

image processing

by using matrix multiplication, addition and subtraction

slash @ https://github.com/KvN1027/la_project_image_processing

outline

image processing

1.intro

2.based knowledge

3.blur

4.edge sketching

5.notetime

6.conclusion

slash @ https://github.com/KvN1027/la_project_image_processing

intro

slash @ https://github.com/KvN1027/la_project_image_processing



the application of linear algebra



Application of linear algebra

Linear algebra

application of linear algebra **in engineering**

application of linear algebra **in games**

applications of linear algebra **in real life ppt**

application of linear algebra **in data science**

application of linear algebra **in computer science pdf**

application of linear algebra **in machine learning**

linear algebra **applications project**

linear **algebra**教學

回報不適當的預測查詢字串

slash @ https://github.com/KvN1027/la_project_image_processing

Applications of linear algebra

- 1)The geometry of the surrounding space. ...
- 2)Analyse of functionality. ...
- 3)Research into more complicated systems. ...
- 4)Scientific computation. ...
- 5)Cryptography. ...
- 6)The theory of games.

更多項目...

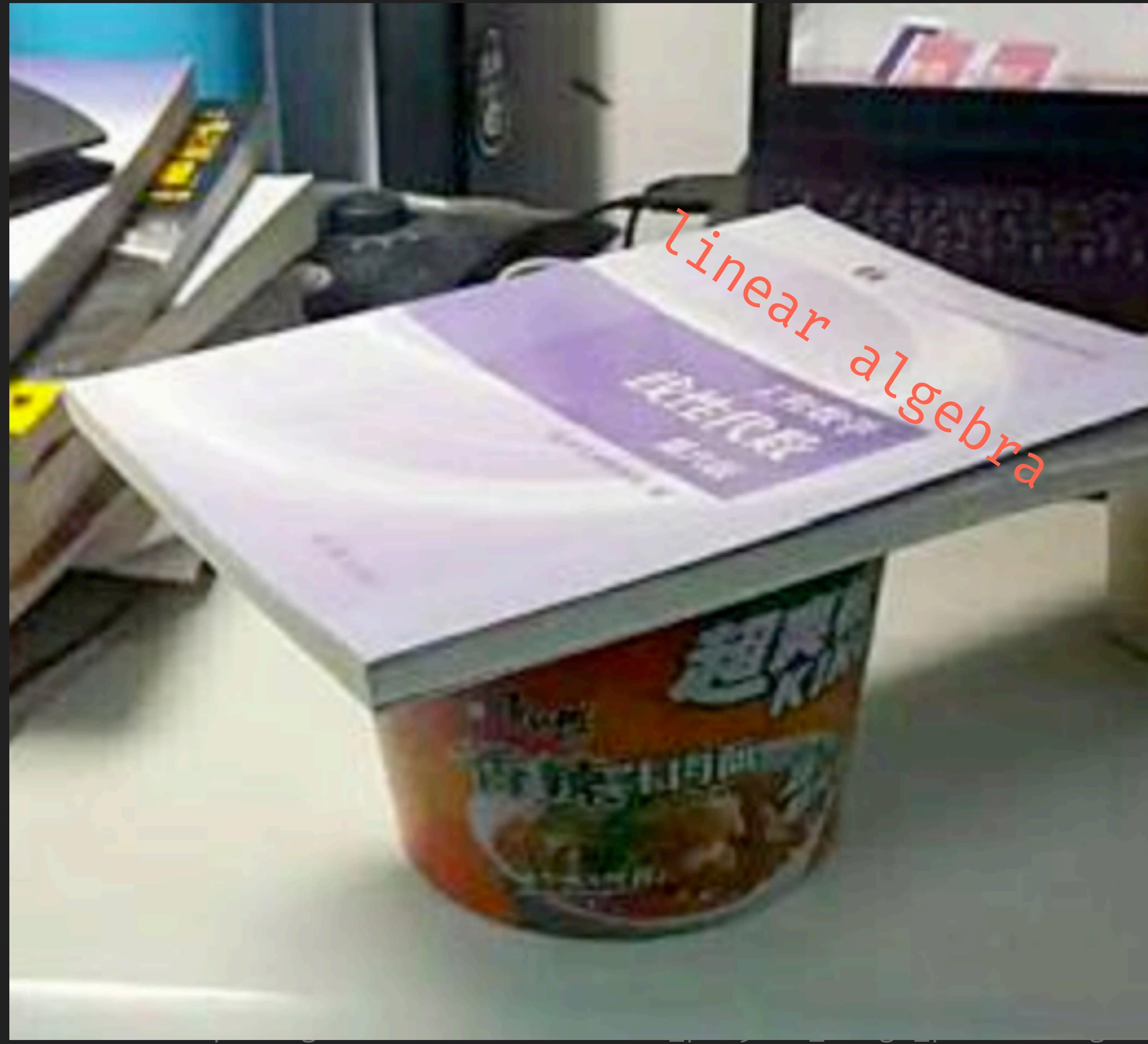


Unacademy

<https://unacademy.com> › ... › Mathematical Sciences

⋮

[Small Note on Linear Algebra Applications - Unacademy](#)



< 99+

好玩線代區 (7)



hsinn(李羿昕)

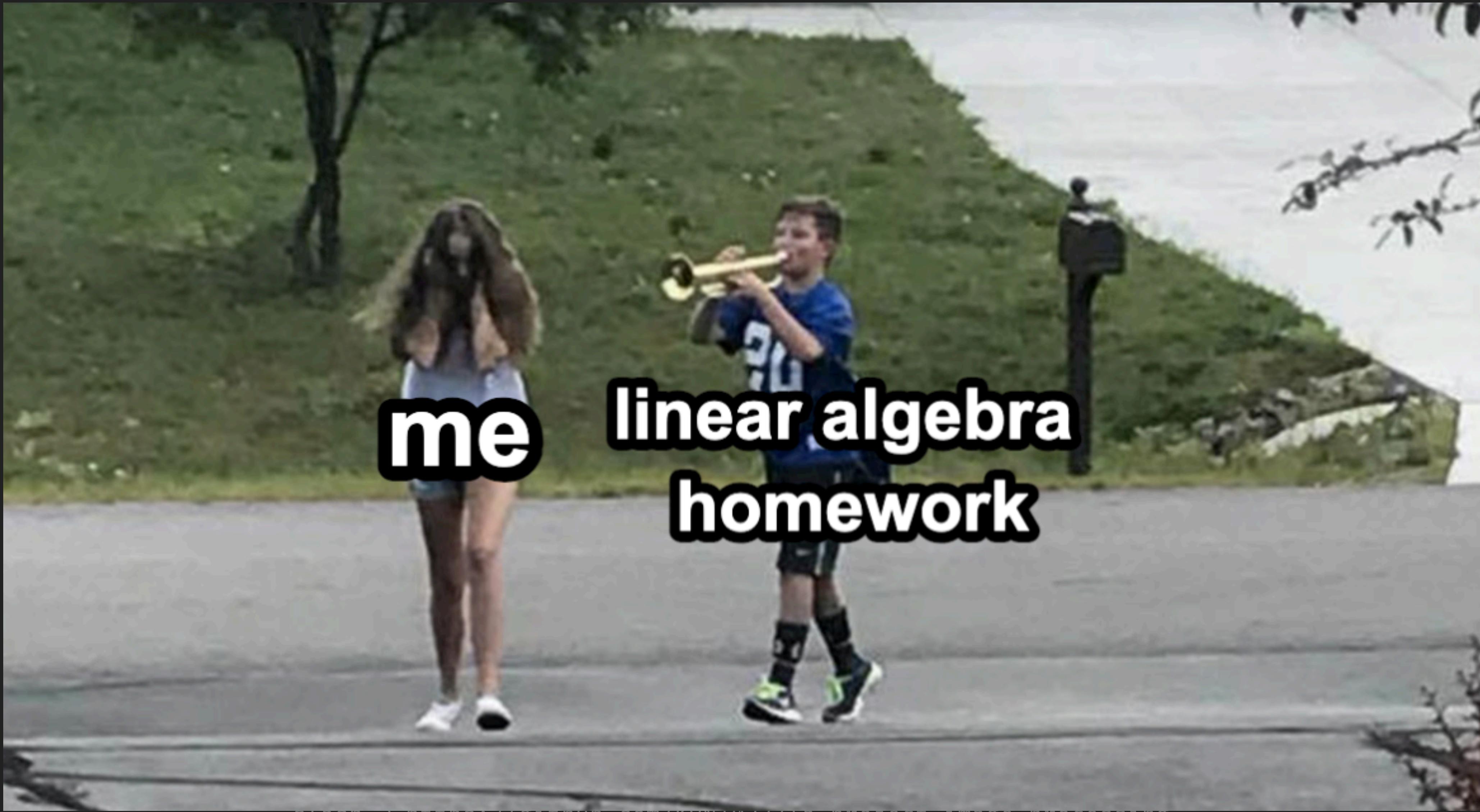
欸不對那個筆記今天要交 homework is due today
但我也都沒做 😭 but I do nothing 😭

晚上 11:41

有人有寫嗎

anybody do it?

晚上 11:41



me linear algebra
homework

← → C https://www.enotes.com/homework-help/an-electron-fired-into-parallel-plate-moving-1-5-x-336861

Charge, e is = 1.602×10^{-19} C

Mass m...

2 EDUCATOR ANSWERS

How do you calculate the mass of neutrons?

1 EDUCATOR ANSWER

What is the order of colors in a rainbow?

7 EDUCATOR ANSWERS

What is the difference between smooth endoplasmic reticulum and rough endoplasmic reticulum?

1 EDUCATOR ANSWER

What are ten examples of organelles that you might find in your body?

2 EDUCATOR ANSWERS

MORE SCIENCE QUESTIONS

Unlock All Answers Now

Start your **48-hour free trial** to unlock this answer and thousands more, and enjoy eNotes ad-free.

Start Free Trial to Unlock

[Already a member? Log in here.](#)

Physics 12 - September 2018

e An electron is fired into a parallel plate moving 1-5-x-336861

C https://www.enotes.com/homework-help/an-electron-fired-into-parallel-plate-moving-1-5-x-336861

Cyber Monday sale—all week long! Get 50% off an eNotes subscription this week only. Join Now X

F = eE where E is electric field.

F = $1.602 \times 10^{-19} \text{ C} \times 15000 \text{ N/C}$

F = $2.403 \times 10^{-15} \text{ N}$.

The acceleration on the electron is,

F = ma

a = F/m

a = $2.403 \times 10^{-15} \text{ N} / 9.109 \times 10^{-31} \text{ kg}$

a = $2.638 \times 10^{15} \text{ ms}^{-2}$.

Let's find the distance the electron travels in y direction in time t,

For this $s = ut + at^2/2$ can be used.

U is initial velocity which is 0 in y direction,

Elements Console Sources Network F

... noreact >_</div>

><div class="...>

><p>_</p>

><p>_</p>

><p>Charge, e is = 1.602×10^{-19}

><p>Mass m is, = 9.109×10^{-31}

><p> </p>

><p>Let's first analyse the motion

><p> $V_x = 1.5 \times 10^7 \text{ ms}^{-1}$ </p>

><p>The time taken to pass the gap

><p> $t = 0.12 \text{ m} / 1.5 \times 10^7 \text{ ms}^{-1}$ </p>

><p> </p>

><p>Now let's analyse the motion

... #page-wrapper div #content div div div div div

Styles Event Listeners DOM Breakpoints Properties Ad

Filter :hov .cls +

element.style {

}

p { margin-bottom: 1.3em; }

p, pre { margin: 1em 0; }

* { box-sizing: border-box; }

p { display: block; margin-block-start: 1em; margin-block-end: 1em; margin-inline-start: 0px; }

: Console What's New X

Highlights from the Chrome 70 update

Live Expressions in the Console

Pin expressions to the top of the Console to monitor their value in real-time.

Highlight DOM nodes during Eager Evaluation

Type an expression that evaluates to a node to highlight the node in the viewport.

