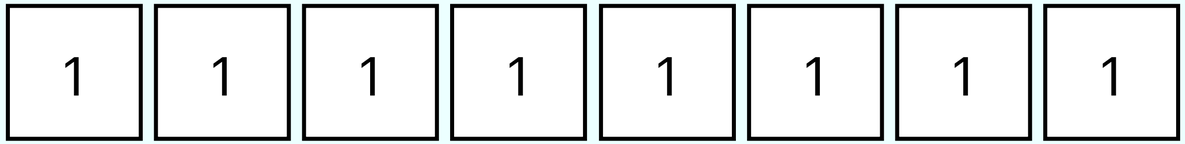
Nibble



$16 = 2^4$

Resolution & Bit-depth

Byte

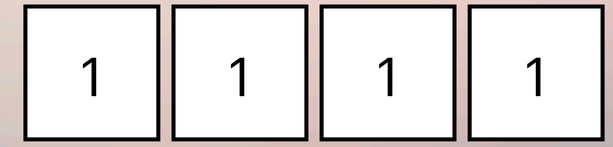


$256 = 2^8$

```
0 0 - 0+
0 1 - 25+
1 0 - 50+
1 1 - 75+
```

2 Bits

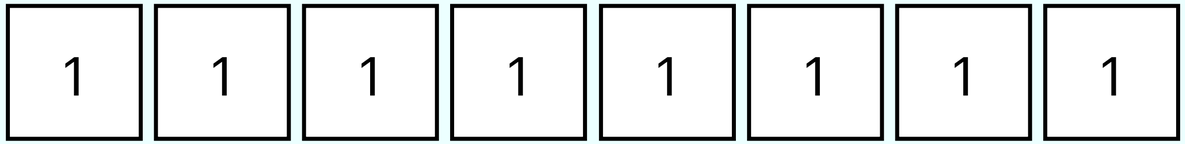
Nibble



$16 = 2^4$

Resolution & Bit-depth

Byte



$256 = 2^8$

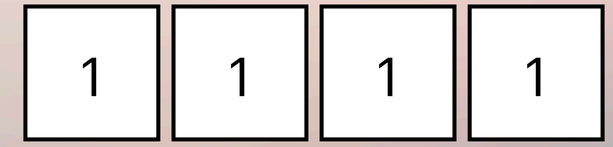
```
0 0 - 0+
0 1 - 25+
1 0 - 50+
1 1 - 75+
```

2 Bits

```
0000 - 0+
0001 - 6.25+
0010 - 12.50+
..
..
1111 - 93.75+
```

4Bits

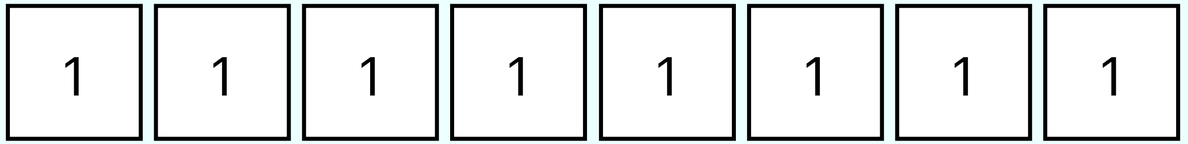
Nibble



$16 = 2^4$

Resolution & Bit-depth

Byte



$256 = 2^8$

```
0 0 - 0+
0 1 - 25+
1 0 - 50+
1 1 - 75+
```

2 Bits

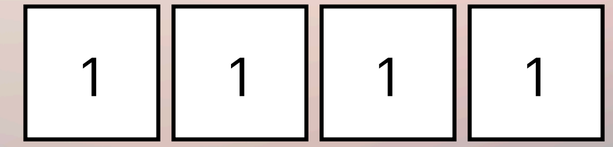
```
0000 - 0+
0001 - 6.25+
0010 - 12.50+
..
..
1111 - 93.75+
```

4Bits

```
00000000 - 0+
00000001 - 0.39+
00000010 - 0.78+
..
..
11111111 - 99.61+
```

8 Bits

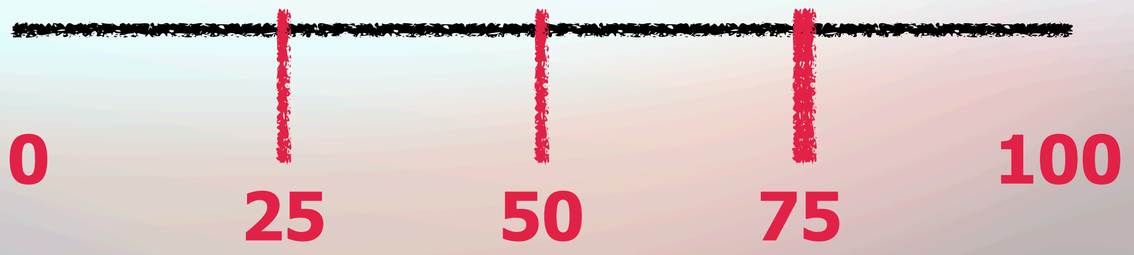
Nibble



$16 = 2^4$

Resolution & Bit-depth

Byte	1	1	1	1	1	1	1	1	$256 = 2^8$
Nibble	1	1	1	1					$16 = 2^4$



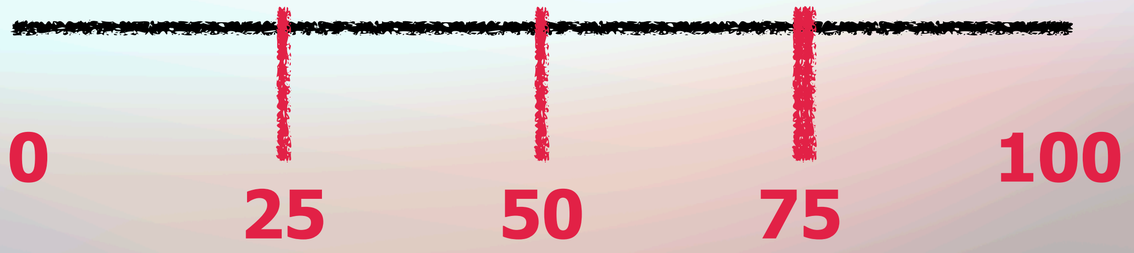
Resolution & Bit-depth

Byte | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | **256 = 2^8**

Nibble | 1 | 1 | 1 | 1 | **16 = 2^4**

```
0 0 - 0+
0 1 - 25+
1 0 - 50+
1 1 - 75+
```

2 Bits



Resolution & Bit-depth

Byte 1 1 1 1 1 1 1 1 — **256 = 2^8**

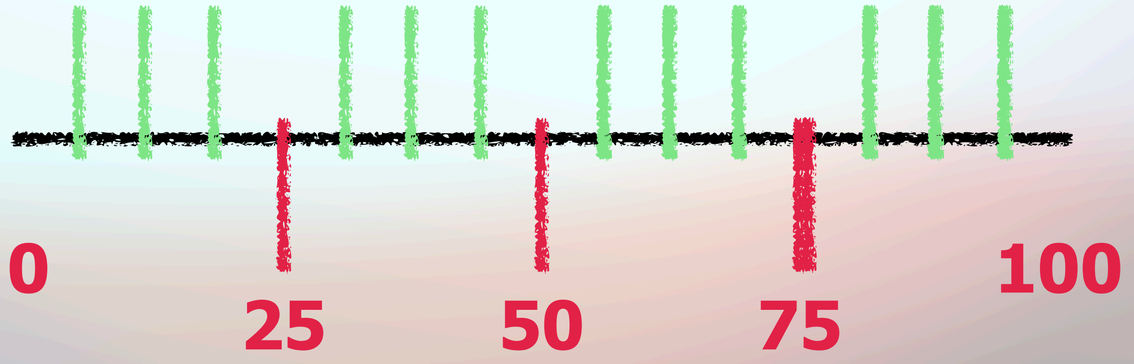
Nibble 1 1 1 1 — **16 = 2^4**

```
0 0 - 0+
0 1 - 25+
1 0 - 50+
1 1 - 75+
```

2 Bits

```
0000 - 0+
0001 - 6.25+
0010 - 12.50+
..
..
1111 - 93.75+
```

4Bits



Resolution & Bit-depth

Byte 1 1 1 1 1 1 1 1 — **256 = 2^8**

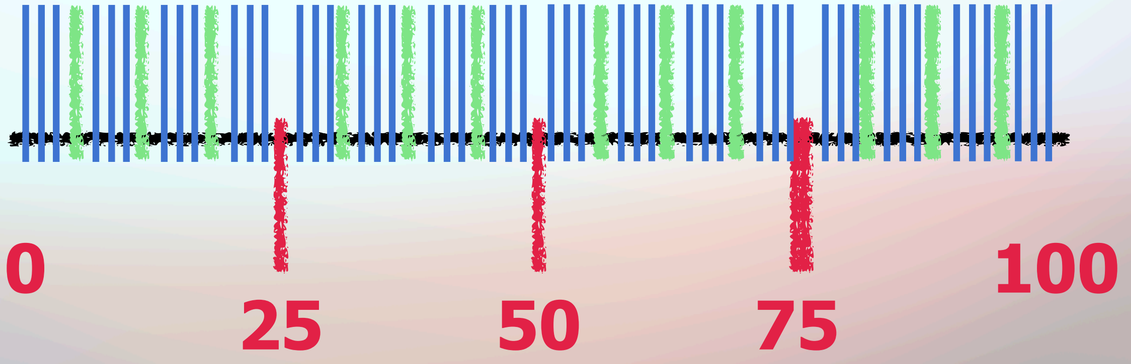
Nibble 1 1 1 1 — **16 = 2^4**

```
0 0 - 0+
0 1 - 25+
1 0 - 50+
1 1 - 75+
```

2 Bits

```
0000 - 0+
0001 - 6.25+
0010 - 12.50+
..
..
1111 - 93.75+
```

4Bits

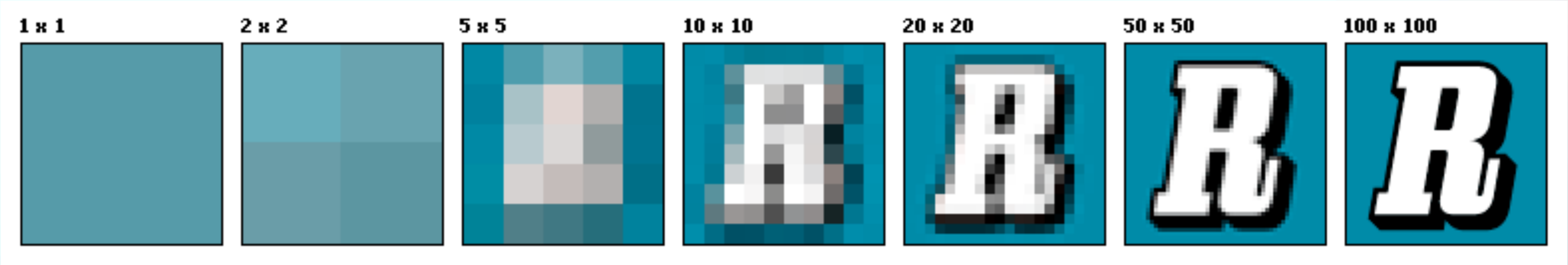



```
00000000 - 0+
00000001 - 0.39+
00000010 - 0.78+
..
..
11111111 - 99.61+
```

8 Bits

Resolution & Bit-depth

Number of pixels



Histogram : Images and Matrices

Histogram : Images and Matrices

- What is an image?
  - Visual representation of anything

Histogram : Images and Matrices

- What is an image?
  - Visual representation of anything
- If RGB, each image is a 3D matrix.
  - Three 2D matrices - Red, Green, Blue

Histogram : Images and Matrices

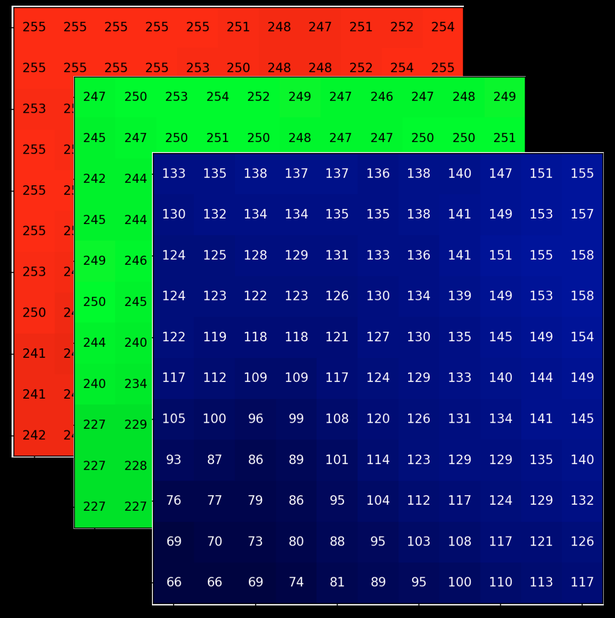
- What is an image?
  - Visual representation of anything
- If RGB, each image is a 3D matrix.
  - Three 2D matrices - Red, Green, Blue

Histogram : Images and Matrices

- What is an image?
  - Visual representation of anything
- If RGB, each image is a 3D matrix.
  - Three 2D matrices - Red, Green, Blue
  - Each image can be represented as 3D matrix.

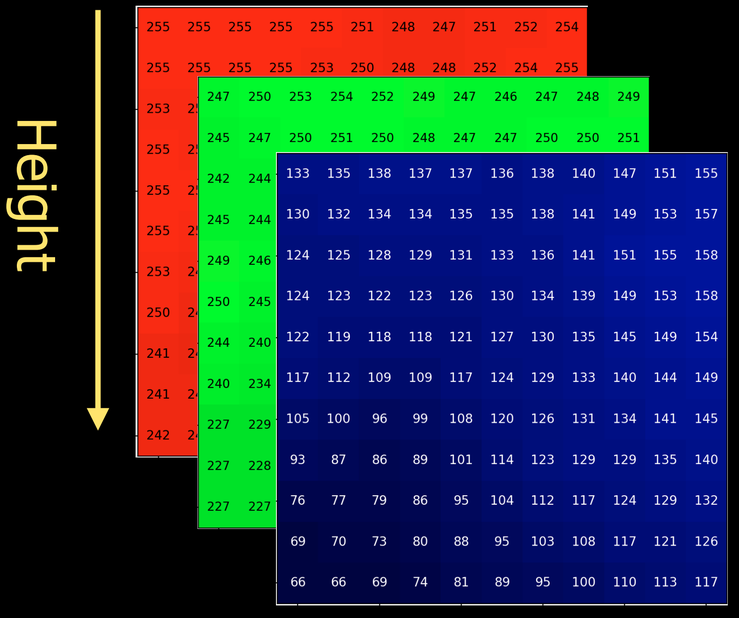
Histogram : Images and Matrices

- What is an image?
  - Visual representation of anything
- If RGB, each image is a 3D matrix.
  - Three 2D matrices - Red, Green, Blue
  - Each image can be represented as 3D matrix.



