

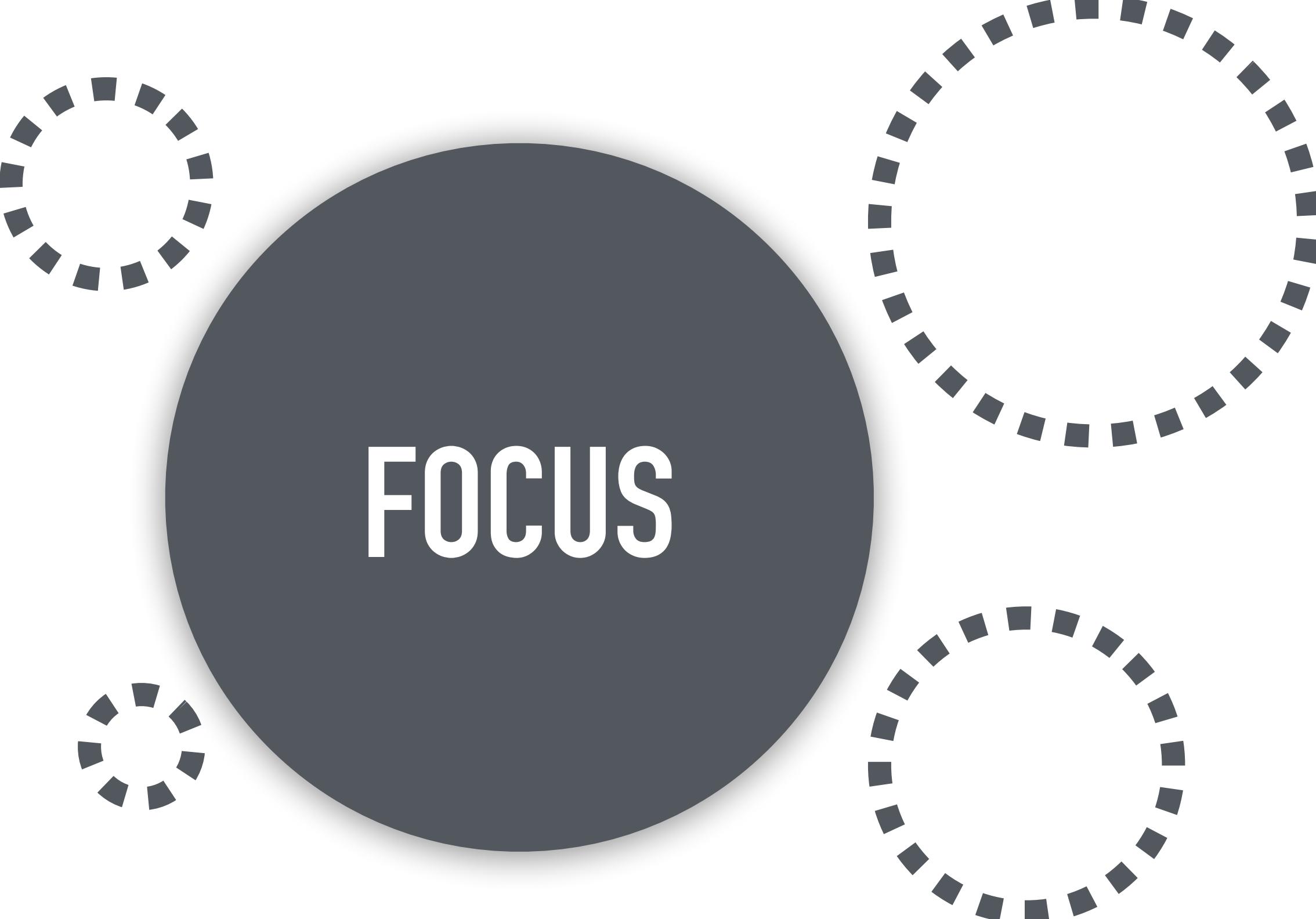


# OBJECT-RELATED WATERMARKS IN CARTOON IMAGES

---

G40N (<https://github.com/GAONNR>)