Autocomplete Conditional Breakpoints



original image

slash @ https://github.com/KvN1027/la_project_image_processing



original image

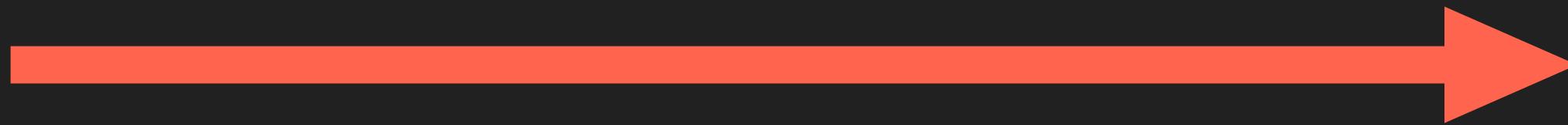


image process



blurred image

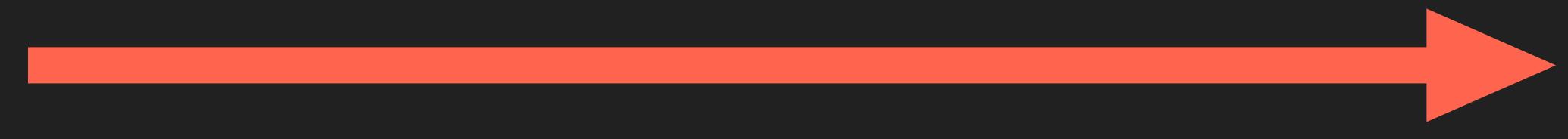


image process

slash @ https://github.com/KvN1027/la_project_image_processing

based knowledge

slash @ https://github.com/KvN1027/la_project_image_processing

RGB

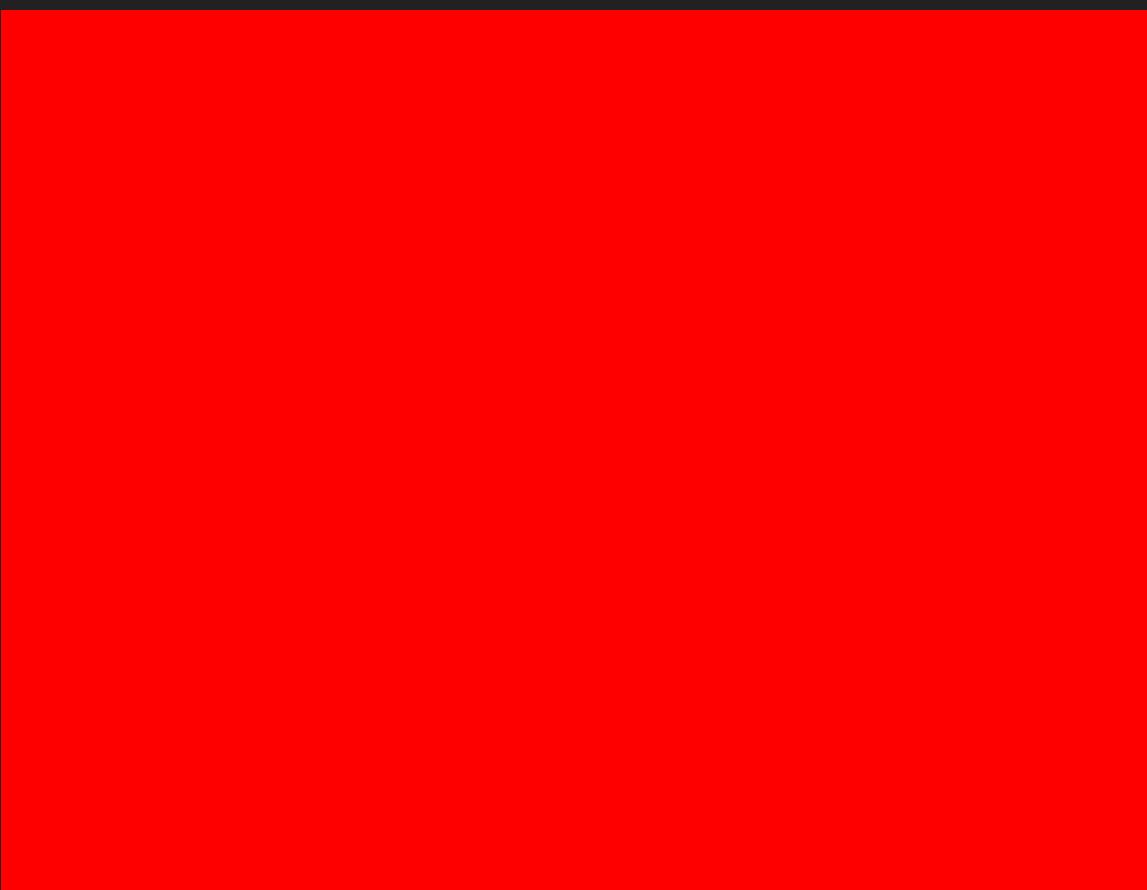
based knowledge

#ff0000

slash @ https://github.com/KvN1027/la_project_image_processing

RGB

based knowledge

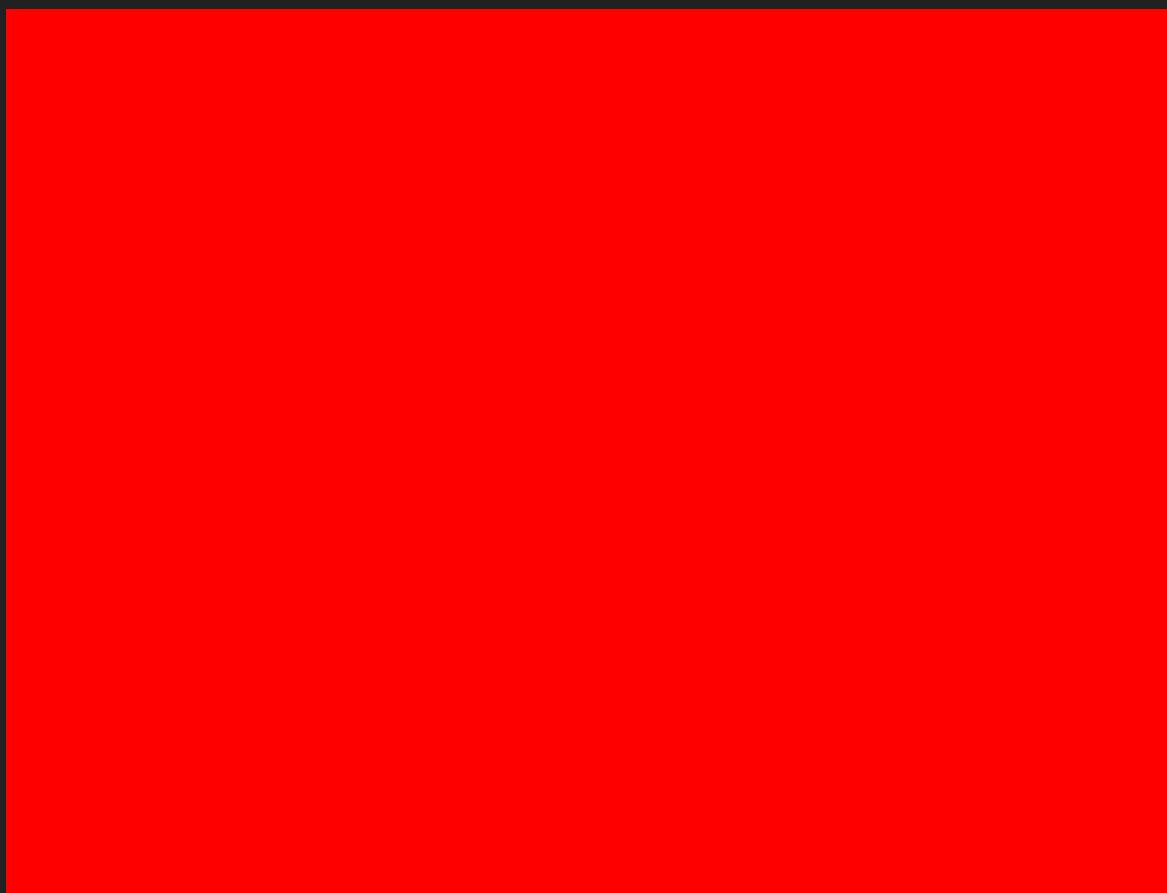


#ff0000

slash @ https://github.com/KvN1027/la_project_image_processing

RGB

based knowledge



#ff0000

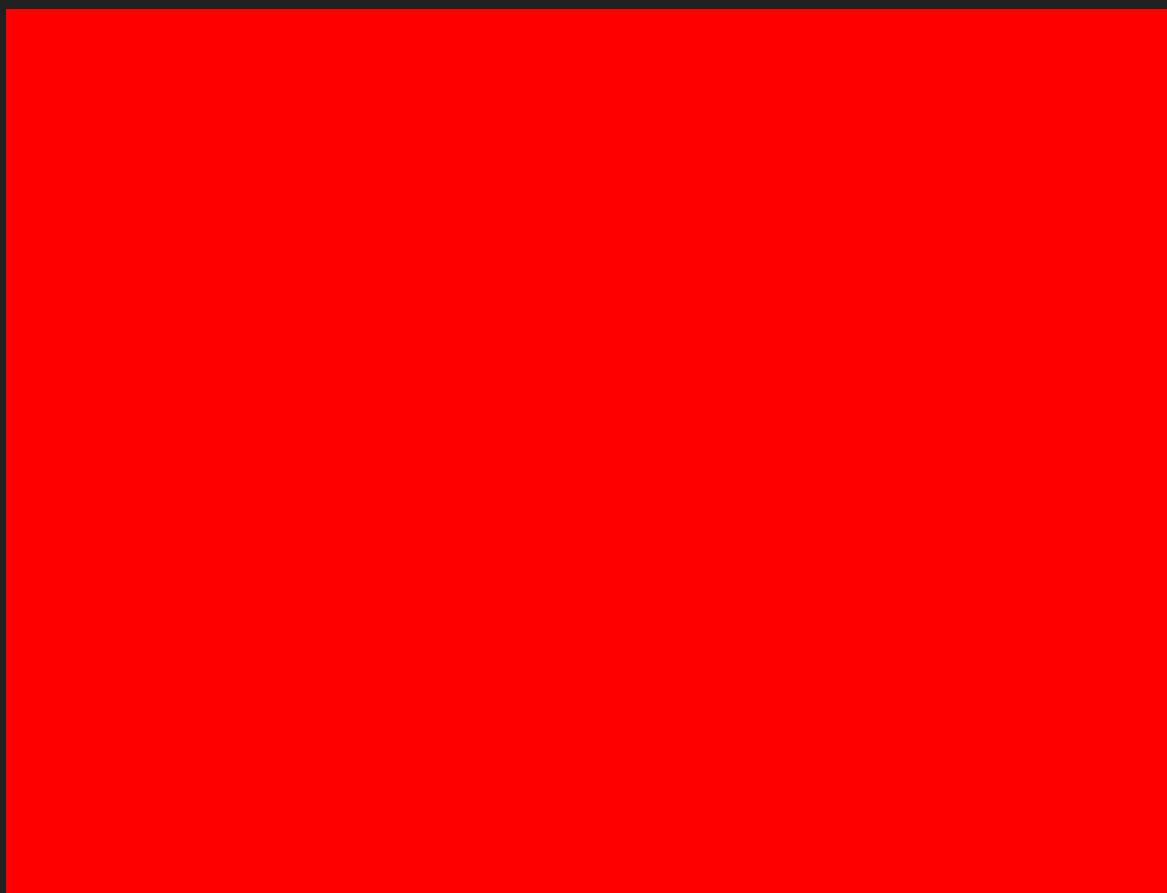


#7b68ee

slash @ https://github.com/KvN1027/la_project_image_processing

RGB

based knowledge



#ff0000



#7b68ee

slash @ https://github.com/KvN1027/la_project_image_processing

RGB

based knowledge

#7b68ee

slash @ https://github.com/KvN1027/la_project_image_processing

RGB

based knowledge

#7b68ee

hex

slash @ https://github.com/KvN1027/la_project_image_processing

RGB

based knowledge

#7b68ee

2 hex digits range=16*16

hex

slash @ https://github.com/KvN1027/la_project_image_processing

RGB

based knowledge

#7b68ee

single color range = 0~255

hex

slash @ https://github.com/KvN1027/la_project_image_processing

RGB

based knowledge

#7b68ee
134 204 238

slash @ https://github.com/KvN1027/la_project_image_processing

RGB

based knowledge

#000000

0

0

0

slash @ https://github.com/KvN1027/la_project_image_processing

RGB

based knowledge

#000000



#fffff

slash @ https://github.com/KvN1027/la_project_image_processing

RGB

based knowledge

#7b68ee

134 204 238

slash @ https://github.com/KvN1027/la_project_image_processing

RGB

based knowledge



#7b68ee

slash @ https://github.com/KvN1027/la_project_image_processing

RGB

based knowledge



slash @ https://github.com/KvN1027/la_project_image_processing

RGB

based knowledge

1080



1920

slash @ https://github.com/KvM027/lap-project_image_processing

RGB

based knowledge

1080

1920



slash @ https://github.com/KvM027/lap-project_image_processing

RGB

based knowledge

RGB

#2C0E43

#2C0E43

#2C0E43

#2C0E43

#2C0E43

#2C0E43

#2C0E43

#2C0E43

#2C0E43

RGB

based knowledge

RGB

#2C

#2C

#2C

C0E43

C0E43

C0E43

Wait a minute...