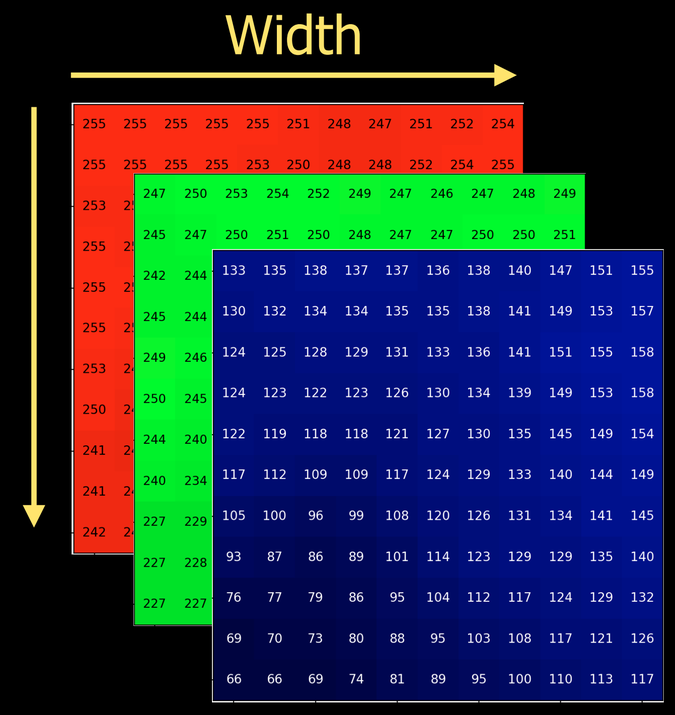
Histogram : Images and Matrices

- What is an image?
  - Visual representation of anything
- If RGB, each image is a 3D matrix.
  - Three 2D matrices - Red, Green, Blue
  - Each image can be represented as 3D matrix.



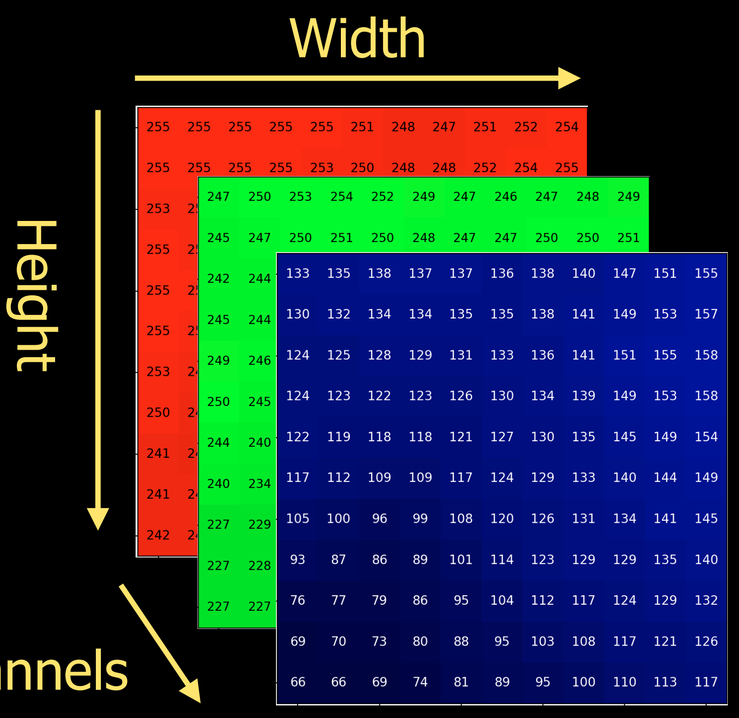
Histogram : Images and Matrices

- What is an image?
  - Visual representation of anything
- If RGB, each image is a 3D matrix.
  - Three 2D matrices - Red, Green, Blue
  - Each image can be represented as 3D matrix.



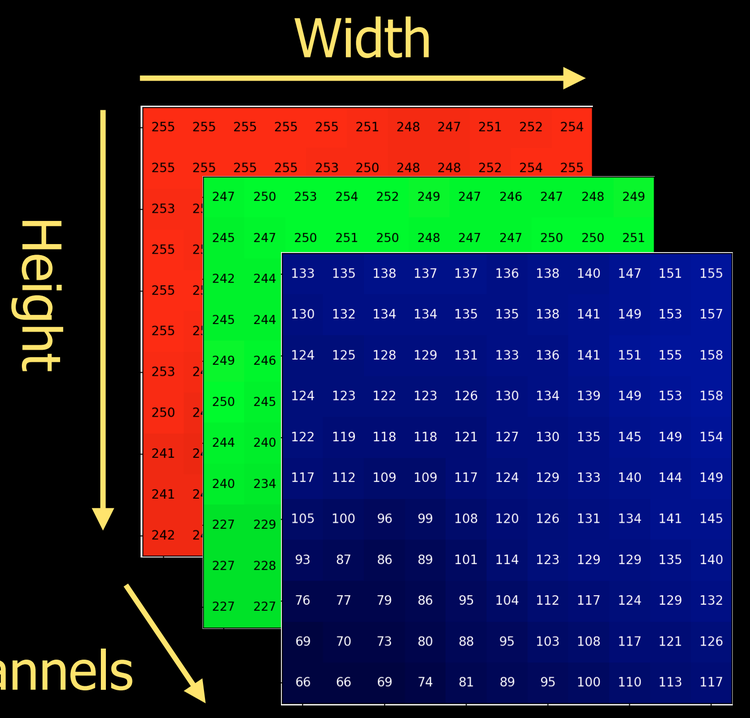
Histogram : Images and Matrices

- What is an image?
  - Visual representation of anything
- If RGB, each image is a 3D matrix.
  - Three 2D matrices - Red, Green, Blue
  - Each image can be represented as 3D matrix.



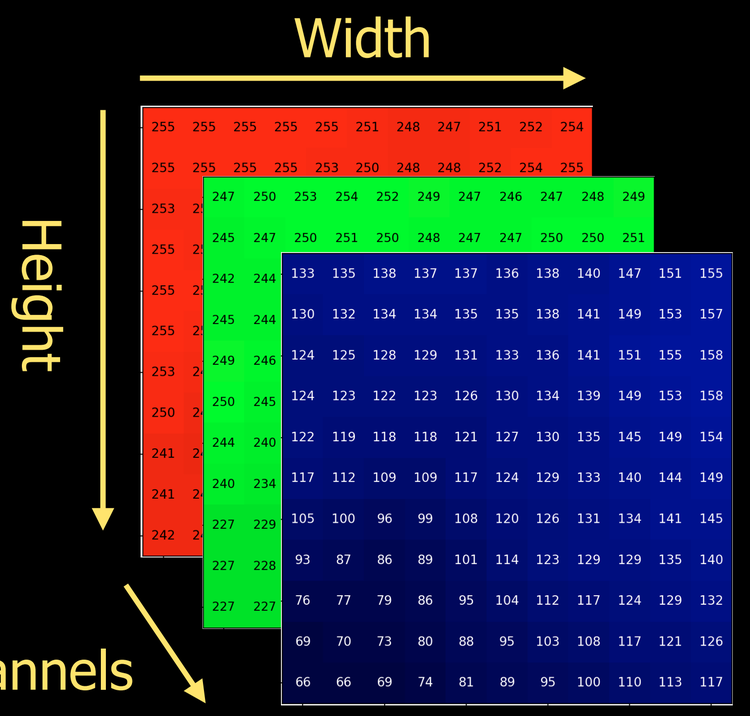
Histogram : Images and Matrices

- What is an image?
  - Visual representation of anything
- If RGB, each image is a 3D matrix.
  - Three 2D matrices - Red, Green, Blue
  - Each image can be represented as 3D matrix.
  - [H, W, C] - Ex : (1920,1080,3)



Histogram : Images and Matrices

- What is an image?
  - Visual representation of anything
- If RGB, each image is a 3D matrix.
  - Three 2D matrices - Red, Green, Blue
  - Each image can be represented as 3D matrix.
  - [H, W, C] - Ex : (1920,1080,3)
- Image processing ≈ Matrix Transformations

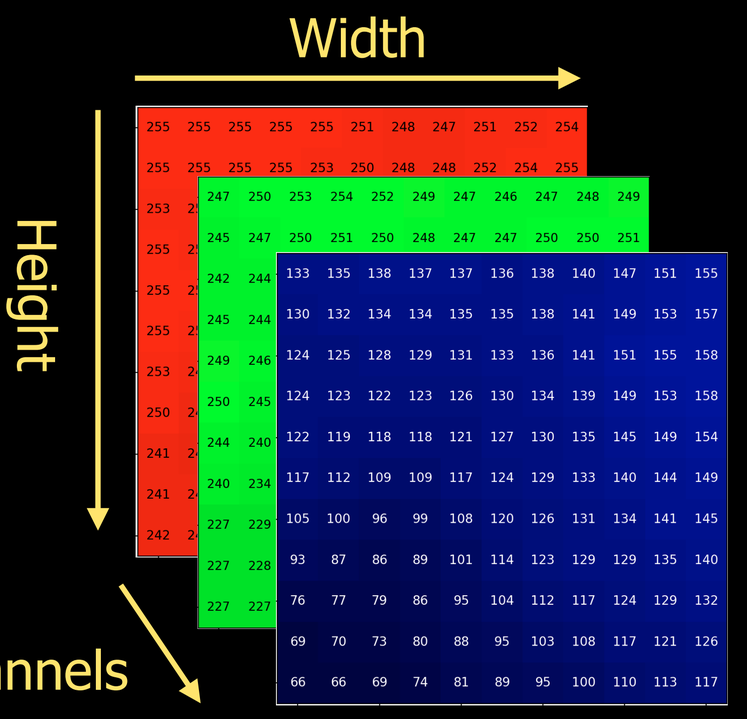


Histogram : Images and Matrices

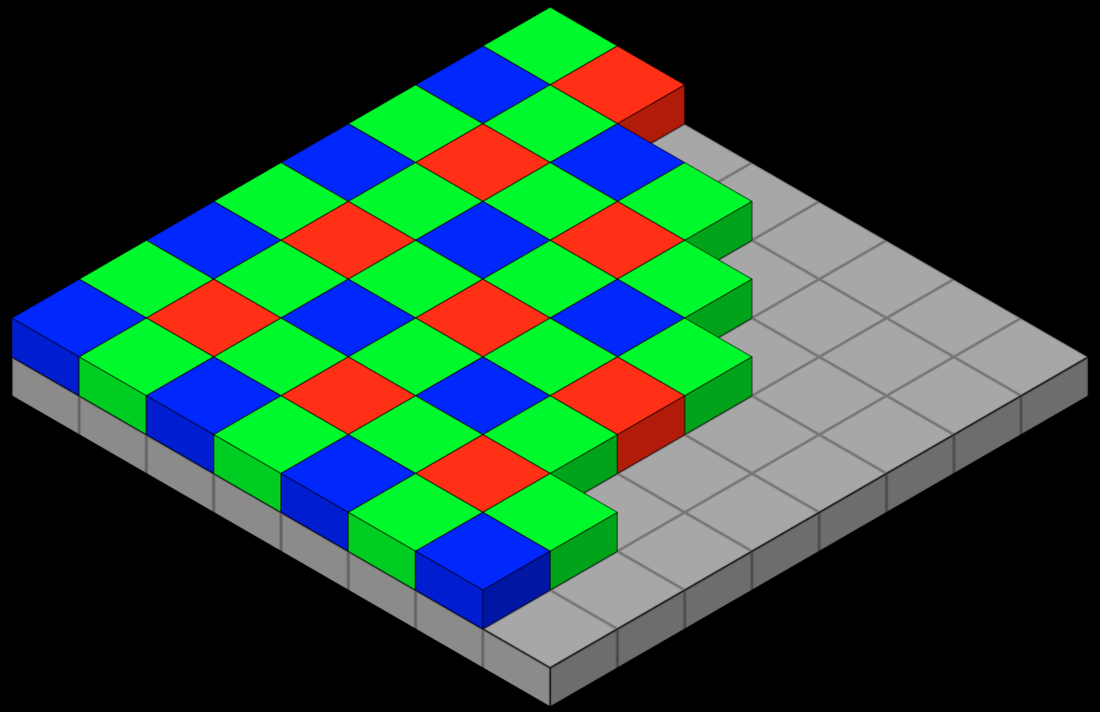
- What is an image?
  - Visual representation of anything
- If RGB, each image is a 3D matrix.
  - Three 2D matrices - Red, Green, Blue
  - Each image can be represented as 3D matrix.
  - [H, W, C] - Ex : (1920,1080,3)
- Image processing ≈ Matrix Transformations

Therefore, to process any image/signal, we need :

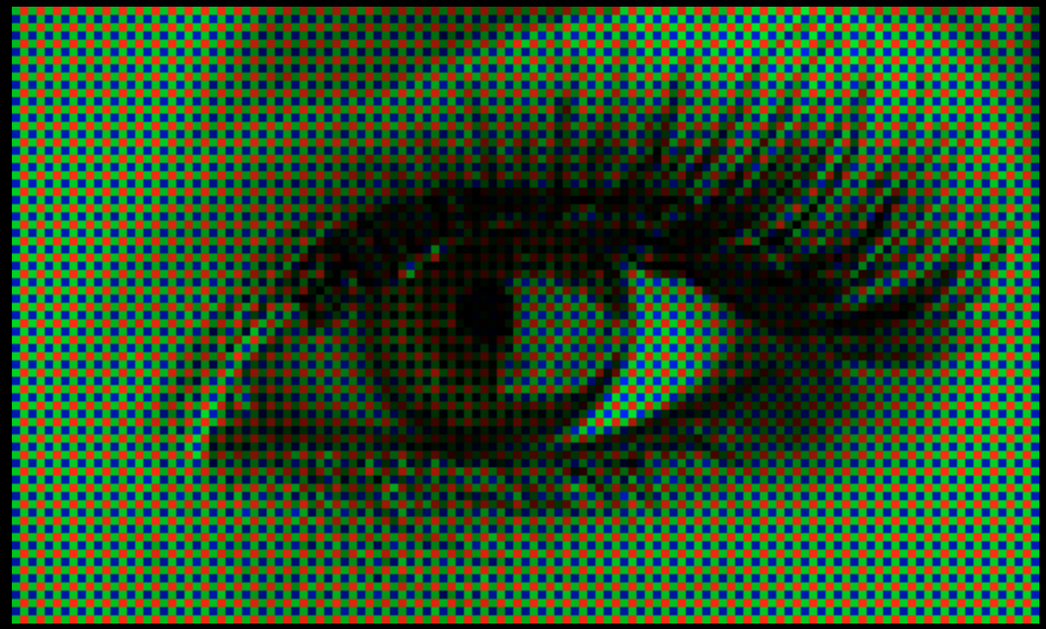
- Matrix operations - Theory
- Computer programming - Practical