RGB

based knowledge

#2C0E43

#2C0E43

#2C0E43

#2C0E43

#2C0E43

#2C0E43

#2C0E43

#2C0E43

#2C0E43

slash @ https://github.com/KvN1027/la_project_image_processing

dude that's
matrix



RGB

based knowledge

#2C0E43

#2C0E43

#2C0E43

dude that's
matrix

that's how image process works

#2C0E43

#2C0E43

#2C0E43

slash @ https://github.com/KvN1027/la_project_image_processing



blur

slash @ https://github.com/KvN1027/la_project_image_processing

how to blur

blur

1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	0	0	1	1	1	1
1	1	1	0	0	0	1	1	1	1
1	1	1	1	0	0	1	1	1	1
1	1	1	1	0	0	1	1	1	1
1	1	1	1	0	0	1	1	1	1
1	1	1	1	0	0	1	1	1	1
1	1	1	0	0	0	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1

slash @ https://github.com/KvN1027/la_project_image_processing

how to blur

blur

1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	0	0	1	1	1	1
1	1	1	0	0	0	1	1	1	1
1	1	1	1	0	0	1	1	1	1
1	1	1	1	0	0	1	1	1	1
1	1	1	1	0	0	1	1	1	1
1	1	1	0	0	0	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1

slash @ https://github.com/KvN1027/la_project_image_processing

how to blur

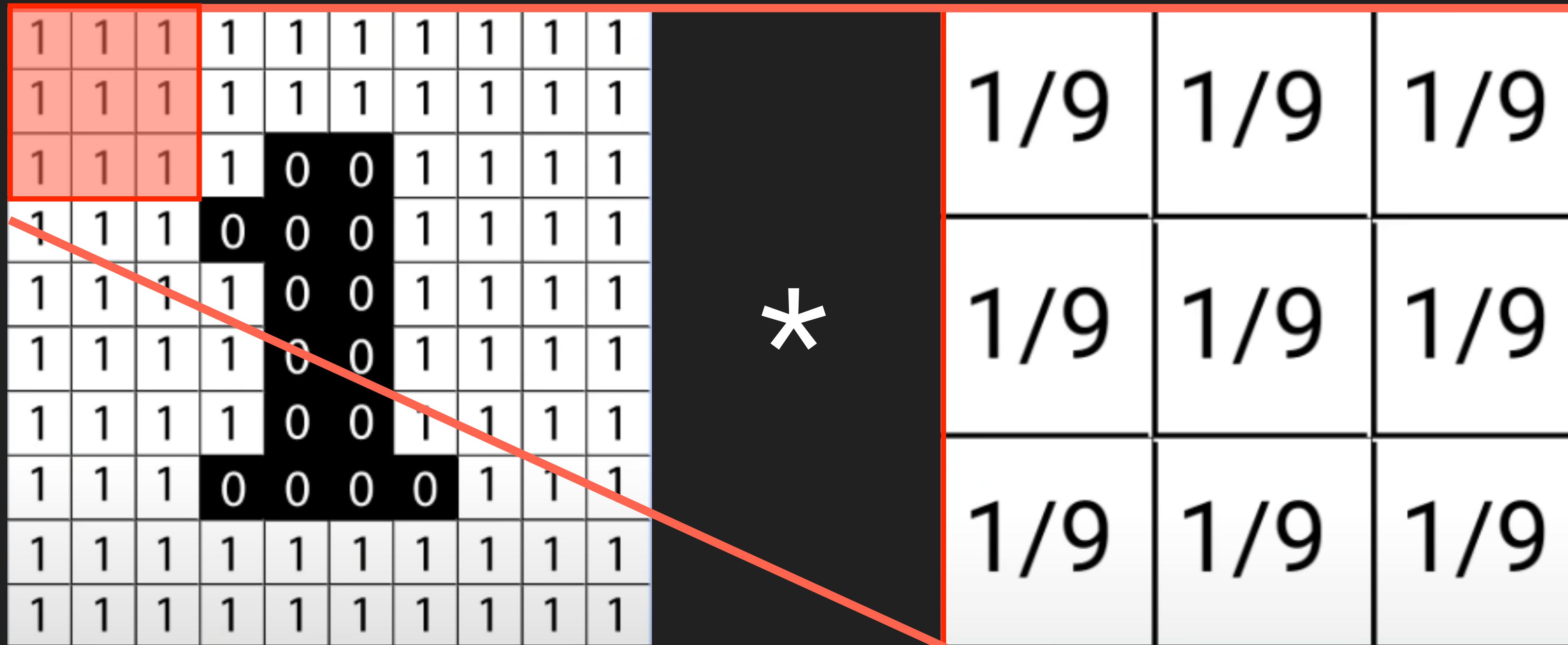
blur

1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	0	0	1	1	1	1	1
1	1	1	0	0	0	1	1	1	1	1
1	1	1	1	0	0	1	1	1	1	1
1	1	1	1	0	0	1	1	1	1	1
1	1	1	1	0	0	1	1	1	1	1
1	1	1	1	0	0	1	1	1	1	1
1	1	1	0	0	0	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1

*

how to blur

blur



how to blur

blur

1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	0	0	1	1	1	1	1
1	1	1	0	0	0	1	1	1	1	1
1	1	1	1	0	0	1	1	1	1	1
1	1	1	1	0	0	1	1	1	1	1
1	1	1	1	0	0	1	1	1	1	1
1	1	1	1	0	0	1	1	1	1	1
1	1	1	0	0	0	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1

*

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

kernel

slash @ https://github.com/KvN1027/la_project_image_processing

how to blur

blur

1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	0	0	1	1	1	1	1
1	1	1	0	0	0	1	1	1	1	1
1	1	1	1	0	0	1	1	1	1	1
1	1	1	1	0	0	1	1	1	1	1
1	1	1	1	0	0	1	1	1	1	1
1	1	1	1	0	0	1	1	1	1	1
1	1	1	0	0	0	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1

*

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

kernel

slash @ https://github.com/KvN1027/la_project_image_processing

how to blur

blur

1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	0	0	1	1	1	1
1	1	1	0	0	0	1	1	1	1
1	1	1	1	0	0	1	1	1	1
1	1	1	1	0	0	1	1	1	1
1	1	1	1	0	0	1	1	1	1
1	1	1	1	0	0	1	1	1	1
1	1	1	0	0	0	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1

$$\begin{matrix} & \begin{matrix} 1/9 & 1/9 & 1/9 \end{matrix} \\ * & \begin{matrix} 1/9 & 1/9 & 1/9 \end{matrix} \\ & \begin{matrix} 1/9 & 1/9 & 1/9 \end{matrix} \end{matrix} = \begin{matrix} & \begin{matrix} & & & & & & & & & & & \end{matrix} \\ & \begin{matrix} & & & & & & & & & & & \end{matrix} \\ & \begin{matrix} & & & & & & & & & & & \end{matrix} \\ & \begin{matrix} & & & & & & & & & & & \end{matrix} \\ & \begin{matrix} & & & & & & & & & & & \end{matrix} \\ & \begin{matrix} & & & & & & & & & & & \end{matrix} \\ & \begin{matrix} & & & & & & & & & & & \end{matrix} \\ & \begin{matrix} & & & & & & & & & & & \end{matrix} \\ & \begin{matrix} & & & & & & & & & & & \end{matrix} \\ & \begin{matrix} & & & & & & & & & & & \end{matrix} \end{matrix}$$

kernel

slash @ https://github.com/KvN1027/la_project_image_processing

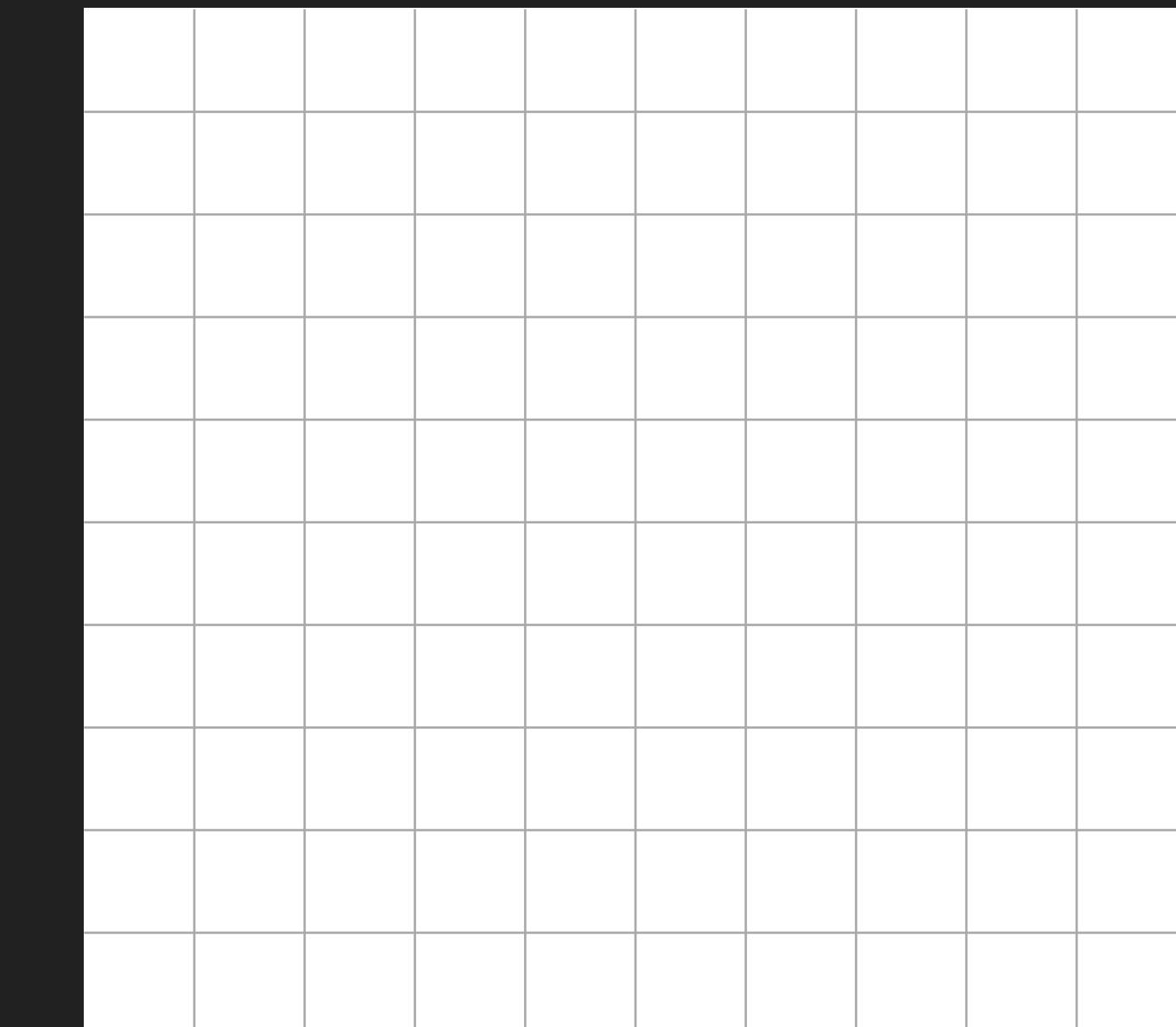
blur image

how to blur

blur

1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	0	0	1	1	1	1	1
1	1	1	0	0	0	1	1	1	1	1
1	1	1	1	0	0	1	1	1	1	1
1	1	1	1	0	0	1	1	1	1	1
1	1	1	1	0	0	1	1	1	1	1
1	1	1	1	0	0	1	1	1	1	1
1	1	1	0	0	0	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1

$$\begin{matrix} & \begin{matrix} 1/9 & 1/9 & 1/9 \end{matrix} \\ * & \begin{matrix} 1/9 & 1/9 & 1/9 \end{matrix} \\ & \begin{matrix} 1/9 & 1/9 & 1/9 \end{matrix} \end{matrix} =$$