Histogram : Digital Image Sensor

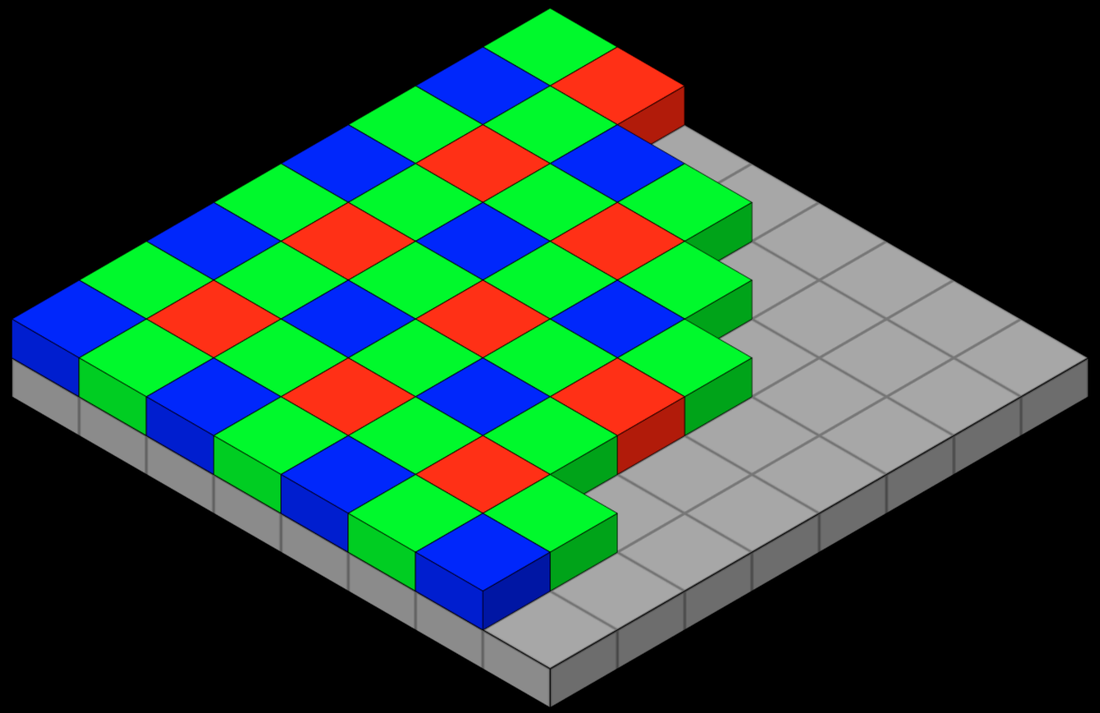


Bayer Sensor

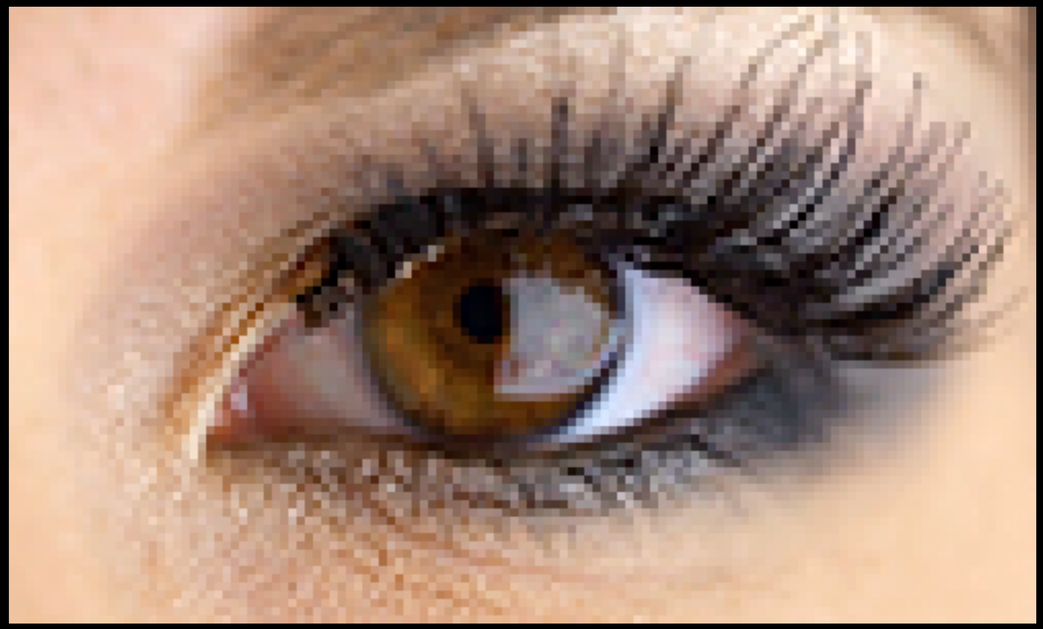


Source : https://www.red.com/red-101/bayer-sensor-strategy

Histogram : Digital Image Sensor



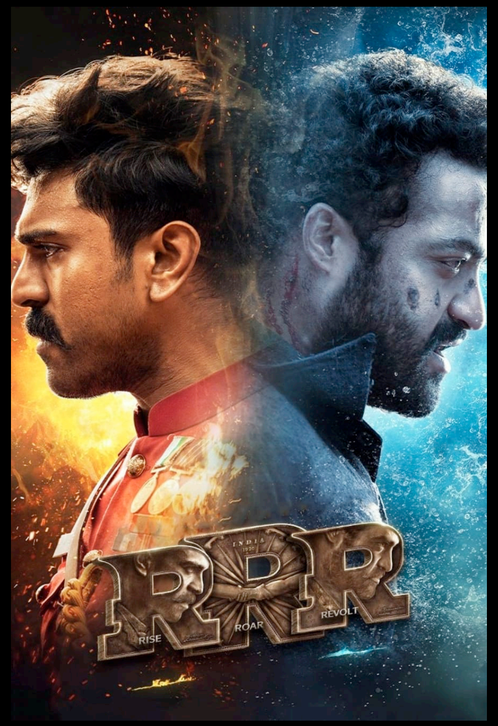
Bayer Sensor



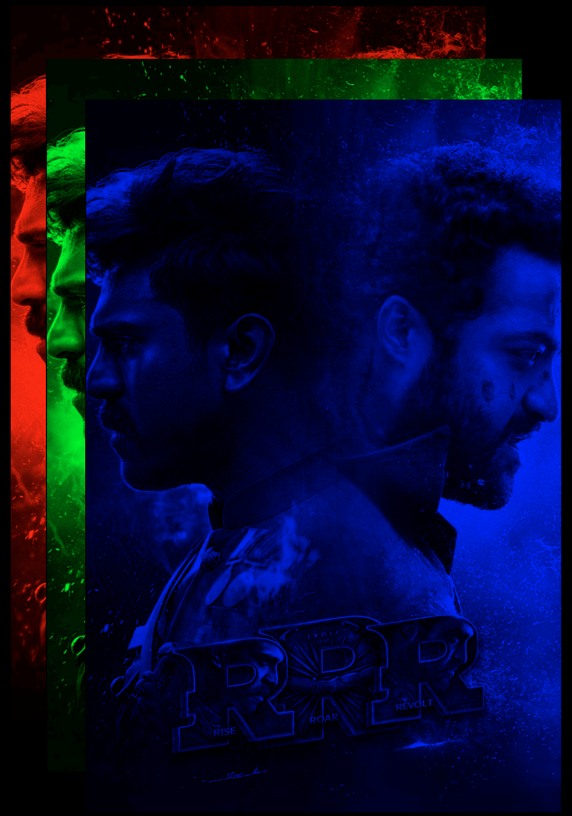
Source : https://www.red.com/red-101/bayer-sensor-strategy

Histogram : Grayscale Images

- Image with single Channel



=

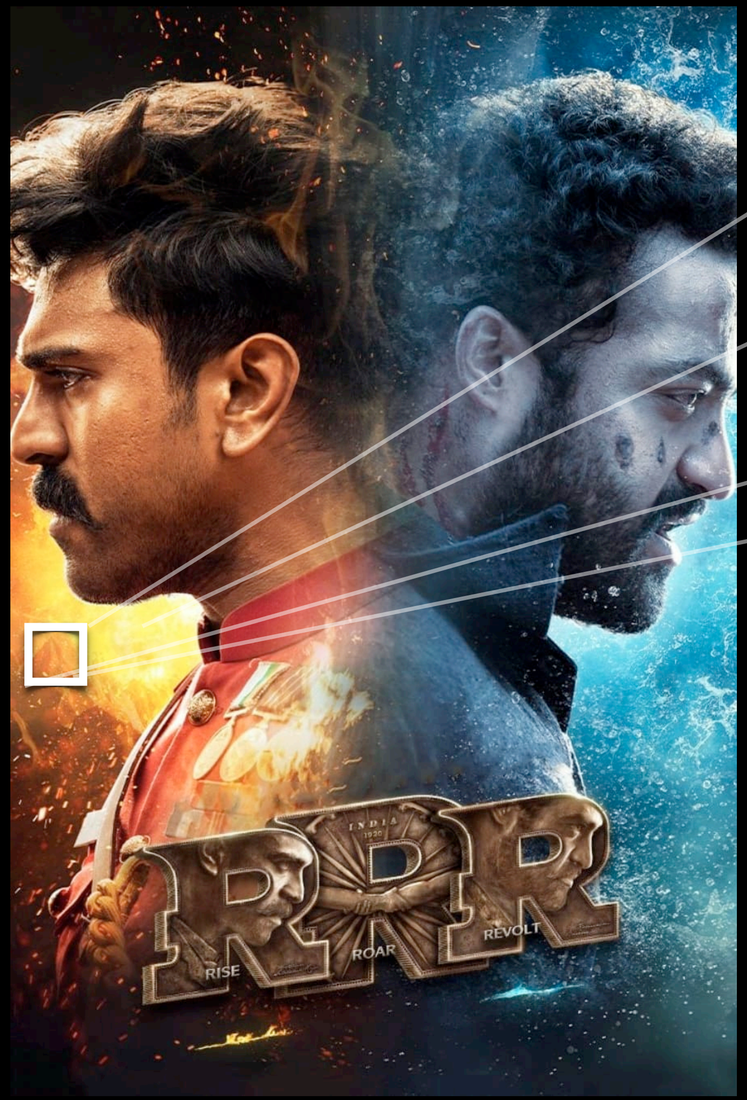


[892, 1338, 3]

3 channels

1 channel = mean of 3 channels

Histogram : Digital Image

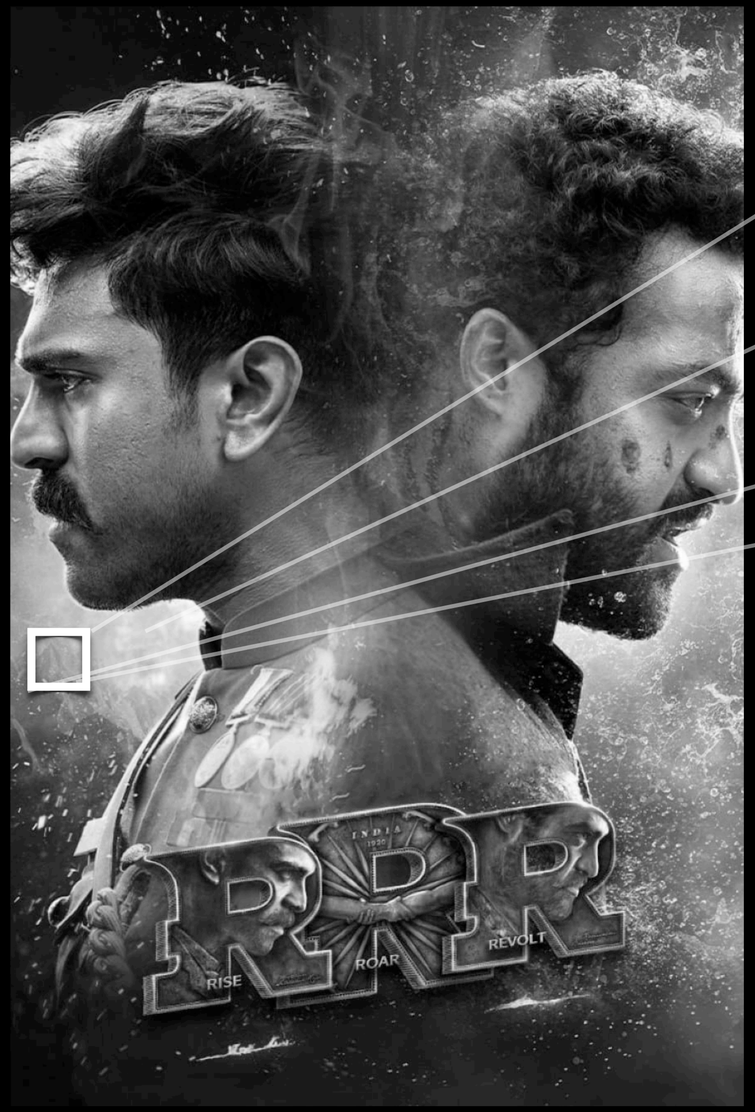


255	255	255	255	255	251	248	247	251	252	254
255	255	255	255	253	250	248	248	252	254	255
253	253	252	250	250	249	248	248	254	255	255
255	253	250	248	248	248	246	246	254	255	255
255	254	249	246	247	248	246	245	250	252	254
255	252	246	243	246	249	250	249	250	251	253
253	247	242	240	245	251	253	253	251	252	254
250	244	238	240	246	253	255	255	252	255	255
241	240	241	243	248	251	254	254	255	255	255
241	242	240	244	247	250	252	252	254	255	255
242	242	241	243	245	248	249	249	251	252	252

247	250	253	254	252	249	247	246	247	248	249
245	247	250	251	250	248	247	247	250	250	251
242	244	245	248	249	250	248	247	252	252	252
245	244	244	246	248	249	249	248	252	253	253
249	246	244	245	248	250	250	248	252	250	251
250	245	242	243	247	251	252	250	249	250	249
244	240	236	239	246	253	254	253	250	250	249
240	234	232	235	245	254	255	255	250	249	250
227	229	232	238	244	249	251	250	250	249	248
227	228	230	235	242	246	247	246	247	245	244
227	227	229	233	239	243	243	242	243	241	241

133	135	138	137	137	136	138	140	147	151	155
130	132	134	134	135	135	138	141	149	153	157
124	125	128	129	131	133	136	141	151	155	158
124	123	122	123	126	130	134	139	149	153	158
122	119	118	118	121	127	130	135	145	149	154
117	112	109	109	117	124	129	133	140	144	149
105	100	96	99	108	120	126	131	134	141	145
93	87	86	89	101	114	123	129	129	135	140
76	77	79	86	95	104	112	117	124	129	132
69	70	73	80	88	95	103	108	117	121	126
66	66	69	74	81	89	95	100	110	113	117

Histogram : Digital Image



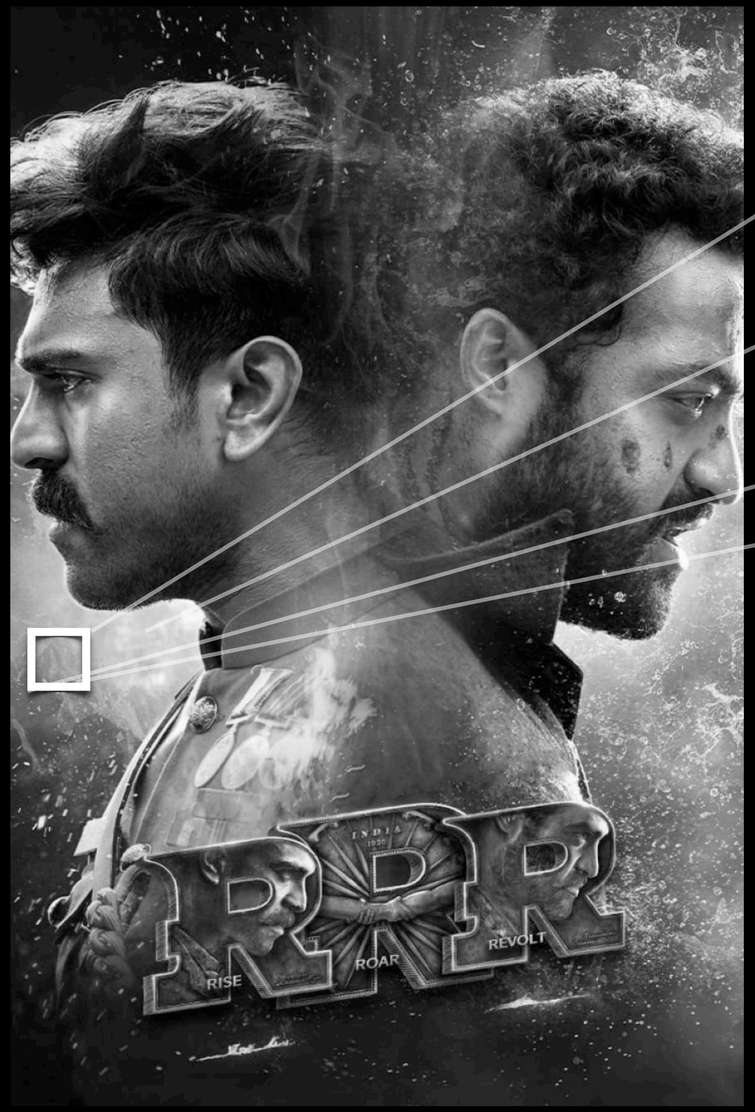
255	255	255	255	255	251	248	247	251	252	254
255	255	255	255	253	250	248	248	252	254	255
253	253	252	250	250	249	248	248	254	255	255
255	253	250	248	248	248	246	246	254	255	255
255	254	249	246	247	248	246	245	250	252	254
255	252	246	243	246	249	250	249	250	251	253
253	247	242	240	245	251	253	253	251	252	254
250	244	238	240	246	253	255	255	252	255	255
241	240	241	243	248	251	254	254	255	255	255
241	242	240	244	247	250	252	252	254	255	255
242	242	241	243	245	248	249	249	251	252	252

247	250	253	254	252	249	247	246	247	248	249
245	247	250	251	250	248	247	247	250	250	251
242	244	245	248	249	250	248	247	252	252	252
245	244	244	246	248	249	249	248	252	253	253
249	246	244	245	248	250	250	248	252	250	251
250	245	242	243	247	251	252	250	249	250	249
244	240	236	239	246	253	254	253	250	250	249
240	234	232	235	245	254	255	255	250	249	250
227	229	232	238	244	249	251	250	250	249	248
227	228	230	235	242	246	247	246	247	245	244
227	227	229	233	239	243	243	242	243	241	241

133	135	138	137	137	136	138	140	147	151	155
130	132	134	134	135	135	138	141	149	153	157
124	125	128	129	131	133	136	141	151	155	158
124	123	122	123	126	130	134	139	149	153	158
122	119	118	118	121	127	130	135	145	149	154
117	112	109	109	117	124	129	133	140	144	149
105	100	96	99	108	120	126	131	134	141	145
93	87	86	89	101	114	123	129	129	135	140
76	77	79	86	95	104	112	117	124	129	132
69	70	73	80	88	95	103	108	117	121	126
66	66	69	74	81	89	95	100	110	113	117

211.0	213.0	215.0	215.0	214.0	212.0	211.0	211.0	215.0	217.0	219.0
210.0	211.0	213.0	213.0	212.0	211.0	211.0	212.0	217.0	219.0	221.0
206.0	207.0	208.0	209.0	210.0	210.0	210.0	212.0	219.0	220.0	221.0
208.0	206.0	205.0	205.0	207.0	209.0	209.0	211.0	218.0	220.0	222.0
208.0	206.0	203.0	203.0	205.0	208.0	208.0	209.0	215.0	217.0	219.0
207.0	203.0	199.0	198.0	203.0	208.0	210.0	210.0	213.0	215.0	217.0
200.0	195.0	191.0	192.0	199.0	208.0	211.0	212.0	211.0	214.0	216.0
194.0	188.0	185.0	188.0	197.0	207.0	211.0	213.0	210.0	213.0	215.0
181.0	182.0	184.0	189.0	195.0	201.0	205.0	207.0	209.0	211.0	211.0
[illegible]	180.0	181.0	186.0	192.0	197.0	200.0	202.0	206.0	207.0	208.0
[illegible]	[illegible]	[illegible]	183.0	188.0	193.0	195.0	197.0	201.0	202.0	203.0

Histogram : Digital Image



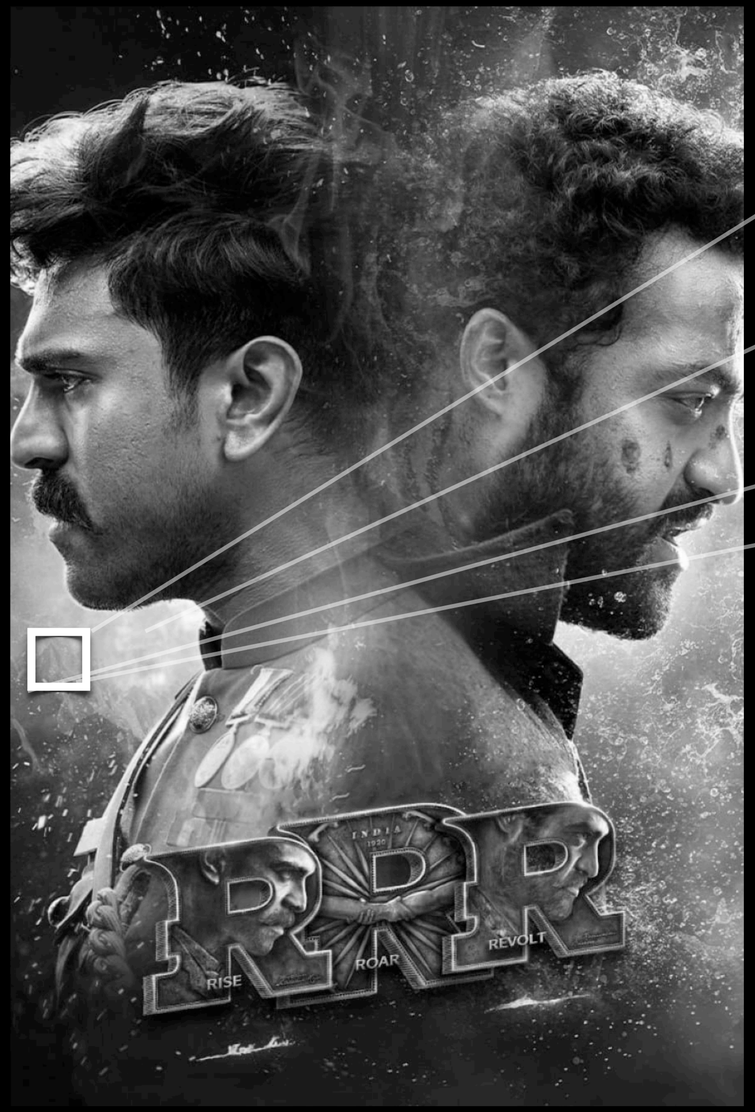
255	255	255	255	255	251	248	247	251	252	254
255	255	255	255	253	250	248	248	252	254	255
253	253	252	250	250	249	248	248	254	255	255
255	253	250	248	248	248	246	246	254	255	255
255	254	249	246	247	248	246	245	250	252	254
255	252	246	243	246	249	250	249	250	251	253
253	247	242	240	245	251	253	253	251	252	254
250	244	238	240	246	253	255	255	252	255	255
241	240	241	243	248	251	254	254	255	255	255
241	242	240	244	247	250	252	252	254	255	255
242	242	241	243	245	248	249	249	251	252	252

247	250	253	254	252	249	247	246	247	248	249
245	247	250	251	250	248	247	247	250	250	251
242	244	245	248	249	250	248	247	252	252	252
245	244	244	246	248	249	249	248	252	253	253
249	246	244	245	248	250	250	248	252	250	251
250	245	242	243	247	251	252	250	249	250	249
244	240	236	239	246	253	254	253	250	250	249
240	234	232	235	245	254	255	255	250	249	250
227	229	232	238	244	249	251	250	250	249	248
227	228	230	235	242	246	247	246	247	245	244
227	227	229	233	239	243	243	242	243	241	241

133	135	138	137	137	136	138	140	147	151	155
130	132	134	134	135	135	138	141	149	153	157
124	125	128	129	131	133	136	141	151	155	158
124	123	122	123	126	130	134	139	149	153	158
122	119	118	118	121	127	130	135	145	149	154
117	112	109	109	117	124	129	133	140	144	149
105	100	96	99	108	120	126	131	134	141	145
93	87	86	89	101	114	123	129	129	135	140
76	77	79	86	95	104	112	117	124	129	132
69	70	73	80	88	95	103	108	117	121	126
66	66	69	74	81	89	95	100	110	113	117

211.0	213.0	215.0	215.0	214.0	212.0	211.0	211.0	215.0	217.0	219.0
210.0	211.0	213.0	213.0	212.0	211.0	211.0	212.0	217.0	219.0	221.0
206.0	207.0	208.0	209.0	210.0	210.0	210.0	212.0	219.0	220.0	221.0
208.0	206.0	205.0	205.0	207.0	209.0	209.0	211.0	218.0	220.0	222.0
208.0	206.0	203.0	203.0	205.0	208.0	208.0	209.0	215.0	217.0	219.0
207.0	203.0	199.0	198.0	203.0	208.0	210.0	210.0	213.0	215.0	217.0
200.0	195.0	191.0	192.0	199.0	208.0	211.0	212.0	211.0	214.0	216.0
194.0	188.0	185.0	188.0	197.0	207.0	211.0	213.0	210.0	213.0	215.0
181.0	182.0	184.0	189.0	195.0	201.0	205.0	207.0	209.0	211.0	211.0
[illegible]	180.0	181.0	186.0	192.0	197.0	200.0	202.0	206.0	207.0	208.0
[illegible]	[illegible]	[illegible]	183.0	188.0	193.0	195.0	197.0	201.0	202.0	203.0

Histogram : Digital Image



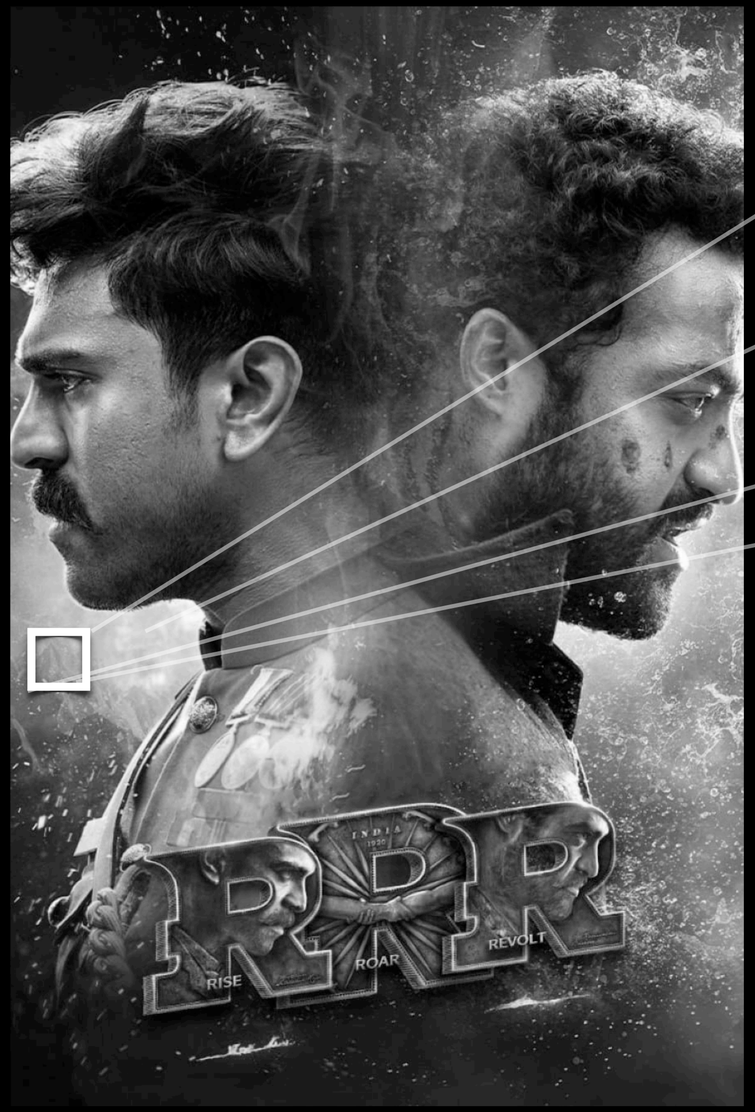
255	255	255	255	255	251	248	247	251	252	254
255	255	255	255	253	250	248	248	252	254	255
253	253	252	250	250	249	248	248	254	255	255
255	253	250	248	248	248	246	246	254	255	255
255	254	249	246	247	248	246	245	250	252	254
255	252	246	243	246	249	250	249	250	251	253
253	247	242	240	245	251	253	253	251	252	254
250	244	238	240	246	253	255	255	252	255	255
241	240	241	243	248	251	254	254	255	255	255
241	242	240	244	247	250	252	252	254	255	255
242	242	241	243	245	248	249	249	251	252	252