kernel

blur image

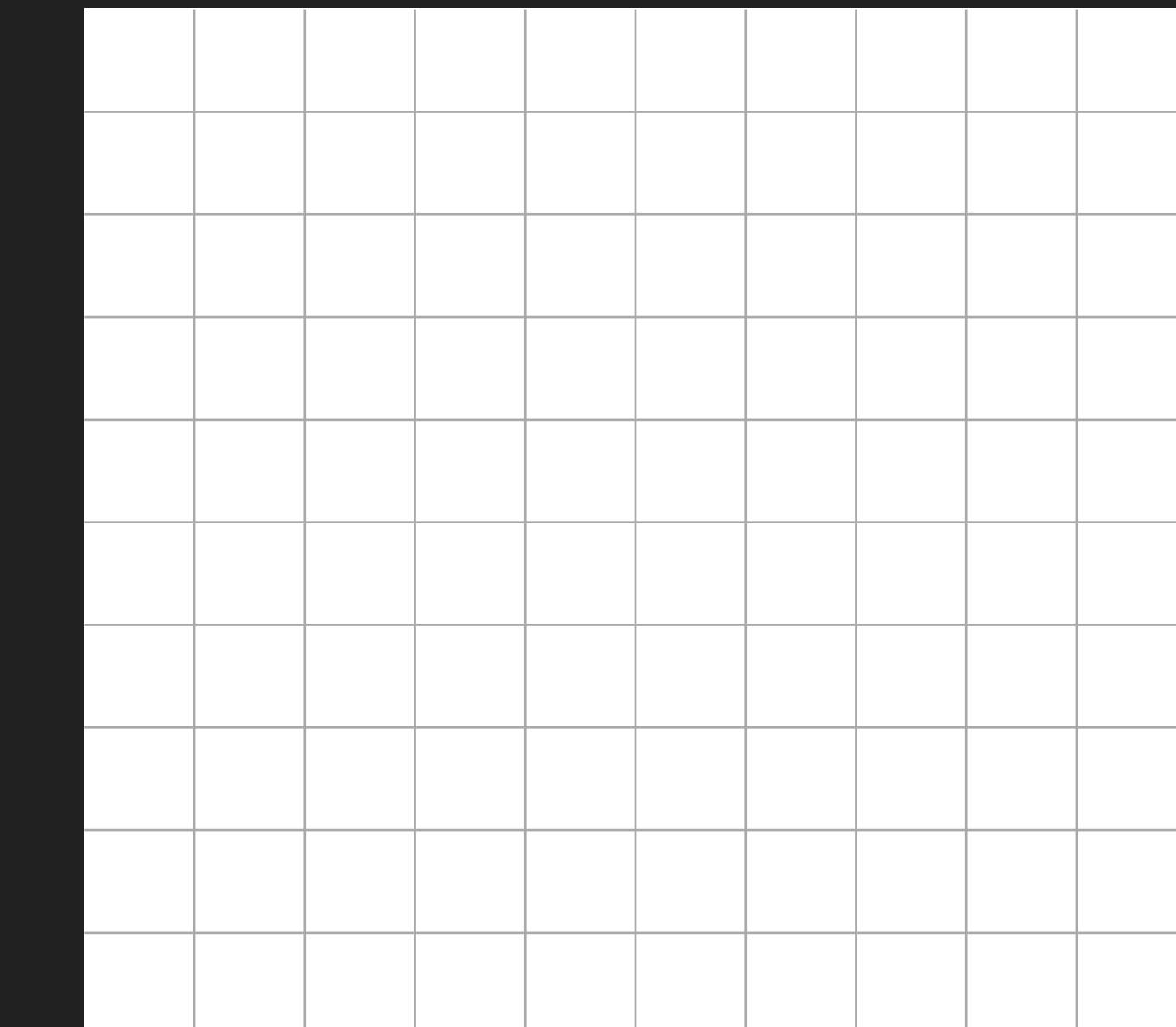
slash @ https://github.com/KvN1027/la_project_image_processing

how to blur

blur

1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	0	0	1	1	1	1	1
1	1	1	0	0	0	1	1	1	1	1
1	1	1	1	0	0	1	1	1	1	1
1	1	1	1	0	0	1	1	1	1	1
1	1	1	1	0	0	1	1	1	1	1
1	1	1	1	0	0	1	1	1	1	1
1	1	1	0	0	0	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1

$$\begin{matrix} & \begin{matrix} 1/9 & 1/9 & 1/9 \end{matrix} \\ * & \begin{matrix} 1/9 & 1/9 & 1/9 \end{matrix} \\ & \begin{matrix} 1/9 & 1/9 & 1/9 \end{matrix} \end{matrix} =$$



kernel

slash @ https://github.com/KvN1027/la_project_image_processing

blur image

how to blur

blur

1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	0	0	1	1	1	1	1
1	1	1	0	0	0	1	1	1	1	1
1	1	1	1	0	0	1	1	1	1	1
1	1	1	1	0	0	1	1	1	1	1
1	1	1	1	0	0	1	1	1	1	1
1	1	1	1	0	0	1	1	1	1	1
1	1	1	0	0	0	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1

$$\begin{matrix} & \begin{matrix} 1/9 & 1/9 & 1/9 \end{matrix} \\ * & \begin{matrix} 1/9 & 1/9 & 1/9 \end{matrix} \\ & \begin{matrix} 1/9 & 1/9 & 1/9 \end{matrix} \end{matrix} = \begin{matrix} & \begin{matrix} & & & & & & & & & & \end{matrix} \\ & \begin{matrix} & & & & & & & & & & \end{matrix} \\ & \begin{matrix} & & & & & & & & & & \end{matrix} \\ & \begin{matrix} & & & & & & & & & & \end{matrix} \\ & \begin{matrix} & & & & & & & & & & \end{matrix} \\ & \begin{matrix} & & & & & & & & & & \end{matrix} \\ & \begin{matrix} & & & & & & & & & & \end{matrix} \\ & \begin{matrix} & & & & & & & & & & \end{matrix} \\ & \begin{matrix} & & & & & & & & & & \end{matrix} \\ & \begin{matrix} & & & & & & & & & & \end{matrix} \end{matrix}$$

kernel

slash @ https://github.com/KvN1027/la_project_image_processing

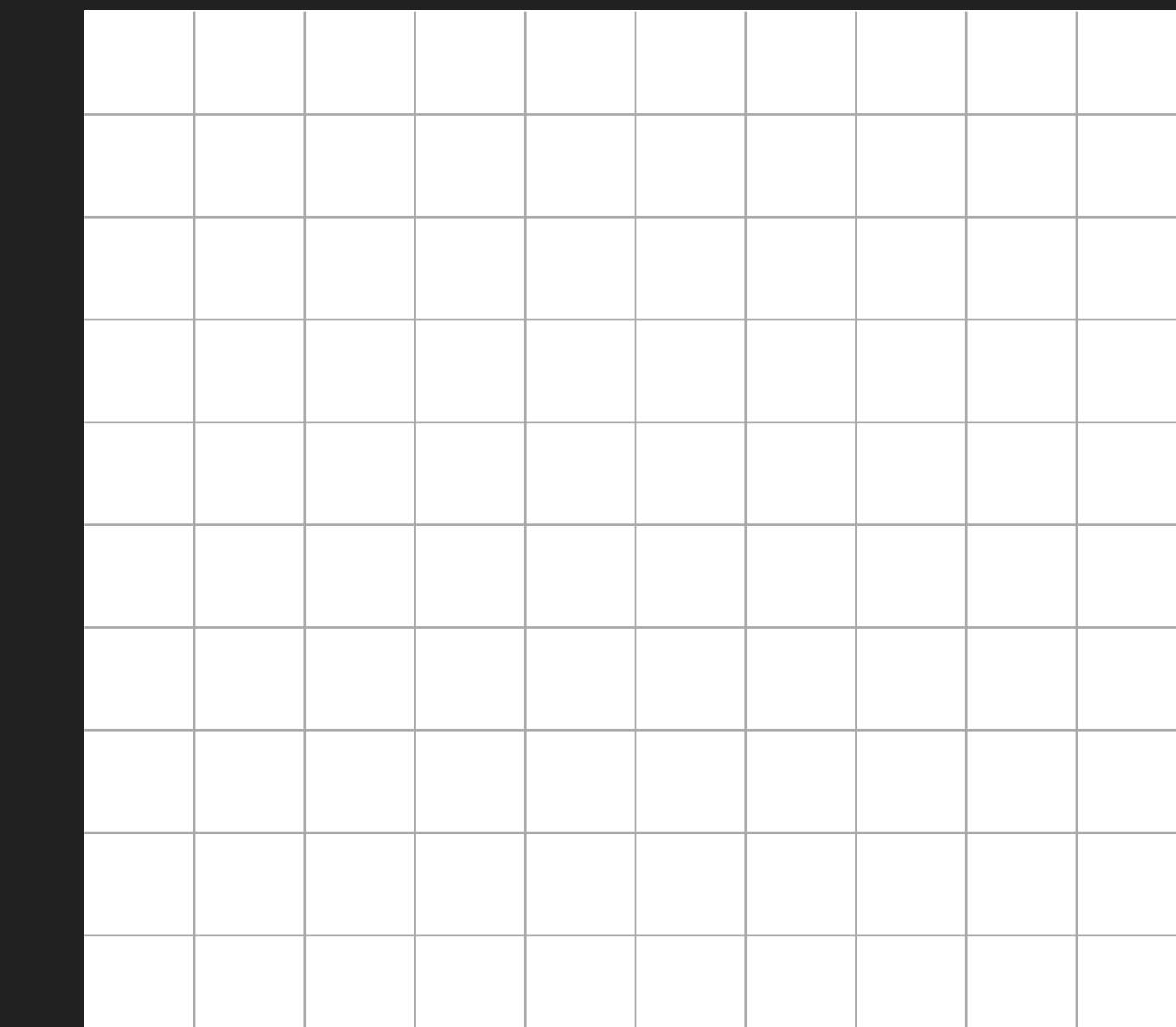
blur image

how to blur

blur

1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	0	0	1	1	1	1	1
1	1	1	0	0	0	1	1	1	1	1
1	1	1	1	0	0	1	1	1	1	1
1	1	1	1	0	0	1	1	1	1	1
1	1	1	1	0	0	1	1	1	1	1
1	1	1	1	0	0	1	1	1	1	1
1	1	1	0	0	0	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1

$$\begin{matrix} & \begin{matrix} 1/9 & 1/9 & 1/9 \end{matrix} \\ * & \begin{matrix} 1/9 & 1/9 & 1/9 \end{matrix} \\ & \begin{matrix} 1/9 & 1/9 & 1/9 \end{matrix} \end{matrix} =$$