247	250	253	254	252	249	247	246	247	248	249
245	247	250	251	250	248	247	247	250	250	251
242	244	245	248	249	250	248	247	252	252	252
245	244	244	246	248	249	249	248	252	253	253
249	246	244	245	248	250	250	248	252	250	251
250	245	242	243	247	251	252	250	249	250	249
244	240	236	239	246	253	254	253	250	250	249
240	234	232	235	245	254	255	255	250	249	250
227	229	232	238	244	249	251	250	250	249	248
227	228	230	235	242	246	247	246	247	245	244
227	227	229	233	239	243	243	242	243	241	241

133	135	138	137	137	136	138	140	147	151	155
130	132	134	134	135	135	138	141	149	153	157
124	125	128	129	131	133	136	141	151	155	158
124	123	122	123	126	130	134	139	149	153	158
122	119	118	118	121	127	130	135	145	149	154
117	112	109	109	117	124	129	133	140	144	149
105	100	96	99	108	120	126	131	134	141	145
93	87	86	89	101	114	123	129	129	135	140
76	77	79	86	95	104	112	117	124	129	132
69	70	73	80	88	95	103	108	117	121	126
66	66	69	74	81	89	95	100	110	113	117

211.0	213.0	215.0	215.0	214.0	212.0	211.0	211.0	215.0	217.0	219.0
210.0	211.0	213.0	213.0	212.0	211.0	211.0	212.0	217.0	219.0	221.0
206.0	207.0	208.0	209.0	210.0	210.0	210.0	212.0	219.0	220.0	221.0
208.0	206.0	205.0	205.0	207.0	209.0	209.0	211.0	218.0	220.0	222.0
208.0	206.0	203.0	203.0	205.0	208.0	208.0	209.0	215.0	217.0	219.0
207.0	203.0	199.0	198.0	203.0	208.0	210.0	210.0	213.0	215.0	217.0
200.0	195.0	191.0	192.0	199.0	208.0	211.0	212.0	211.0	214.0	216.0
194.0	188.0	185.0	188.0	197.0	207.0	211.0	213.0	210.0	213.0	215.0
181.0	182.0	184.0	189.0	195.0	201.0	205.0	207.0	209.0	211.0	211.0
[illegible]	180.0	181.0	186.0	192.0	197.0	200.0	202.0	206.0	207.0	208.0
[illegible]	[illegible]	[illegible]	183.0	188.0	193.0	195.0	197.0	201.0	202.0	203.0

Histogram : Digital Image



255	255	255	255	255	251	248	247	251	252	254
255	255	255	255	253	250	248	248	252	254	255
253	253	252	250	250	249	248	248	254	255	255
255	253	250	248	248	248	246	246	254	255	255
255	254	249	246	247	248	246	245	250	252	254
255	252	246	243	246	249	250	249	250	251	253
253	247	242	240	245	251	253	253	251	252	254
250	244	238	240	246	253	255	255	252	255	255
241	240	241	243	248	251	254	254	255	255	255
241	242	240	244	247	250	252	252	254	255	255
242	242	241	243	245	248	249	249	251	252	252

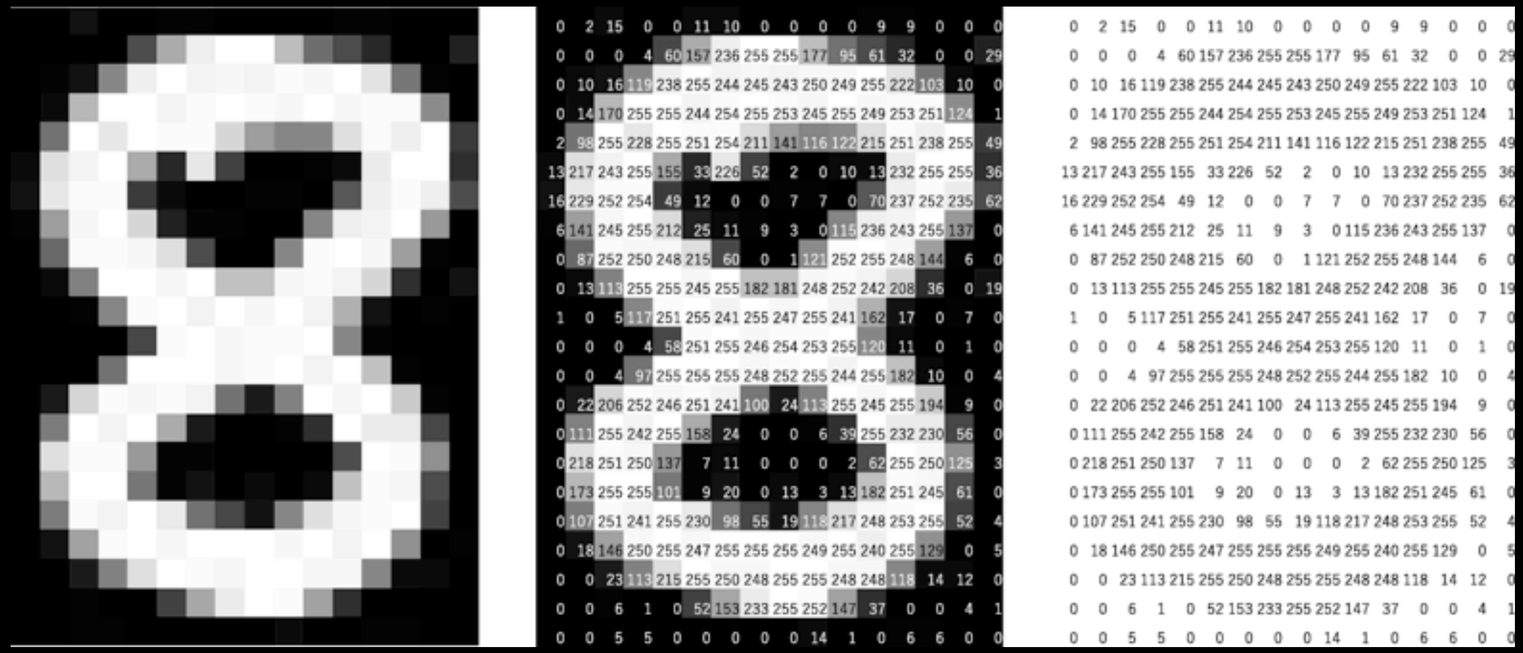
247	250	253	254	252	249	247	246	247	248	249
245	247	250	251	250	248	247	247	250	250	251
242	244	245	248	249	250	248	247	252	252	252
245	244	244	246	248	249	249	248	252	253	253
249	246	244	245	248	250	250	248	252	250	251
250	245	242	243	247	251	252	250	249	250	249
244	240	236	239	246	253	254	253	250	250	249
240	234	232	235	245	254	255	255	250	249	250
227	229	232	238	244	249	251	250	250	249	248
227	228	230	235	242	246	247	246	247	245	244
227	227	229	233	239	243	243	242	243	241	241

133	135	138	137	137	136	138	140	147	151	155
130	132	134	134	135	135	138	141	149	153	157
124	125	128	129	131	133	136	141	151	155	158
124	123	122	123	126	130	134	139	149	153	158
122	119	118	118	121	127	130	135	145	149	154
117	112	109	109	117	124	129	133	140	144	149
105	100	96	99	108	120	126	131	134	141	145
93	87	86	89	101	114	123	129	129	135	140
76	77	79	86	95	104	112	117	124	129	132
69	70	73	80	88	95	103	108	117	121	126
66	66	69	74	81	89	95	100	110	113	117

211.0	213.0	215.0	215.0	214.0	212.0	211.0	211.0	215.0	217.0	219.0
210.0	211.0	213.0	213.0	212.0	211.0	211.0	212.0	217.0	219.0	221.0
206.0	207.0	208.0	209.0	210.0	210.0	210.0	212.0	219.0	220.0	221.0
208.0	206.0	205.0	205.0	207.0	209.0	209.0	211.0	218.0	220.0	222.0
208.0	206.0	203.0	203.0	205.0	208.0	208.0	209.0	215.0	217.0	219.0
207.0	203.0	199.0	198.0	203.0	208.0	210.0	210.0	213.0	215.0	217.0
200.0	195.0	191.0	192.0	199.0	208.0	211.0	212.0	211.0	214.0	216.0
194.0	188.0	185.0	188.0	197.0	207.0	211.0	213.0	210.0	213.0	215.0
181.0	182.0	184.0	189.0	195.0	201.0	205.0	207.0	209.0	211.0	211.0
[illegible]	180.0	181.0	186.0	192.0	197.0	200.0	202.0	206.0	207.0	208.0
[illegible]	[illegible]	[illegible]	183.0	188.0	193.0	195.0	197.0	201.0	202.0	203.0

Histogram : Grayscale Images

MNIST Data

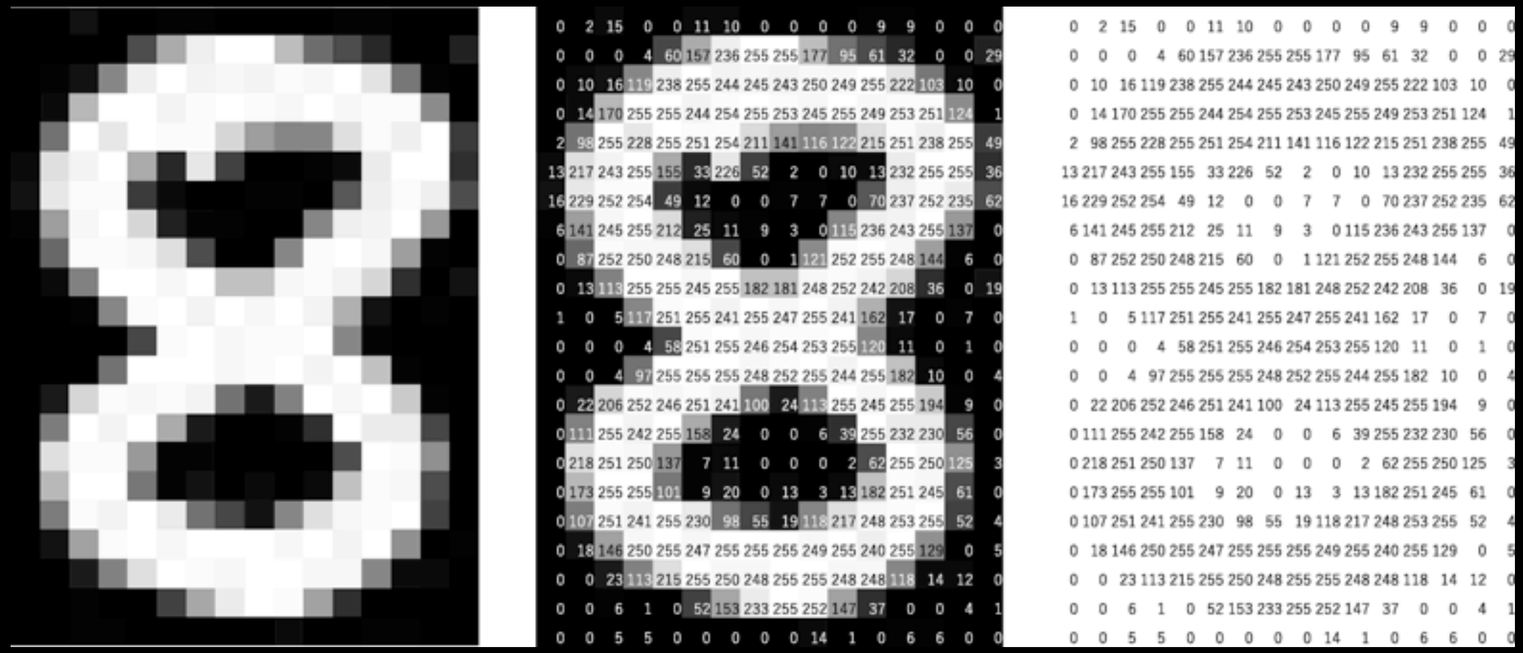


Source : https://github.com/NehaCkumari/MNIST-Handwritten-Digit-Recognition-using-CNN

Histogram : Grayscale Images

- Image with single Channel

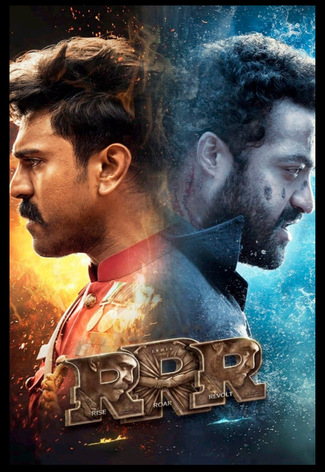
MNIST Data

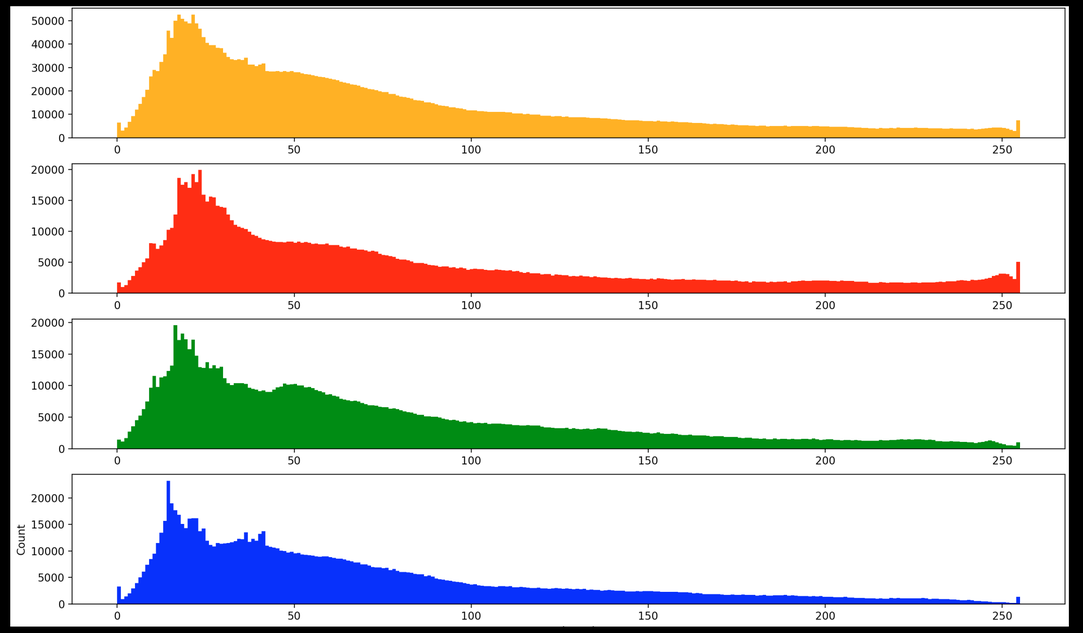


Source : https://github.com/NehaCkumari/MNIST-Handwritten-Digit-Recognition-using-CNN