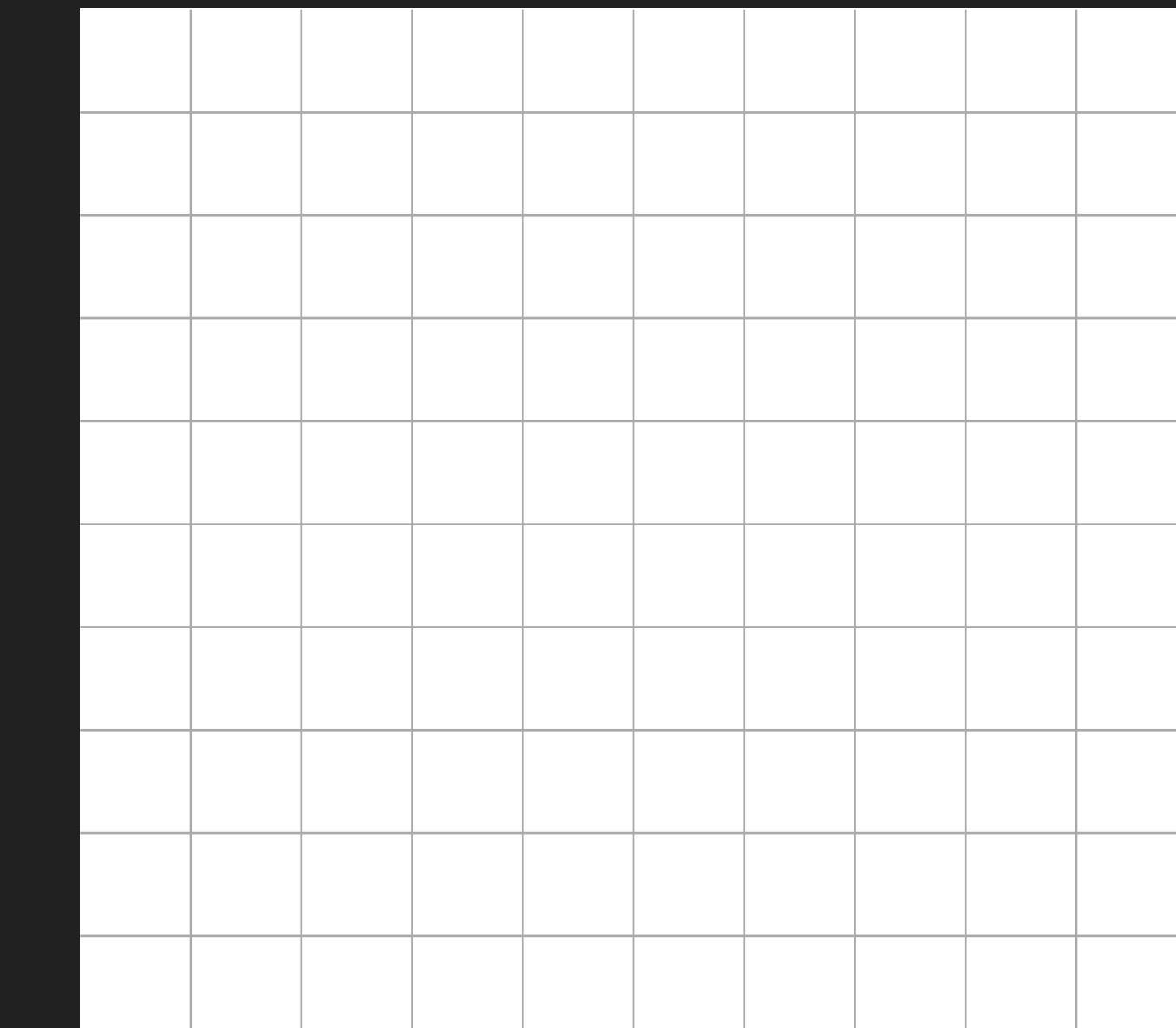
kernel

blur image

slash @ https://github.com/KvN1027/la_project_image_processing

how to blur blur

	$1/9$	$1/9$	$1/9$
*	$1/9$	$1/9$	$1/9$
	$1/9$	$1/9$	$1/9$



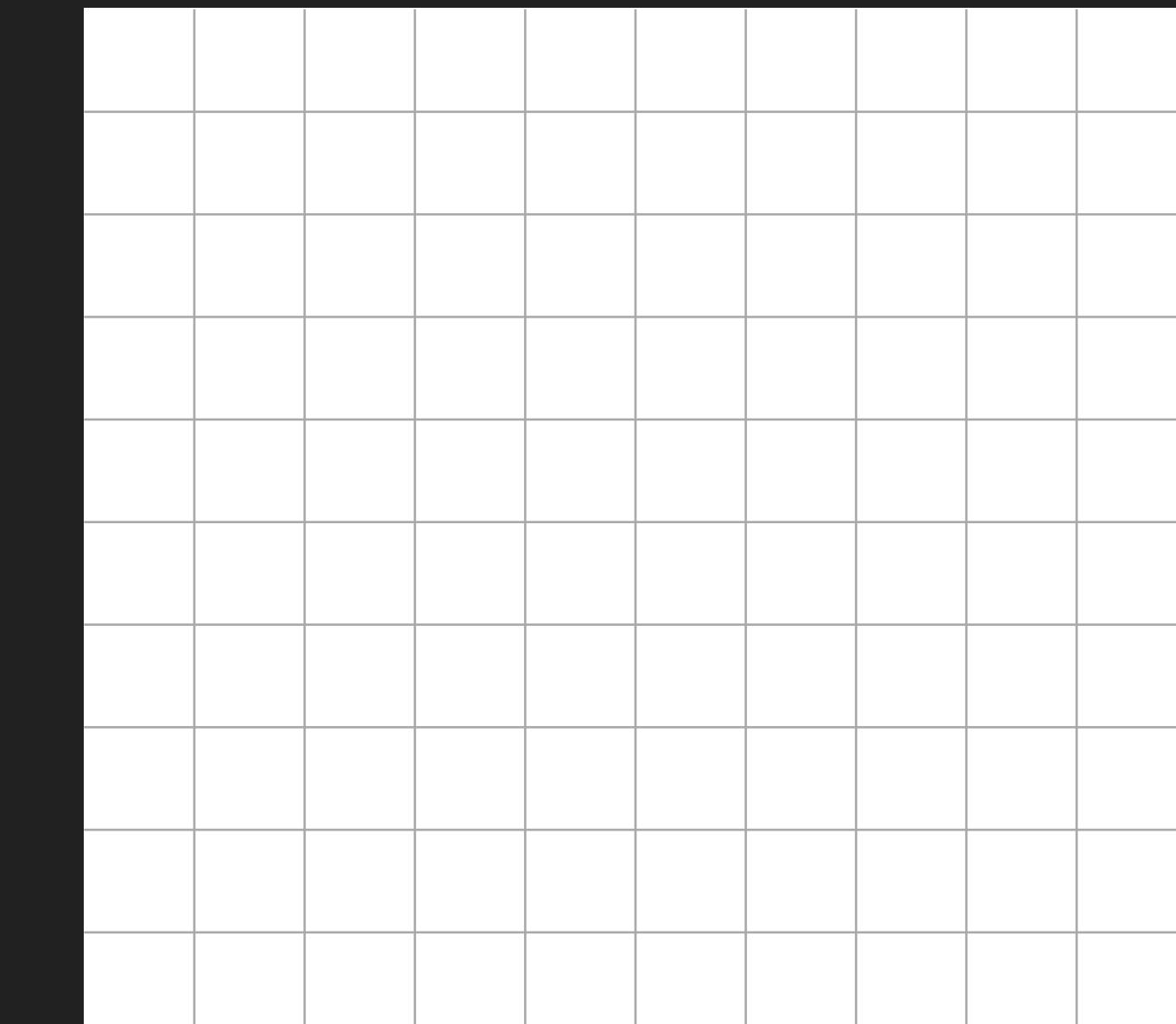
kernel blur image

how to blur

blur

1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	0	0	1	1	1	1	1
1	1	1	0	0	0	1	1	1	1	1
1	1	1	1	0	0	1	1	1	1	1
1	1	1	1	0	0	1	1	1	1	1
1	1	1	1	0	0	1	1	1	1	1
1	1	1	1	0	0	1	1	1	1	1
1	1	1	0	0	0	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1

$$\begin{matrix} & \begin{matrix} 1/9 & 1/9 & 1/9 \end{matrix} \\ * & \begin{matrix} 1/9 & 1/9 & 1/9 \end{matrix} \\ & \begin{matrix} 1/9 & 1/9 & 1/9 \end{matrix} \end{matrix} =$$



kernel

blur image

slash @ https://github.com/KvN1027/la_project_image_processing

how to blur

blur

1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	0	0	1	1	1	1	1
1	1	1	0	0	0	1	1	1	1	1
1	1	1	1	0	0	1	1	1	1	1
1	1	1	1	0	0	1	1	1	1	1
1	1	1	1	0	0	1	1	1	1	1
1	1	1	1	0	0	1	1	1	1	1
1	1	1	0	0	0	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1

$$\begin{matrix} & \begin{matrix} 1/9 & 1/9 & 1/9 \end{matrix} \\ * & \begin{matrix} 1/9 & 1/9 & 1/9 \end{matrix} \\ & \begin{matrix} 1/9 & 1/9 & 1/9 \end{matrix} \end{matrix} = \begin{matrix} & \begin{matrix} & & & & & & & & & & \end{matrix} \\ & \begin{matrix} & & & & & & & & & & \end{matrix} \\ & \begin{matrix} & & & & & & & & & & \end{matrix} \\ & \begin{matrix} & & & & & & & & & & \end{matrix} \\ & \begin{matrix} & & & & & & & & & & \end{matrix} \\ & \begin{matrix} & & & & & & & & & & \end{matrix} \\ & \begin{matrix} & & & & & & & & & & \end{matrix} \\ & \begin{matrix} & & & & & & & & & & \end{matrix} \\ & \begin{matrix} & & & & & & & & & & \end{matrix} \\ & \begin{matrix} & & & & & & & & & & \end{matrix} \end{matrix}$$

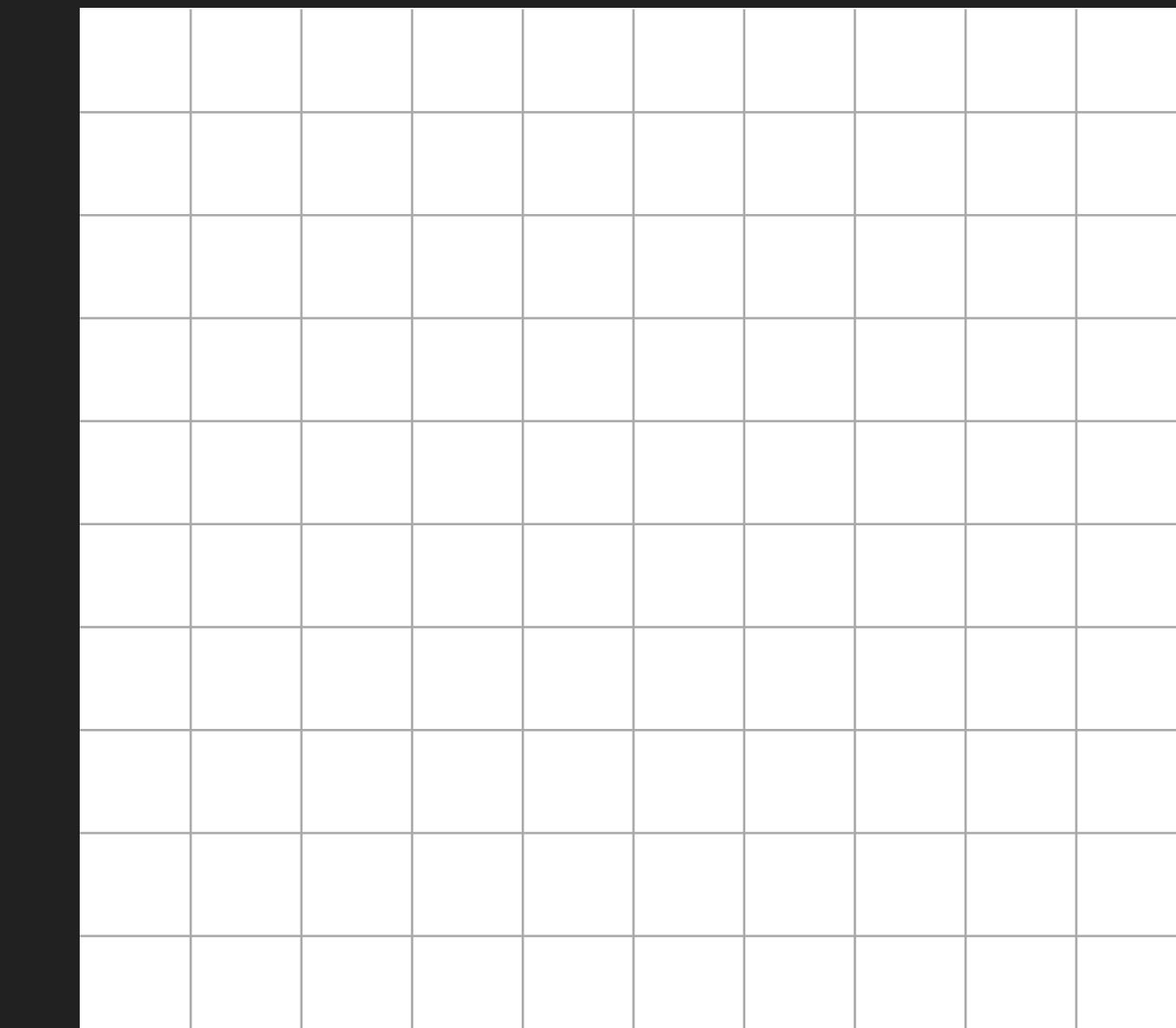
kernel

slash @ https://github.com/KvN1027/la_project_image_processing

blur image

how to blur blur

	$1/9$	$1/9$	$1/9$
*	$1/9$	$1/9$	$1/9$
	$1/9$	$1/9$	$1/9$



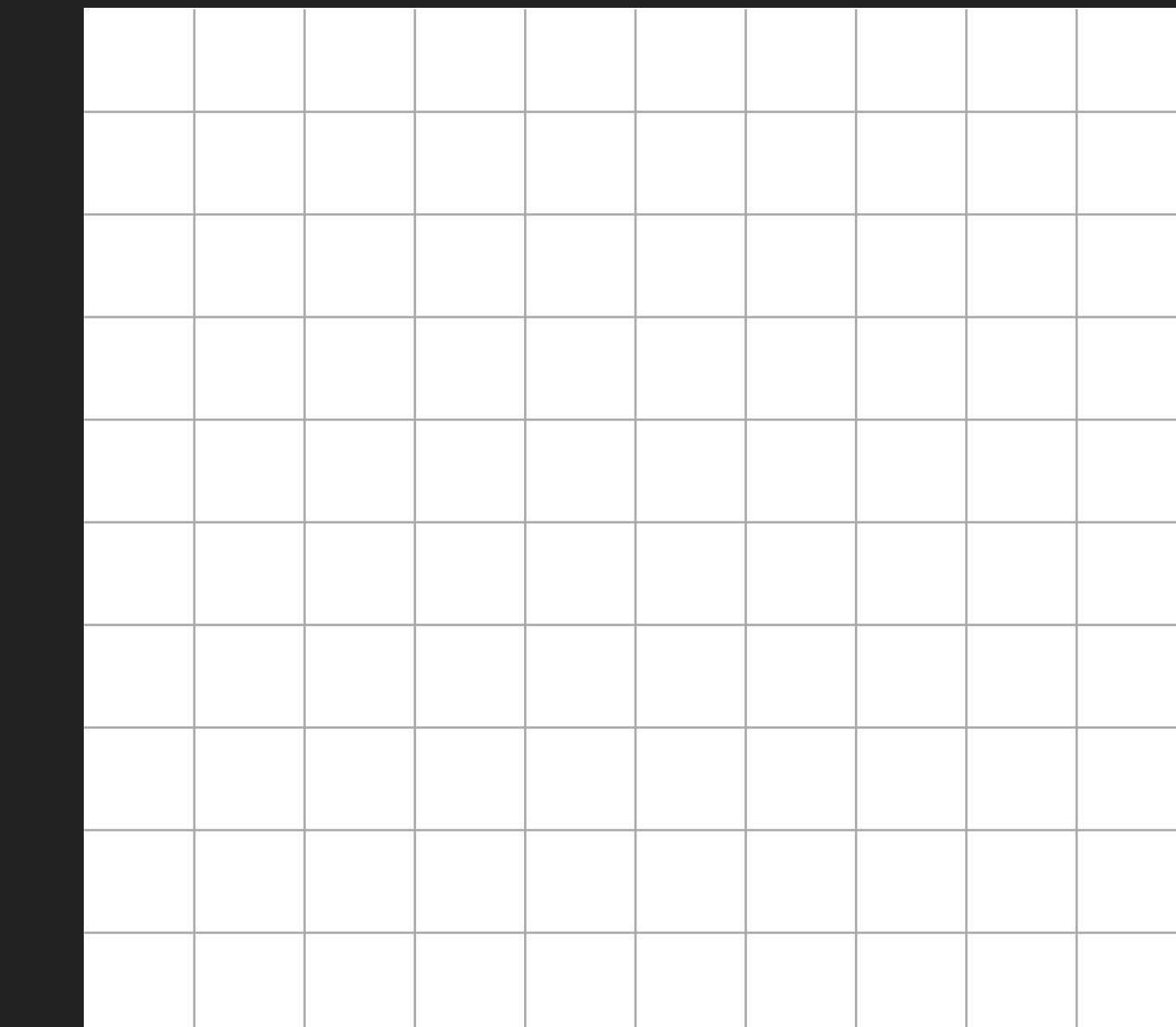
kernel blur image

how to blur

blur

1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	0	0	1	1	1	1	1
1	1	1	0	0	0	1	1	1	1	1
1	1	1	1	0	0	1	1	1	1	1
1	1	1	1	0	0	1	1	1	1	1
1	1	1	1	0	0	1	1	1	1	1
1	1	1	1	0	0	1	1	1	1	1
1	1	1	0	0	0	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1

$$\begin{matrix} & \begin{matrix} 1/9 & 1/9 & 1/9 \end{matrix} \\ * & \begin{matrix} 1/9 & 1/9 & 1/9 \end{matrix} \\ & \begin{matrix} 1/9 & 1/9 & 1/9 \end{matrix} \end{matrix} =$$



find the average of the pixel values

slash @ https://github.com/KvN1027/la_project_image_processing

blur image

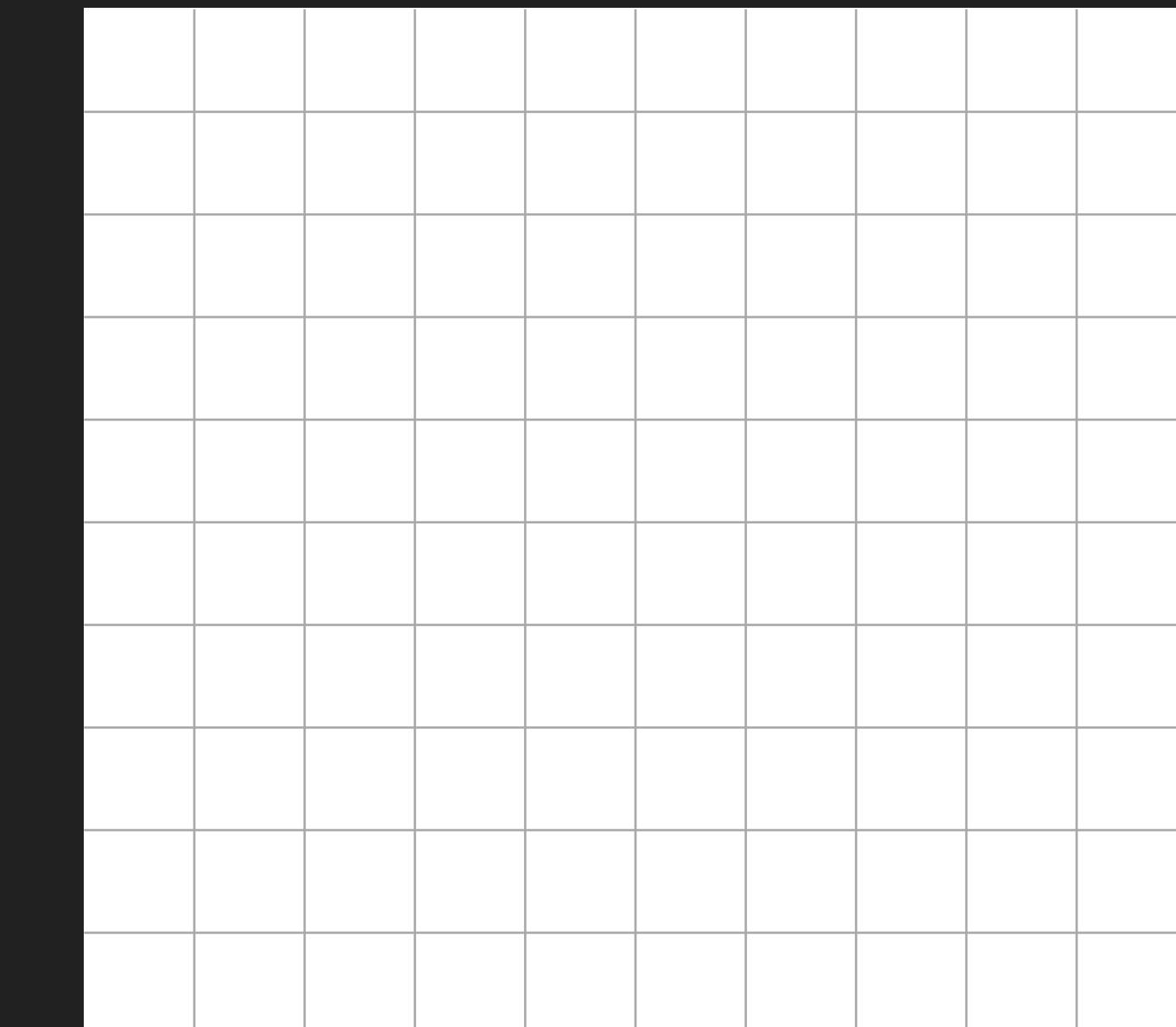
how to blur

blur

avg: 1

1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	0	0	1	1	1	1
1	1	1	0	0	0	1	1	1	1
1	1	1	1	0	0	1	1	1	1
1	1	1	1	0	0	1	1	1	1
1	1	1	1	0	0	1	1	1	1
1	1	1	1	0	0	1	1	1	1
1	1	1	0	0	0	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1

$$\begin{matrix} & \begin{matrix} 1/9 & 1/9 & 1/9 \end{matrix} \\ * & \begin{matrix} 1/9 & 1/9 & 1/9 \end{matrix} \\ & \begin{matrix} 1/9 & 1/9 & 1/9 \end{matrix} \end{matrix} =$$



kernel

slash @ https://github.com/KvN1027/la_project_image_processing

blur image

how to blur

blur

avg: 1

1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	0	0	1	1	1	1
1	1	1	0	0	0	1	1	1	1
1	1	1	1	0	0	1	1	1	1
1	1	1	1	0	0	1	1	1	1
1	1	1	1	0	0	1	1	1	1
1	1	1	1	0	0	1	1	1	1
1	1	1	0	0	0	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1

$$\begin{matrix} & \begin{matrix} 1/9 & 1/9 & 1/9 \end{matrix} \\ * & \begin{matrix} 1/9 & 1/9 & 1/9 \end{matrix} \\ & \begin{matrix} 1/9 & 1/9 & 1/9 \end{matrix} \end{matrix} = \begin{matrix} & \begin{matrix} 1 \end{matrix} \end{matrix}$$

kernel

slash @ https://github.com/KvN1027/la_project_image_processing

blur image

how to blur

blur

1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	0	0	1	1	1	1	1	1
1	1	1	0	0	0	1	1	1	1	1	1
1	1	1	1	0	0	1	1	1	1	1	1
1	1	1	1	0	0	1	1	1	1	1	1
1	1	1	1	0	0	1	1	1	1	1	1
1	1	1	1	0	0	1	1	1	1	1	1
1	1	1	0	0	0	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1

$$\begin{matrix} & 1/9 & 1/9 & 1/9 \\ * & \hline & 1/9 & 1/9 & 1/9 \\ & 1/9 & 1/9 & 1/9 \end{matrix}$$

1	1										

kernel

blur image

slash @ https://github.com/KvN1027/la_project_image_processing

how to blur

blur

1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	0	0	1	1	1	1	1
1	1	1	0	0	0	1	1	1	1	1
1	1	1	1	0	0	1	1	1	1	1
1	1	1	1	0	0	1	1	1	1	1
1	1	1	1	0	0	1	1	1	1	1
1	1	1	1	0	0	1	1	1	1	1
1	1	1	0	0	0	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1

$$\begin{matrix} & \begin{matrix} 1/9 & 1/9 & 1/9 \end{matrix} \\ * & \begin{matrix} 1/9 & 1/9 & 1/9 \end{matrix} \\ & \begin{matrix} 1/9 & 1/9 & 1/9 \end{matrix} \end{matrix} =$$

1	1	0.89								

kernel

slash @ https://github.com/KvN1027/la_project_image_processing

blur image

how to blur

blur

1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	0	0	1	1	1	1	1
1	1	1	0	0	0	1	1	1	1	1
1	1	1	1	0	0	1	1	1	1	1
1	1	1	1	1	0	0	1	1	1	1
1	1	1	1	1	0	0	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1

$$\begin{matrix} & 1/9 & 1/9 & 1/9 \\ * & \hline & 1/9 & 1/9 & 1/9 \\ & 1/9 & 1/9 & 1/9 \end{matrix}$$

1	1	0.89	0.78							

kernel

slash @ https://github.com/KvN1027/la_project_image_processing

blur image

how to blur

blur

1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	0	0	1	1	1	1	1	1
1	1	1	0	0	0	1	1	1	1	1	1
1	1	1	1	0	0	1	1	1	1	1	1
1	1	1	1	0	0	1	1	1	1	1	1
1	1	1	1	0	0	1	1	1	1	1	1
1	1	1	1	0	0	0	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1

$$\begin{matrix} & 1/9 & 1/9 & 1/9 \\ * & \hline & 1/9 & 1/9 & 1/9 \\ & 1/9 & 1/9 & 1/9 \end{matrix}$$

1	1	0.89	0.78	0.78							

kernel

blur image

slash @ https://github.com/KvN1027/la_project_image_processing

how to blur

blur

1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	0	0	1	1	1	1
1	1	1	0	0	0	1	1	1	1
1	1	1	1	0	0	1	1	1	1
1	1	1	1	0	0	1	1	1	1
1	1	1	1	0	0	1	1	1	1
1	1	1	1	0	0	1	1	1	1
1	1	1	0	0	0	0	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1

$$\begin{matrix} & 1/9 & 1/9 & 1/9 \\ * & \hline & 1/9 & 1/9 & 1/9 \\ & 1/9 & 1/9 & 1/9 \end{matrix}$$