Histogram : Let's Define Goals

- Always reiterate inputs
- And where we need to reach from here

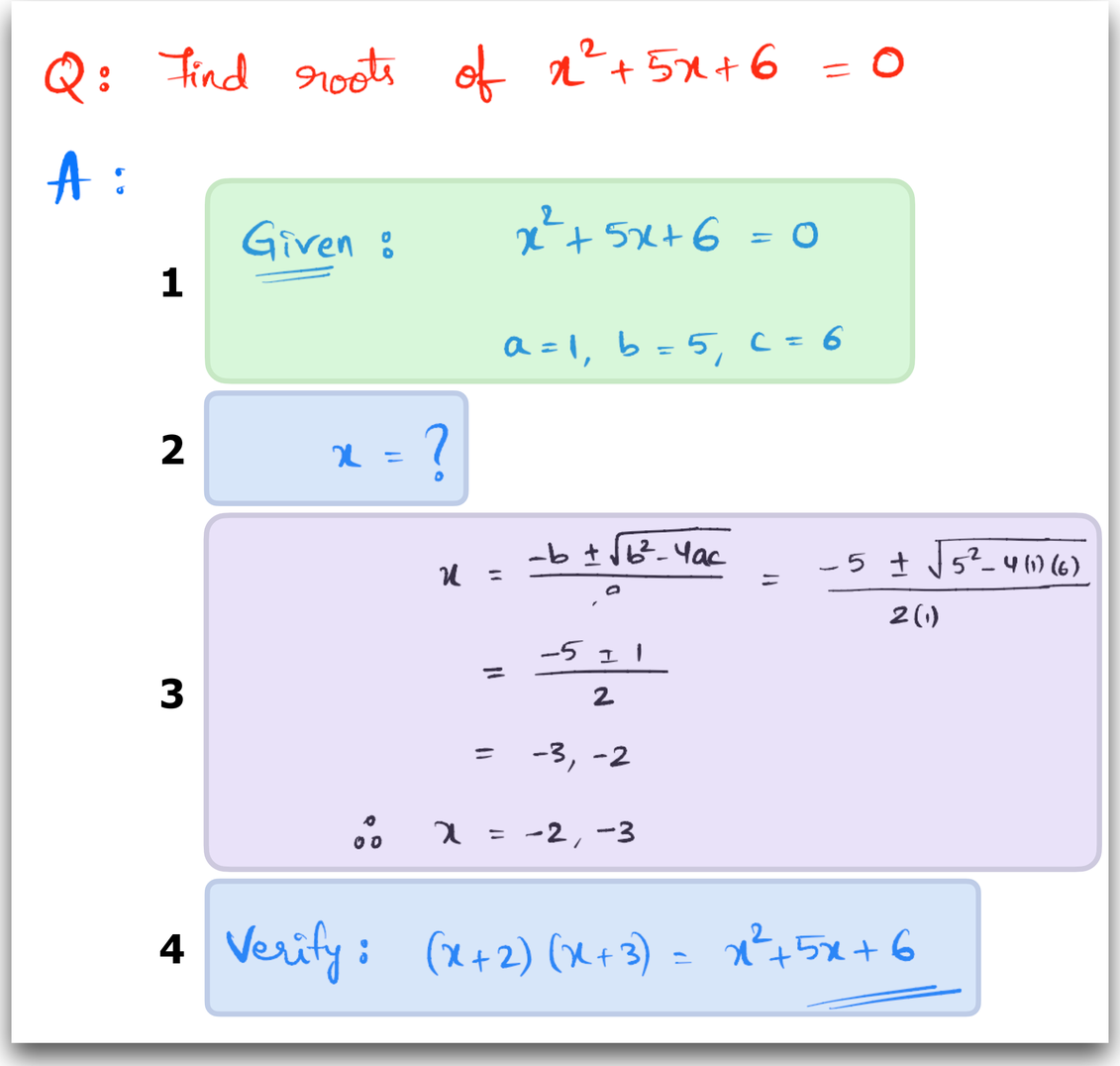




How to approach any problem ?

1. Note down given information
2. Note down what we desire to find
3. Go on with the approach
4. Verify your result with the desired result

STAR : Situation, Task, Action, Result



How to approach our problem ?

1. Given Image.
2. Required output : Histogram
3. Go on with the coding and matrix manipulations
4. Verify your result with the desired result

STAR : Situation, Task, Action, Result

Computer Programming :

- Why do we need language?
  - To communicate with ourselves
  - Life evolution

Natural Language :

- Why do we need language?
  - To communicate with ourselves
  - Life evolution

Natural Language :

- Earth before larynx?
- Before reptiles
- Frogs were first to develop larynx
- Why?
- Or What advantage?
- Dinosaurs in Jurassic Park

https://www.scientificamerican.com/article/fossils-reveal-when-animals-started-making-noise/

Natural Language :

- Why do we need language?
  - To communicate with ourselves
  - Helped animals to survive and mate.
- Humans were first to map words with abstract ideas.
- Humans were first to map words with visual symbols
- Later, humans framed rules for communication
- And, called it language
- Grammar

- Language is symbols (with rules) for real world.

https://socratic.org/questions/what-is-an-abstract-idea

https://guernseydonkey.com/the-origin-of-writing/

Computer Programming :

Computer Programming :

- Computer Languages

Computer Programming :

- Computer Languages
- To tinker with the computers

Computer Programming :

- Computer Languages
- To tinker with the computers
- To make computers do what we want

Computer Programming :

- Computer Languages
- To tinker with the computers
- To make computers do what we want
- A set of instructions in the particular language : **Code , program**

Computer Programming :

- Computer Languages
- To tinker with the computers
- To make computers do what we want
- A set of instructions in the particular language : **Code , program**
- Any computer code cycle :

Computer Programming :

- Computer Languages
- To tinker with the computers
- To make computers do what we want
- A set of instructions in the particular language : **Code , program**
- Any computer code cycle :
  - Human writes the code using keyboard

Computer Programming :

- Computer Languages
- To tinker with the computers
- To make computers do what we want
- A set of instructions in the particular language : **Code , program**
- Any computer code cycle :
  - Human writes the code using keyboard
  - The code will be converted to 0 & 1 (010010100..) - using interpreter or compiler

Computer Programming :

- Computer Languages
- To tinker with the computers
- To make computers do what we want
- A set of instructions in the particular language : **Code , program**
- Any computer code cycle :
  - Human writes the code using keyboard
  - The code will be converted to 0 & 1 (010010100..) - using interpreter or compiler
  - The 0 & 1 (010010100..) code will be passed to microprocessor

Computer Programming :

- Computer Languages
- To tinker with the computers
- To make computers do what we want
- A set of instructions in the particular language : **Code , program**
- Any computer code cycle :
  - Human writes the code using keyboard
  - The code will be converted to 0 & 1 (010010100..) - using interpreter or compiler
  - The 0 & 1 (010010100..) code will be passed to microprocessor
  - Output from microprocessor will also be in 0 & 1 (010010100..)

Computer Programming :

- Computer Languages
- To tinker with the computers
- To make computers do what we want
- A set of instructions in the particular language : **Code , program**
- Any computer code cycle :
  - Human writes the code using keyboard
  - The code will be converted to 0 & 1 (010010100..) - using interpreter or compiler
  - The 0 & 1 (010010100..) code will be passed to microprocessor
  - Output from microprocessor will also be in 0 & 1 (010010100..)
  - Need to be parsed again to the output we require / understand

Computer Programming :

- Question 1 :
  - How do we store the real world data into computers?
  - Especially to the programming language?

- How do we do in real world?
- Simpler question : How do we store information in real world ? Before computers ?

- Take a book / taalapatram
- Take a pen / ghantamu / dabbalamu
- Jot down the ideas

Computer Programming :

Computer Programming :

- Library

http://www.renovatedlearning.com/2014/02/14/3-reasons-to-rethink-your-reference-section/

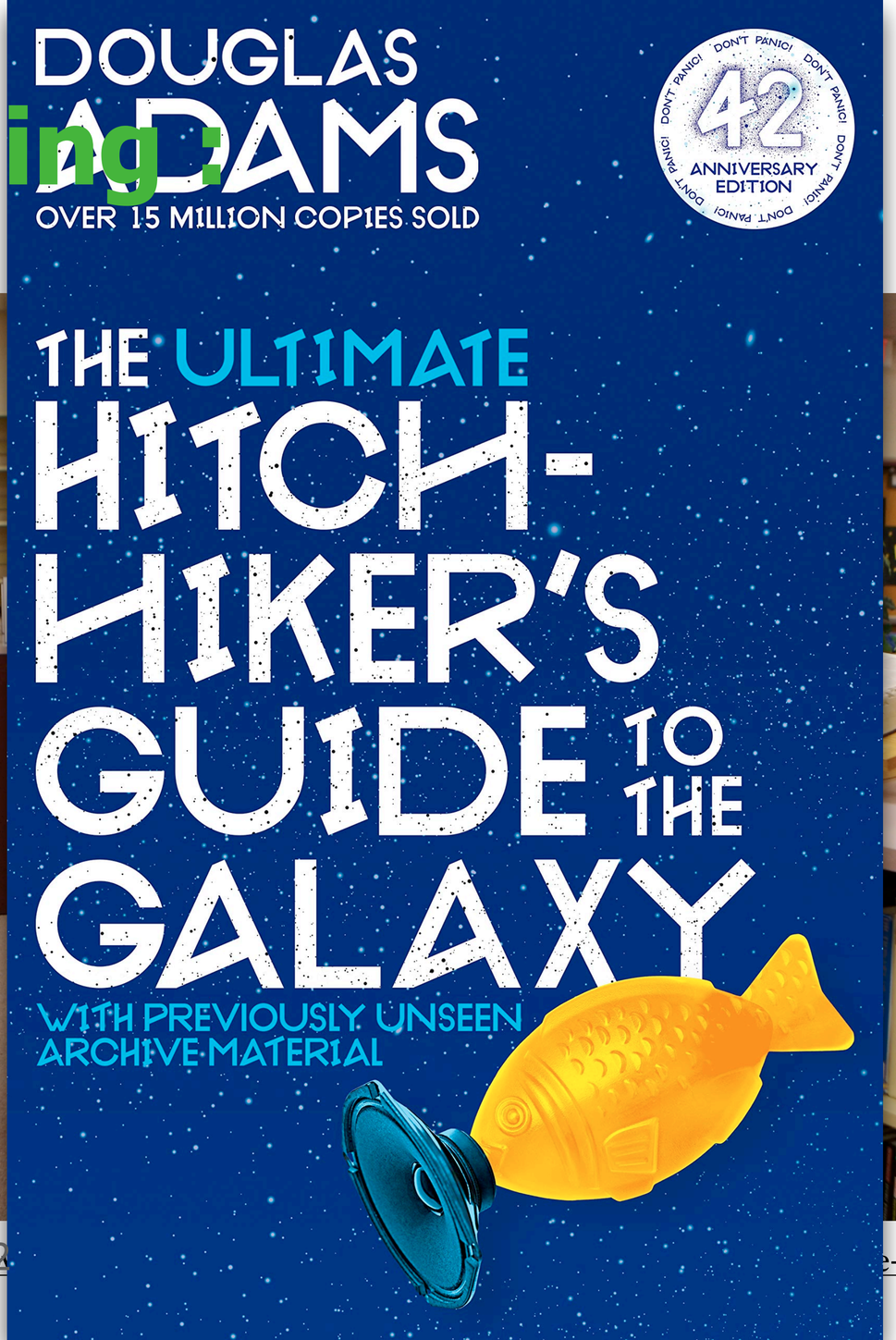
Computer Programming :

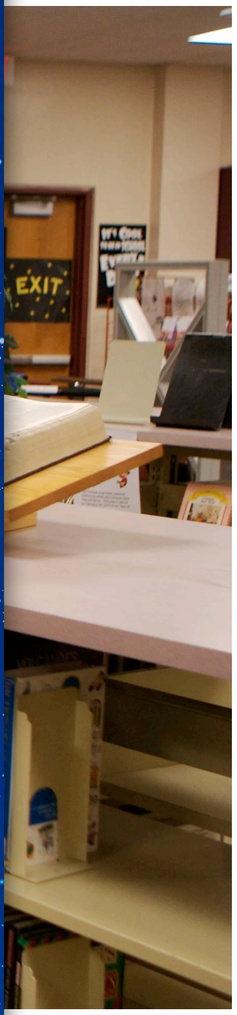
- Library
- Sections

http://www.renovatedlearning.com/2014/02/14/3-reasons-to-rethink-your-reference-section/

Computer Programming :

- Library
- Sections
- Book Name





http://www.ren 42 e-section/

Computer Programming:

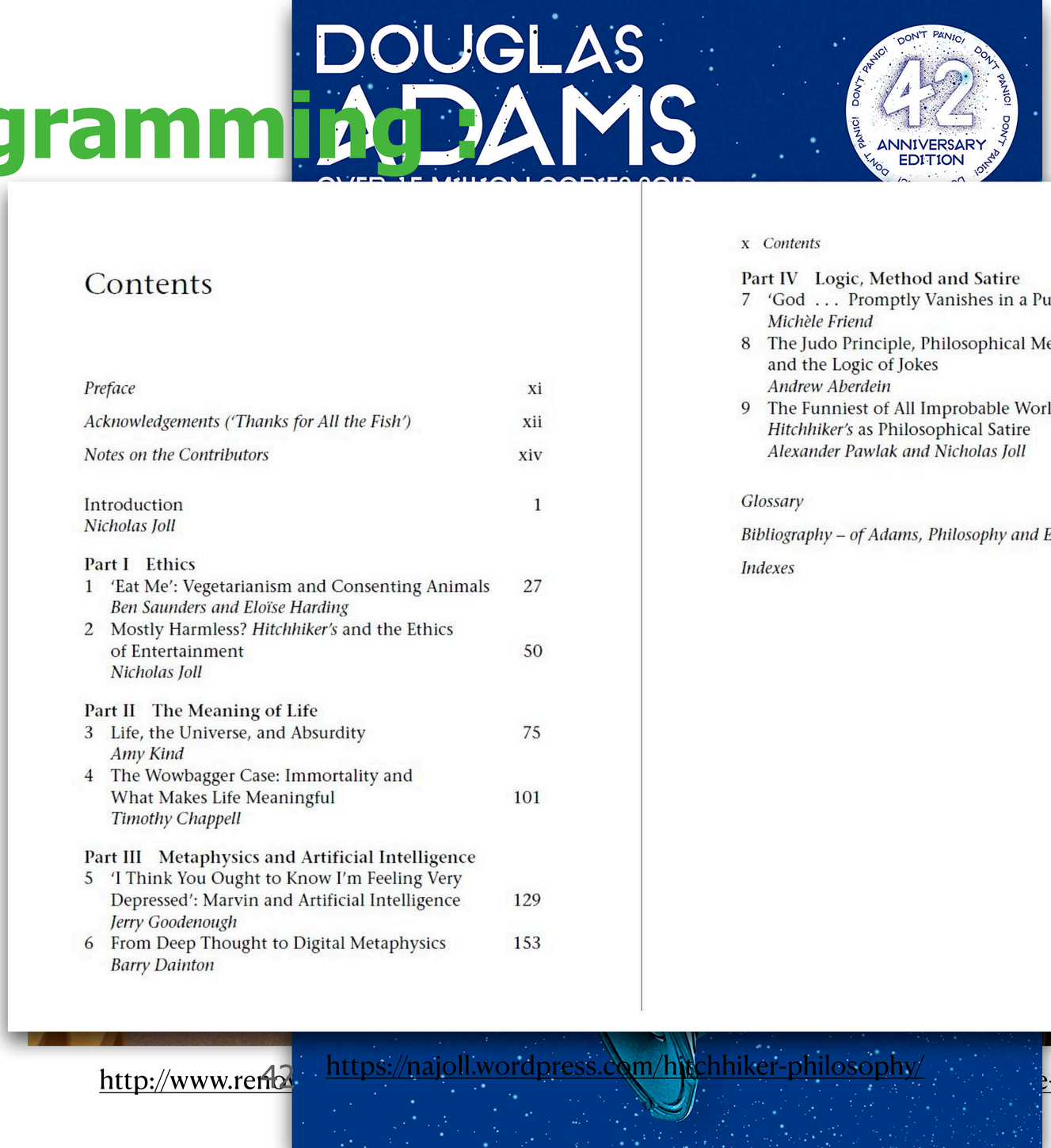
- Library
- Sections
- Book Name
- Book contents

Contents

http://www.ren… 42 https://najoll.wordpress.com/hitchhiker-philosophy/ …e-section/

Computer Programming:

- Library
- Sections
- Book Name
- Book contents
- Chapter

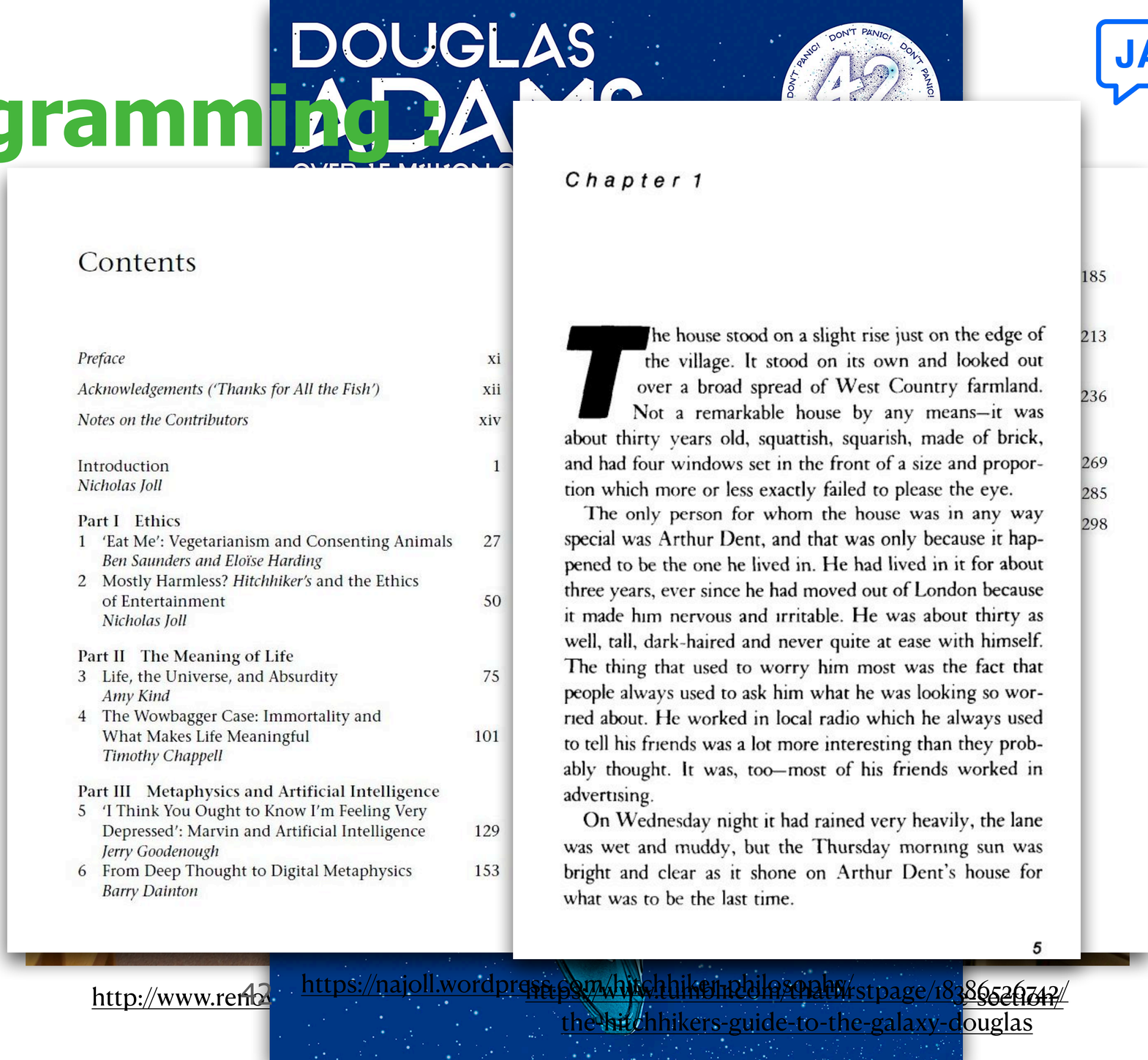


Contents

http://www.ren… 42 https://najoll.wordpress.com/hitchhiker-philosophy/ …e-section/

Computer Programming :

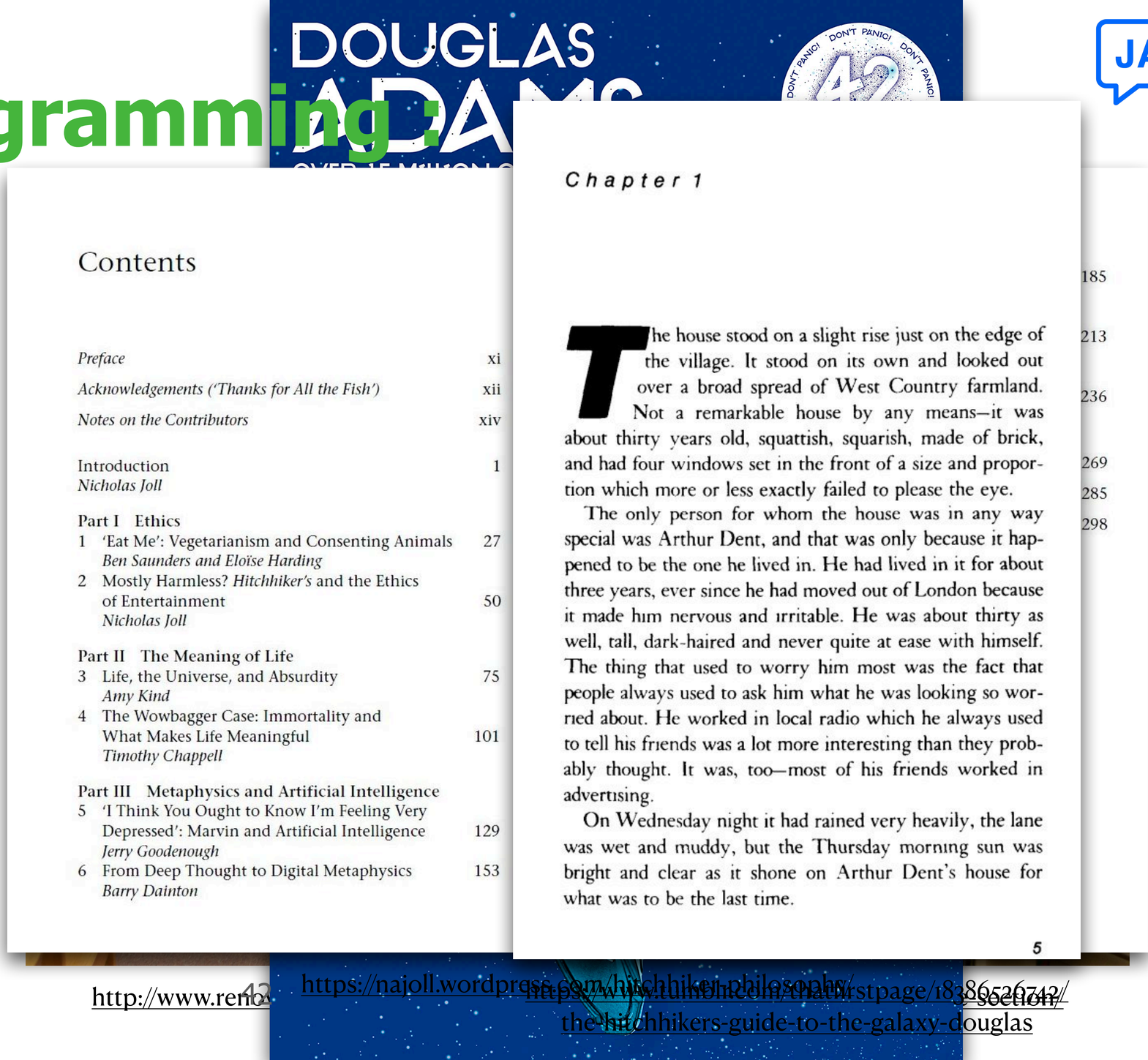
- Library
- Sections
- Book Name
- Book contents
- Chapter
- Paragraph = Information



http://www.rene... 42 https://najoll.wordpress.com/hitchhikers-philosophy/ https://www.tumblr.com/thatfirstpage/183652674 2/the-hitchhikers-guide-to-the-galaxy-douglas

Computer Programming :

- Library
- Sections
- Book Name
- Book contents
- Chapter
- Paragraph = Information

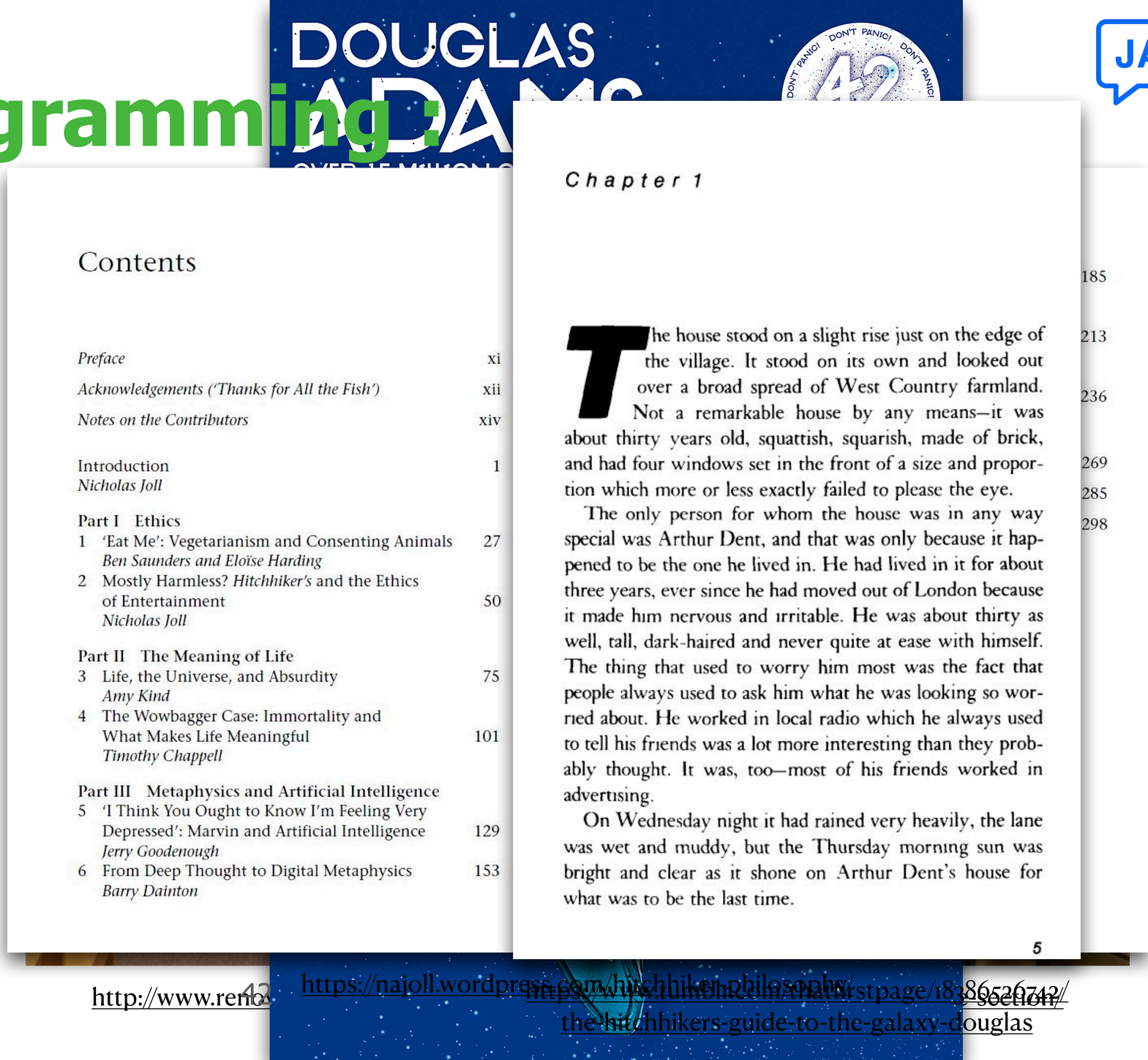


http://www.rene... 42 https://najoll.wordpress.com/hitchhikers-philosophy/ https://www.tumblr.com/thatfirstpage/183652674 2/the-hitchhikers-guide-to-the-galaxy-douglas

Computer Programming :