1	1	0.89	0.78	0.78	0.89				

kernel

slash @ https://github.com/KvN1027/la_project_image_processing

blur image

how to blur

blur

1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	0	0	1	1	1	1
1	1	1	1	0	0	0	1	1	1	1
1	1	1	1	0	0	0	1	1	1	1
1	1	1	1	1	0	0	1	1	1	1
1	1	1	1	1	0	0	1	1	1	1
1	1	1	1	1	0	0	1	1	1	1
1	1	1	1	1	0	0	0	0	1	1
1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1

$$\begin{matrix} & \begin{matrix} 1/9 & 1/9 & 1/9 \end{matrix} \\ * & \begin{matrix} 1/9 & 1/9 & 1/9 \end{matrix} \\ & \begin{matrix} 1/9 & 1/9 & 1/9 \end{matrix} \end{matrix} =$$

1	1	0.89	0.78	0.78	0.89

kernel

blur image

slash @ https://github.com/KvN1027/la_project_image_processing

how to blur

blur

1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	0	0	1	1	1	1
1	1	1	1	0	0	0	1	1	1	1
1	1	1	1	0	0	0	1	1	1	1
1	1	1	1	0	0	0	1	1	1	1
1	1	1	1	0	0	0	1	1	1	1
1	1	1	1	0	0	0	1	1	1	1
1	1	1	1	0	0	0	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1

$$\begin{matrix} & \begin{matrix} 1/9 & 1/9 & 1/9 \end{matrix} \\ * & \begin{matrix} 1/9 & 1/9 & 1/9 \end{matrix} \\ & \begin{matrix} 1/9 & 1/9 & 1/9 \end{matrix} \end{matrix} = \begin{matrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & .89 & .78 & .78 & .89 & 1 & 1 & 1 \\ 1 & 1 & .89 & .67 & .44 & .56 & .78 & 1 & 1 & 1 \\ 1 & 1 & .89 & .56 & .22 & .33 & .67 & 1 & 1 & 1 \\ 1 & 1 & .89 & .56 & .22 & .33 & .67 & 1 & 1 & 1 \\ 1 & 1 & 1 & .67 & .33 & .33 & .67 & 1 & 1 & 1 \\ 1 & 1 & .89 & .56 & .22 & .22 & .56 & .89 & 1 & 1 \\ 1 & 1 & .89 & .67 & .44 & .44 & .67 & .89 & 1 & 1 \\ 1 & 1 & .89 & .78 & .67 & .67 & .78 & .89 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{matrix}$$

kernel

slash @ https://github.com/KvN1027/la_project_image_processing

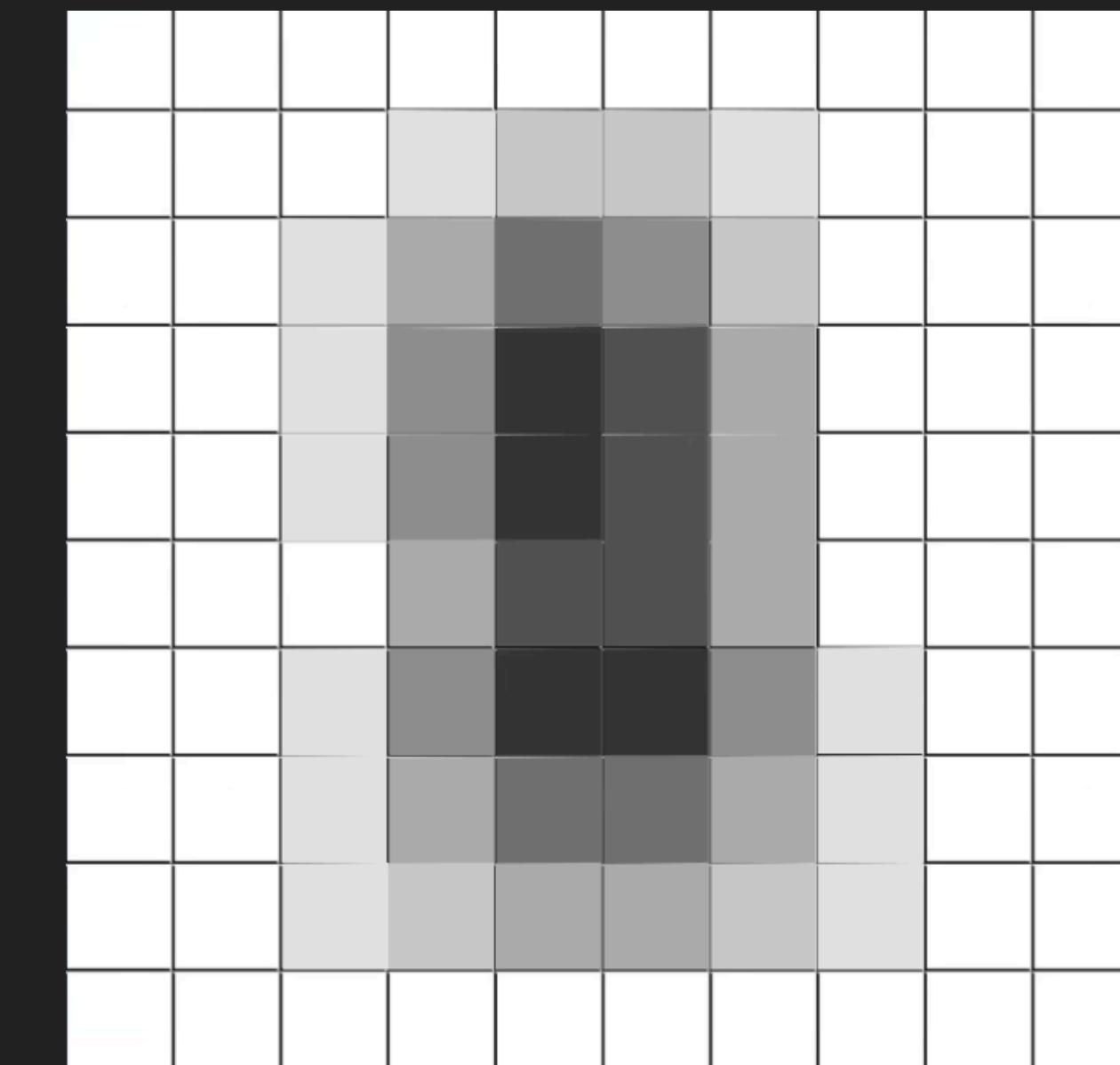
blur image

how to blur

blur

1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	0	0	1	1	1	1
1	1	1	0	0	0	1	1	1	1
1	1	1	1	0	0	1	1	1	1
1	1	1	1	0	0	1	1	1	1
1	1	1	1	0	0	1	1	1	1
1	1	1	1	0	0	1	1	1	1
1	1	1	0	0	0	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1

$$\begin{matrix} & \begin{matrix} 1/9 & 1/9 & 1/9 \end{matrix} \\ * & \begin{matrix} 1/9 & 1/9 & 1/9 \end{matrix} \\ & \begin{matrix} 1/9 & 1/9 & 1/9 \end{matrix} \end{matrix} =$$



kernel

slash @ https://github.com/KvN1027/la_project_image_processing

blur image

how to blur

blur



*

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

=



box blur

how to blur

blur



$$\begin{matrix} * & \begin{matrix} 1/16 & 1/8 & 1/16 \\ 1/8 & 1/4 & 1/8 \\ 1/16 & 1/8 & 1/16 \end{matrix} & = \end{matrix}$$



gaussian blur

slash @ https://github.com/KvN1027/la_project_image_processing

how to blur

blur



box blur



gaussian blur

slash @ https://github.com/KvN1027/la_project_image_processing

gaussian blur

blur

```
11 # Set the size of the Gaussian kernel and standard deviation
12 kernel_size = 50 # A larger kernel size
13 sigma = 10 # A higher standard deviation for more blur
14
15 # Create Gaussian filter
16 ax = np.linspace(-(kernel_size - 1) / 2., (kernel_size - 1) / 2., kernel_size)
17 x, y = np.meshgrid(ax, ax)
18 gaussian_kernel = np.exp(-0.5 * (np.square(x) + np.square(y)) / np.square(sigma))
19 gaussian_kernel = gaussian_kernel / np.sum(gaussian_kernel) # Normalization
20
21 # Read the original image
22 image_path = 'meowmeow.jpg' # Replace with your image path
23 image = mpimg.imread(image_path)
24
25 # Save the original image without axis
26 fig, ax = plt.subplots()
27 ax.imshow(image)
28 ax.axis('off') # Turn off the axis
29 plt.savefig('original_image.png', bbox_inches='tight', pad_inches=0)
30 plt.close(fig) # Close the figure to free up memory
31
32 # Apply Gaussian filter (convolution) to each channel
33 red = signal.convolve2d(image[:, :, 0], gaussian_kernel, boundary='symm', mode='same')
34 green = signal.convolve2d(image[:, :, 1], gaussian_kernel, boundary='symm', mode='same')
35 blue = signal.convolve2d(image[:, :, 2], gaussian_kernel, boundary='symm', mode='same')
36
37 # Stack the channels back into a 3D array
38 blurred_image = np.stack([red, green, blue], axis=2)
39
40 # Ensure the pixel values are properly scaled
41 blurred_image = np.clip(blurred_image, 0, 255).astype(np.uint8)
42
43 # Save the blurred image without axis
44 fig, ax = plt.subplots()
45 ax.imshow(blurred_image)
46 ax.axis('off') # Turn off the axis
47 plt.savefig('enhanced_blurred_image-less.png', bbox_inches='tight', pad_inches=0)
48 plt.close(fig) # Close the figure to free up memory
```

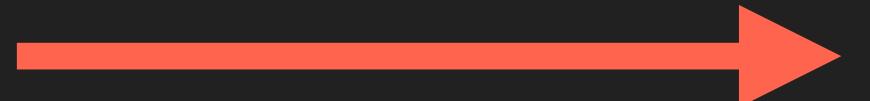
slashed image

edge sketching

slash @ https://github.com/KvN1027/la_project_image_processing

flows

edge sketching



original image

grayscale

slash @ https://github.com/KvN1027/la_project_image_processing

flows

edge sketching



Grayscale Value=0.299×Red+0.587×Green+0.114×Blue



grayscale

slash @ https://github.com/KvN1027/la_project_image_processing

flows

edge sketching



*

-1	-1	-1
-1	8	-1
-1	-1	-1

=



edge detection

slash @ https://github.com/KvN1027/la_project_image_processing

flows

edge sketching

edge detection matrix →

```
11  # Laplacian edge detection kernel
12  laplacian_kernel = np.array([[-1, -1, -1],
13  [-1, 8, -1],
14  [-1, -1, -1]])
15
16  # Read the original image
17  image_path = 'meowmeow.jpg' # Replace with your image path
18  image = mpimg.imread(image_path)
19  if image.ndim == 3 and image.shape[2] == 3:
20      # Convert image to grayscale if it's color
21      image = np.dot(image[...,:3], [0.2989, 0.5870, 0.1140])
22
23  # Apply Laplacian kernel to the grayscale image for edge detection
24  edges = convolve(image, laplacian_kernel)
25
26  # Ensure the pixel values are properly scaled
27  edges = np.clip(edges, 0, 255).astype(np.uint8)
28
29  # Save the edge-detected image without axis
30  fig, ax = plt.subplots()
31  ax.imshow(edges, cmap='gray')
32  ax.axis('off') # Turn off the axis
33  plt.savefig('edge_detected_image.png', bbox_inches='tight', pad_inches=0)
34  plt.close(fig) # Close the figure to free up memory
```

slash @ <https://git>

flows

edge sketching

turn it to grayscale →



```
11 # Laplacian edge detection kernel
12 laplacian_kernel = np.array([[-1, -1, -1],
13                             [-1, 8, -1],
14                             [-1, -1, -1]])
15
16 # Read the original image
17 image_path = 'meowmeow.jpg' # Replace with your image path
18 image = mpimg.imread(image_path)
19 if image.ndim == 3 and image.shape[2] == 3:
20     # Convert image to grayscale if it's color
21     image = np.dot(image[:, :, 3], [0.2989, 0.5870, 0.1140])
22
23 # Apply Laplacian kernel to the grayscale image for edge detection
24 edges = convolve(image, laplacian_kernel)
25
26 # Ensure the pixel values are properly scaled
27 edges = np.clip(edges, 0, 255).astype(np.uint8)
28
29 # Save the edge-detected image without axis
30 fig, ax = plt.subplots()
31 ax.imshow(edges, cmap='gray')
32 ax.axis('off') # Turn off the axis
33 plt.savefig('edge_detected_image.png', bbox_inches='tight', pad_inches=0)
34 plt.close(fig) # Close the figure to free up memory
```