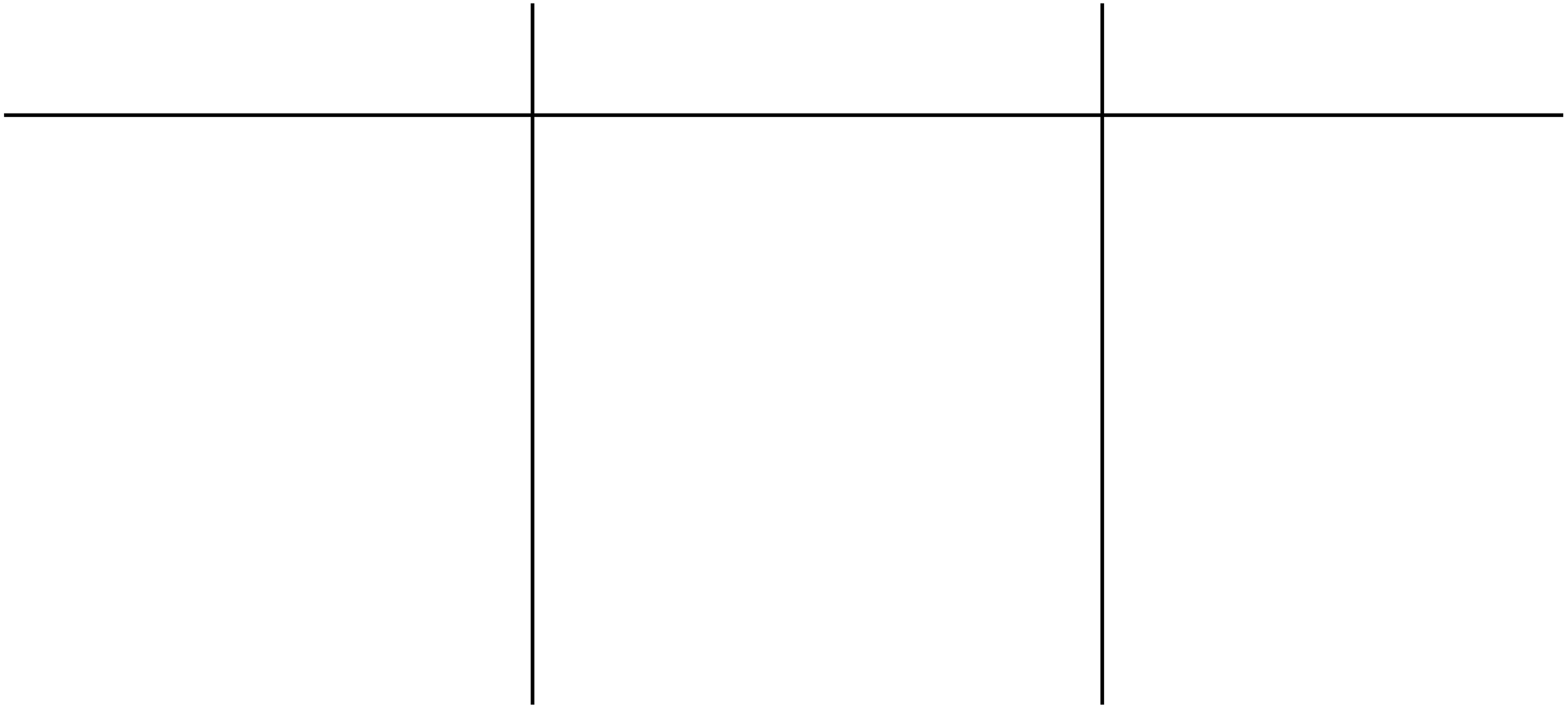
- Library
- Sections
- Book Name
- Book contents
- Chapter
- Paragraph = Information

- What about them in the programming languages?
- C, Python, Matlab



http://www.rene... 42 https://najoll.wordpress.com/hitchhikers-philosophy/ https://www.tumblr.com/thatfirstpage/183865267 42/section/the-hitchhikers-guide-to-the-galaxy-douglas

Computer Programming : Variables



Computer Programming : Variables

C		

• int -> 2 bytes
• float -> 4 bytes
• char
  • 1 byte
• bool
  • 1 bit		

Computer Programming : Variables

C

- int -> 2 bytes
- float -> 4 bytes
- char
  - 1 byte
- bool
  - 1 bit

- int age = 42
- float weight = 23.24
- char first_letter = “a”
- bool light = True

Computer Programming : Variables

C	Python	

- int -> 2 bytes
- float -> 4 bytes
- char
  - 1 byte
- bool
  - 1 bit

- int age = 42
- float weight = 23.24
- char first_letter = “a”
- bool light = True

- Integers
- Float
- Characters/strings
- Boolean

Computer Programming : Variables

C

- int -> 2 bytes
- float -> 4 bytes
- char
  - 1 byte
- bool
  - 1 bit

- int age = 42
- float weight = 23.24
- char first_letter = “a”
- bool light = True

Python

- Integers
- Float
- Characters/strings
- Boolean

- x = 42
- x = 234.23
- x = “hello”
- x = True
- X = False

Computer Programming : Variables

C

- int -> 2 bytes
- float -> 4 bytes
- char
  - 1 byte
- bool
  - 1 bit

- int age = 42
- float weight = 23.24
- char first_letter = “a”
- bool light = True

Python

- Integers
- Float
- Characters/strings
- Boolean

- x = 42
- x = 234.23
- x = “hello”
- x = True
- X = False

Matlab

- Integers
- Float
- Characters/strings
- Boolean

Computer Programming : Variables

C

- int -> 2 bytes
- float -> 4 bytes
- char
  - 1 byte
- bool
  - 1 bit

- int age = 42
- float weight = 23.24
- char first_letter = “a”
- bool light = True

Python

- Integers
- Float
- Characters/strings
- Boolean

- x = 42
- x = 234.23
- x = “hello”
- x = True
- X = False

Matlab

- Integers
- Float
- Characters/strings
- Boolean

- x = 42
- x = 234.23
- x = “hello”
- x = True
- x = False

Computer Programming : Syntax

Computer Programming : Syntax

- Syntax = Grammar
- Natural languages rules is Grammar
- Programming languages rules is Syntax
- Each language has its own rules.

Computer Programming : Syntax

- Syntax = Grammar
- Natural languages rules is Grammar
- Programming languages rules is Syntax
- Each language has its own rules.

Computer Programming : Syntax

- Syntax = Grammar
- Natural languages rules is Grammar
- Programming languages rules is Syntax
- Each language has its own rules.

- C says :
  - “Use semicolon (;) after every statement ;”
  - “I will ignore everything after “//“ and everything between “/* */“ ”.

Computer Programming : Syntax

Computer Programming : Syntax

- Syntax = Grammar
- C says :
  - “Use semicolon (;) after every statement ;”
  - “Use flower brackets {} to each specify scope”
  - “Will ignore everything after “//“ and everything between “/* */“ ”.

Computer Programming : Syntax

- Syntax = Grammar
- C says :
  - “Use semicolon (;) after every statement ;”
  - “Use flower brackets {} to each specify scope”
  - “Will ignore everything after “//“ and everything between “/* */“ ”.
- Python says :
  - “Need indentation for scope”
  - “Will ignore everything after “#“”.

Computer Programming : Syntax

- Syntax = Grammar
- C says :
  - “Use semicolon (;) after every statement ;”
  - “Use flower brackets {} to each specify scope”
  - “Will ignore everything after “//“ and everything between “/* */“ ”.
- Python says :
  - “Need indentation for scope”
  - “Will ignore everything after “#“”.
- Matlab says :
  - “No need for indentation, semicolons and {} ; But I will print if you don’t use (;)”.
  - “Will ignore everything after “%“”.

Computer Programming : Syntax

C

- int -> 2 bytes
- float -> 4 bytes
- char
  - 1 byte
- bool
  - 1 bit

- int x = 42
- float y = 23.24
- char z = “a”
- bool a = True

Python

- Integers
- Float
- Characters/strings
- Boolean

- x = 42
- x = 234.23
- x = “hello”
- x = True
- X = False

Matlab

- Integers
- Float
- Characters/strings
- Boolean

- x = 42
- x = 234.23
- x = “hello”
- x = True
- x = False

Computer Programming : Syntax

C

- int -> 2 bytes
- float -> 4 bytes
- char
  - 1 byte
- bool
  - 1 bit

- int x = 42 ;
- float y = 23.24 ;
- char z = “a” ;
- bool a = True ;

Python

- Integers
- Float
- Characters/strings
- Boolean

- x = 42
- x = 234.23
- x = “hello”
- x = True
- X = False

Matlab

- Integers
- Float
- Characters/strings
- Boolean

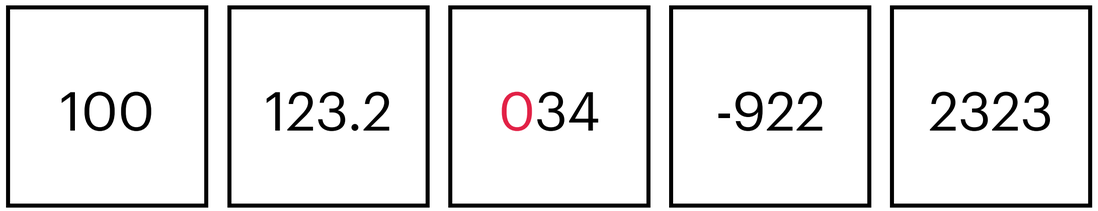
- x = 42
- x = 234.23
- x = “hello”
- x = True
- x = False

Computer Programming : Arrays

- We need group of values often.
- Roll numbers, Names, Sentences
- To store them we need arrays.

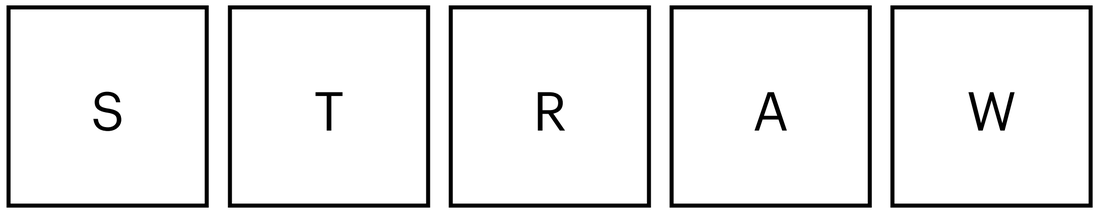
- C : “I will give only same type arrays.”
  - int x[100]; float y[100]; char z[100]; bool p[100];
- Python : “Take any type you want”
  - x = [100, 123.345, “Hello, world!”, True, False]
  - x = [“string”, [100, 123.345, “Hello, world!”, True, False]]
- Matlab : “You can have an array of Numbers”
  - x = [100, 20.23, 502, 1e2];

Computer Programming : Arrays & Strings



Arrays

[100, 123.2, 34, -922, 2323]



Strings

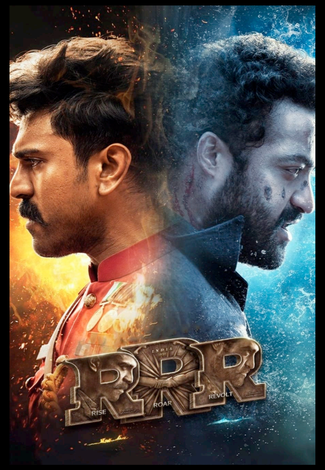
“Straw” or [“s”, “t”, “r”, “a”, “w”]

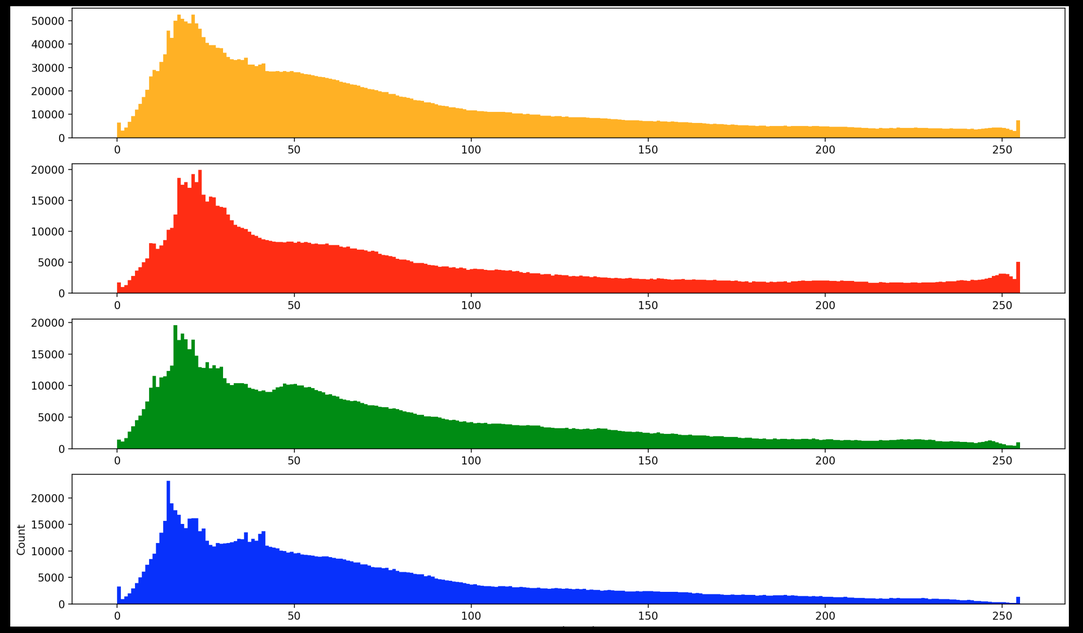
Computer Programming : Functions

- Functions
- Custom Variables :
  - Custom variables in C, Python, Matlab
  - Blueprints
  - Classes
  - Object Oriented Programming
- Libraries
  - Need for libraries
  - Libraries in C, Python, Matlab

Histogram in Python

- Always reiterate inputs
- And where we need to reach from here





Natural Language - Metaphors :

- Up
- Down
- Warm
- Cool

Assignment 2 : George Lakoff and Mark Johnsen - **Metaphors we live by (2003)**

One person per chapter - 2 to 3 pages max

Koushik and Madhava will assign you.

Markdown :

- Similar to HTML
- Easy to parse

- Online tool : https://dillinger.io/
- https://www.markdowntutorial.com/
- https://www.markdownguide.org/cheat-sheet/

Element	Markdown Syntax
Heading	# H1 ## H2 ### H3
Bold	**bold text**
Italic	*italicized text*
Blockquote	> blockquote
Ordered List	1. First item 2. Second item 3. Third item
Unordered List	- First item - Second item - Third item
Code	`code`
Horizontal Rule	---
Link	[title](https://www.example.com)
Image	![alt text](image.jpg)

Thank you

Be Hungry & Active

https://webbtelescope.org/contents/media/images/2022/053/01GFRYSFM89AFADVAA0W625BSB