@ <https://git>

flows

edge sketching

apply the kernel →

```
11  # Laplacian edge detection kernel
12  laplacian_kernel = np.array([[-1, -1, -1],
13  [-1, 8, -1],
14  [-1, -1, -1]])
15
16  # Read the original image
17  image_path = 'meowmeow.jpg' # Replace with your image path
18  image = mpimg.imread(image_path)
19  if image.ndim == 3 and image.shape[2] == 3:
20      # Convert image to grayscale if it's color
21      image = np.dot(image[...,:3], [0.2989, 0.5870, 0.1140])
22
23  # Apply Laplacian kernel to the grayscale image for edge detection
24  edges = convolve(image, laplacian_kernel)
25
26  # Ensure the pixel values are properly scaled
27  edges = np.clip(edges, 0, 255).astype(np.uint8)
28
29  # Save the edge-detected image without axis
30  fig, ax = plt.subplots()
31  ax.imshow(edges, cmap='gray')
32  ax.axis('off') # Turn off the axis
33  plt.savefig('edge_detected_image.png', bbox_inches='tight', pad_inches=0)
34  plt.close(fig) # Close the figure to free up memory
```

slash @ <https://git>

flows

edge sketching

final result 



slash @ <https://git>

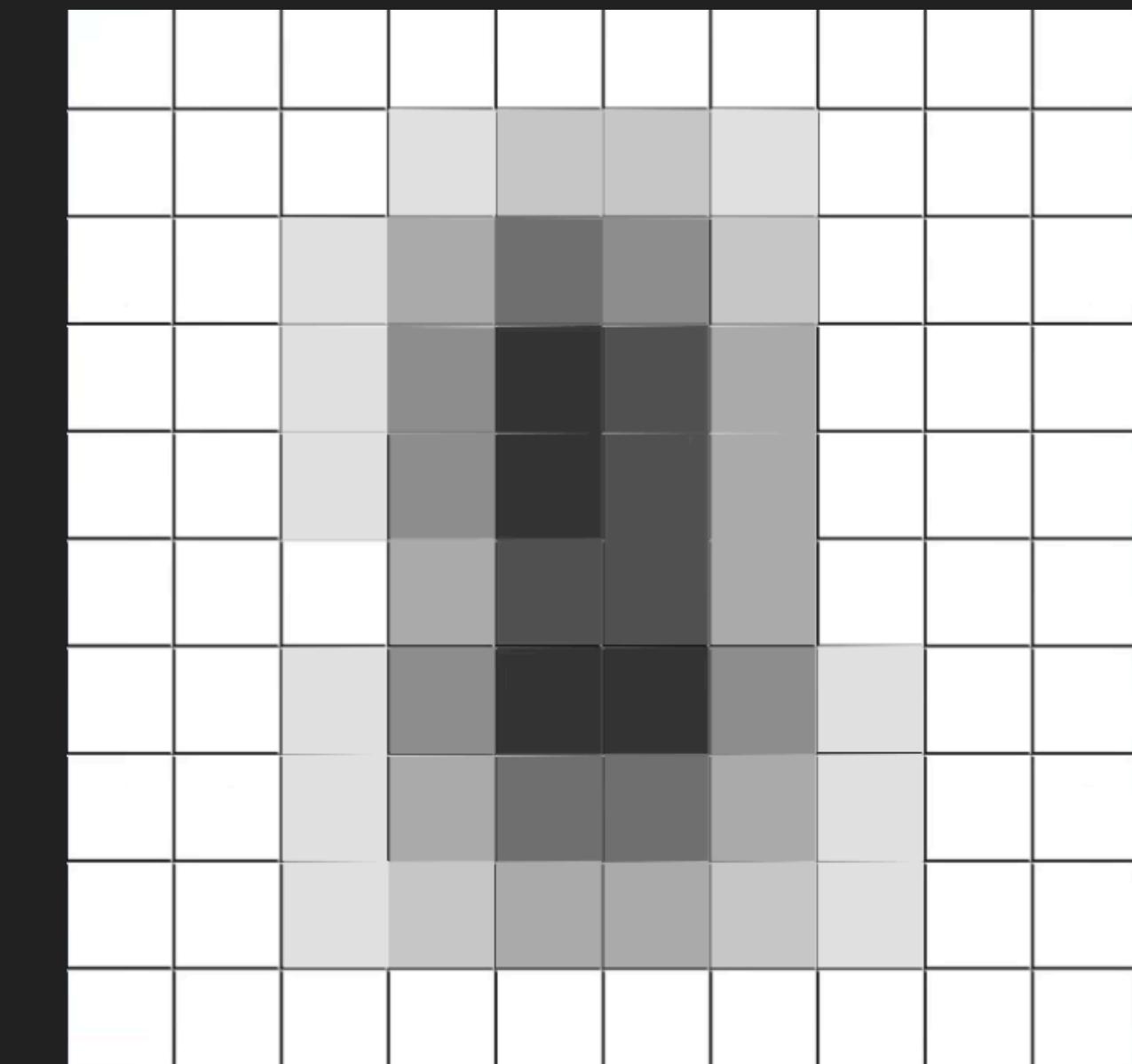
```
11  # Laplacian edge detection kernel
12  laplacian_kernel = np.array([[-1, -1, -1],
13                               [-1, 8, -1],
14                               [-1, -1, -1]])
15
16  # Read the original image
17  image_path = 'meowmeow.jpg' # Replace with your image path
18  image = mpimg.imread(image_path)
19  if image.ndim == 3 and image.shape[2] == 3:
20      # Convert image to grayscale if it's color
21      image = np.dot(image[...,:3], [0.2989, 0.5870, 0.1140])
22
23  # Apply Laplacian kernel to the grayscale image for edge detection
24  edges = convolve(image, laplacian_kernel)
25
26  # Ensure the pixel values are properly scaled
27  edges = np.clip(edges, 0, 255).astype(np.uint8)
28
29  # Save the edge-detected image without axis
30  fig, ax = plt.subplots()
31  ax.imshow(edges, cmap='gray')
32  ax.axis('off') # Turn off the axis
33  plt.savefig('edge_detected_image.png', bbox_inches='tight', pad_inches=0)
34  plt.close(fig) # Close the figure to free up memory
```

note time!
(100% handmade)

slash @ https://github.com/KvN1027/la_project_image_processing

1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	0	0	1	1	1	1
1	1	1	0	0	0	1	1	1	1
1	1	1	1	0	0	1	1	1	1
1	1	1	1	0	0	1	1	1	1
1	1	1	1	0	0	1	1	1	1
1	1	1	1	0	0	1	1	1	1
1	1	1	0	0	0	1	1	1	1
1	1	1	1	1	1	1	1	1	1

*
$$\begin{array}{|c|c|c|} \hline 1/9 & 1/9 & 1/9 \\ \hline \hline 1/9 & 1/9 & 1/9 \\ \hline \hline 1/9 & 1/9 & 1/9 \\ \hline \end{array}$$
 =



kernel

blur image

the original picture is an $n \times n$ matrix P_0 and the one time blurred pic is P_1
 n times blurred pic is P_n

$$\Rightarrow P^1 = APB$$

A is
 row combination
 (vertical blurring)
 B is
 column combination
 (horizontal blurring)

$$A = \frac{1}{3} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} = \frac{1}{3} A_0$$

$$B = \frac{1}{3} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} = \frac{1}{3} A_0$$

$$\text{which } A = B$$

so that

$$P_1 = \left(\frac{1}{3} A_0\right) P_0 \left(\frac{1}{3} A_0\right) = \frac{1}{9} A_0 P_0 A_0$$

$$P_2 = \left(\frac{1}{3} A_0\right) P_1 \left(\frac{1}{3} A_0\right) = \left(\frac{1}{9}\right)^2 A_0^2 P_0 A_0^2$$

⋮
 ⋮
 ⋮

$$\text{so, } P_n = \left(\frac{1}{3} A_0\right) P_{n-1} \left(\frac{1}{3} A_0\right) = \left(\frac{1}{9}\right)^n A_0^n P_0 A_0^n$$

and how to find A_0^n ?

for $n \times n$

$$A_0 = \begin{bmatrix} 1 & & & \\ 1 & 1 & & \\ 1 & 1 & 1 & \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \cdots & 1 \end{bmatrix}_{n \times n}$$

$$= I_{n \times n} + \begin{bmatrix} 0 & 1 & & & \\ 1 & 0 & 1 & & \\ & 1 & 0 & \ddots & \\ & & \ddots & 0 & 1 \\ & & & 1 & 0 \end{bmatrix}_{n \times n}$$

Suppose that $R = \begin{bmatrix} 0 & 1 & 0 & & \\ 1 & 0 & 1 & \ddots & 0 \\ 0 & & \ddots & 0 & 1 \\ & & & 0 & 0 \end{bmatrix}_{n \times n}$

we get $A_0^n = (R+I)^n$

we can use binomial theorem,

$$\begin{aligned} \text{So } A_0^n &= R^n + C_1^n R^{n-1} I + \dots \\ &\quad + C_{n-1}^n R^1 (I)^{n-1} + C_n^n I^n \end{aligned}$$

and that we found that

$$R, R^1, R^2, R^3, \dots, R^n$$

has some special regularity

for example, an $R_{4 \times 4}$ regularity

$$R = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$R^4 = \begin{bmatrix} 2 & 0 & 3 & 0 \\ 0 & 5 & 0 & 3 \\ 3 & 0 & 5 & 0 \\ 0 & 3 & 0 & 2 \end{bmatrix}$$

$$R^2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 2 & 0 & 1 \\ 0 & 0 & 2 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}$$

$$R^5 = \begin{bmatrix} 0 & 5 & 0 & 3 \\ 5 & 0 & 8 & 0 \\ 0 & 0 & 0 & 5 \\ 3 & 0 & 5 & 0 \end{bmatrix}$$

$$R^3 = \begin{bmatrix} 0 & 2 & 0 & 1 \\ 2 & 0 & 3 & 0 \\ 0 & 3 & 0 & 2 \\ 1 & 0 & 2 & 0 \end{bmatrix}$$

$$R^6 = \begin{bmatrix} 5 & 0 & 8 & 0 \\ 0 & 13 & 0 & 8 \\ 8 & 0 & 13 & 0 \\ 0 & 8 & 0 & 5 \end{bmatrix}$$

$$\begin{matrix} 1 & 1 & 2 & 2 & 5 & 5 \\ 0 & 1 & 1 & 3 & 3 & 8 \end{matrix} \dots$$

!

how to blur the picture but with less distortion

→ put more weight on the center pixel

for example:

Gaussian filter matrix

$$\frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

after Gaussian blurred n times, (P_0 turns into P_n)

$$P_n = \left(\frac{1}{16}\right)^n A^n P_0 A^n, \text{ where } A = \begin{bmatrix} \frac{1}{16} & \frac{1}{8} & \frac{1}{16} & \dots & \frac{1}{16} \\ \frac{1}{8} & \frac{1}{4} & \frac{1}{8} & \dots & \frac{1}{8} \\ \frac{1}{16} & \frac{1}{8} & \frac{1}{4} & \dots & \frac{1}{8} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{1}{16} & \frac{1}{8} & \frac{1}{8} & \dots & \frac{1}{16} \end{bmatrix}$$

conclusion

(We got through it!)

slash @ https://github.com/KvN1027/la_project_image_processing

Matrix !

conclusion

- We did
 - Kernel Generation
 - Convolution Operation
 - Real-Time Preview (On Your Phone!)

slash @ https://github.com/KvN1027/la_project_image_processing

Matrix !

conclusion

The screenshot shows a GitHub repository page for 'la_project_image_processing'. The repository is public and has 1 branch and 0 tags. The main branch contains 4 commits from user 'KvN1027' made 13 hours ago. The commits are:

- examples: hope we can get 100 ;;
- templates: hope we can get 100 ;;
- Dockerfile: hope we can get 100 ;;
- README.md: Update README.md

The README file contains the following content:

image processing with linear algebra

NYCU 515504:Linear Algebra project by group 14 image processing by using matrix multiplication, addition and subtraction

The repository has 0 stars, 1 watching, and 0 forks. It also has 0 releases published and no packages published. The languages used are Python (59.1%), HTML (39.6%), and Dockerfile (1.3%).

slash @ https://github.com/KvN1027/la_project_image_processing