# OBJECT-RELATED?



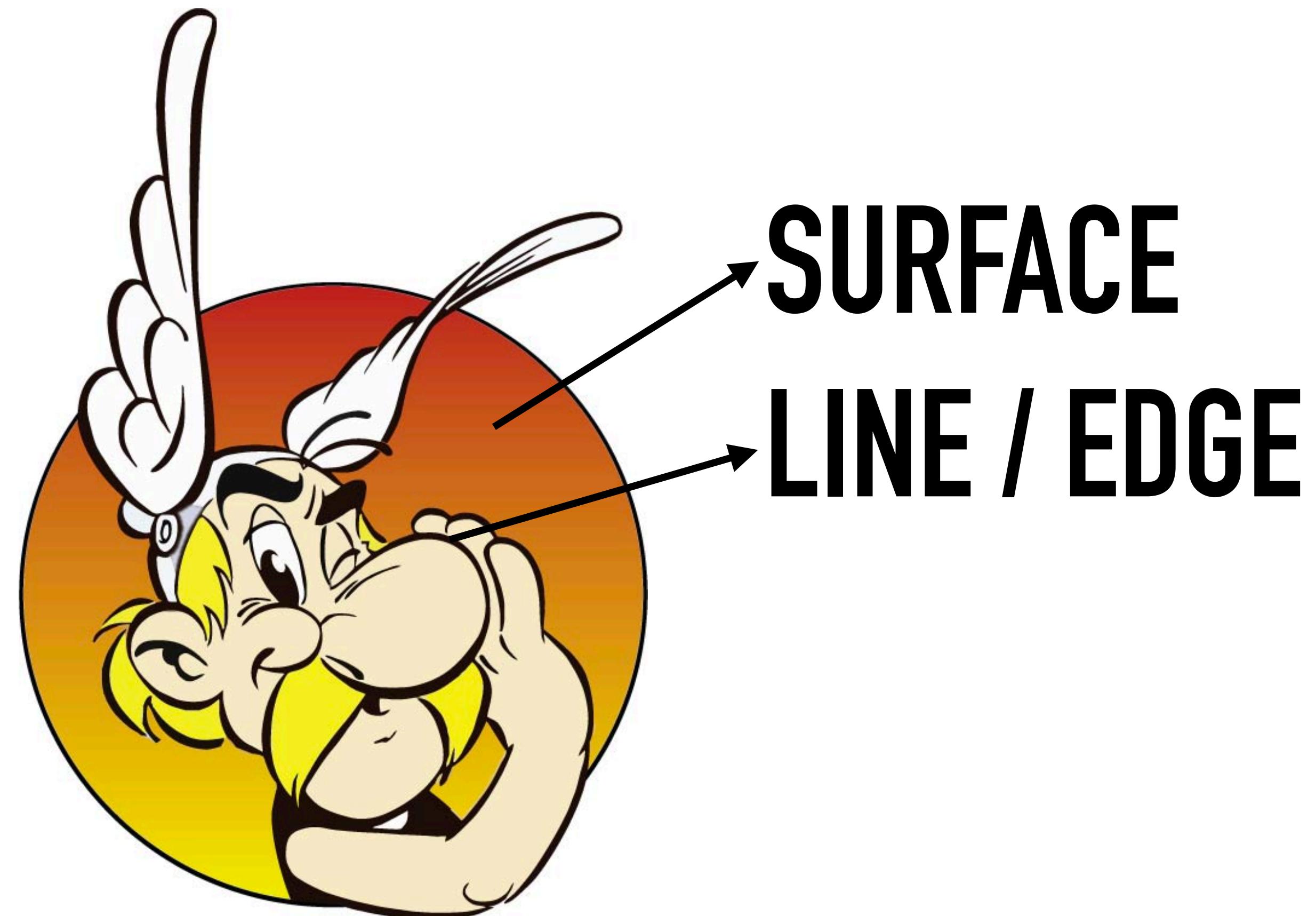
**FOCUS**



**WATERMARK**



**WHY CARTOON?**



**“EASY TO DISTINGUISH EDGES”**



# REAL-WORLD PICTURE

# HAND-DRAWN PICTURE

# WATERMARKING PROCESS

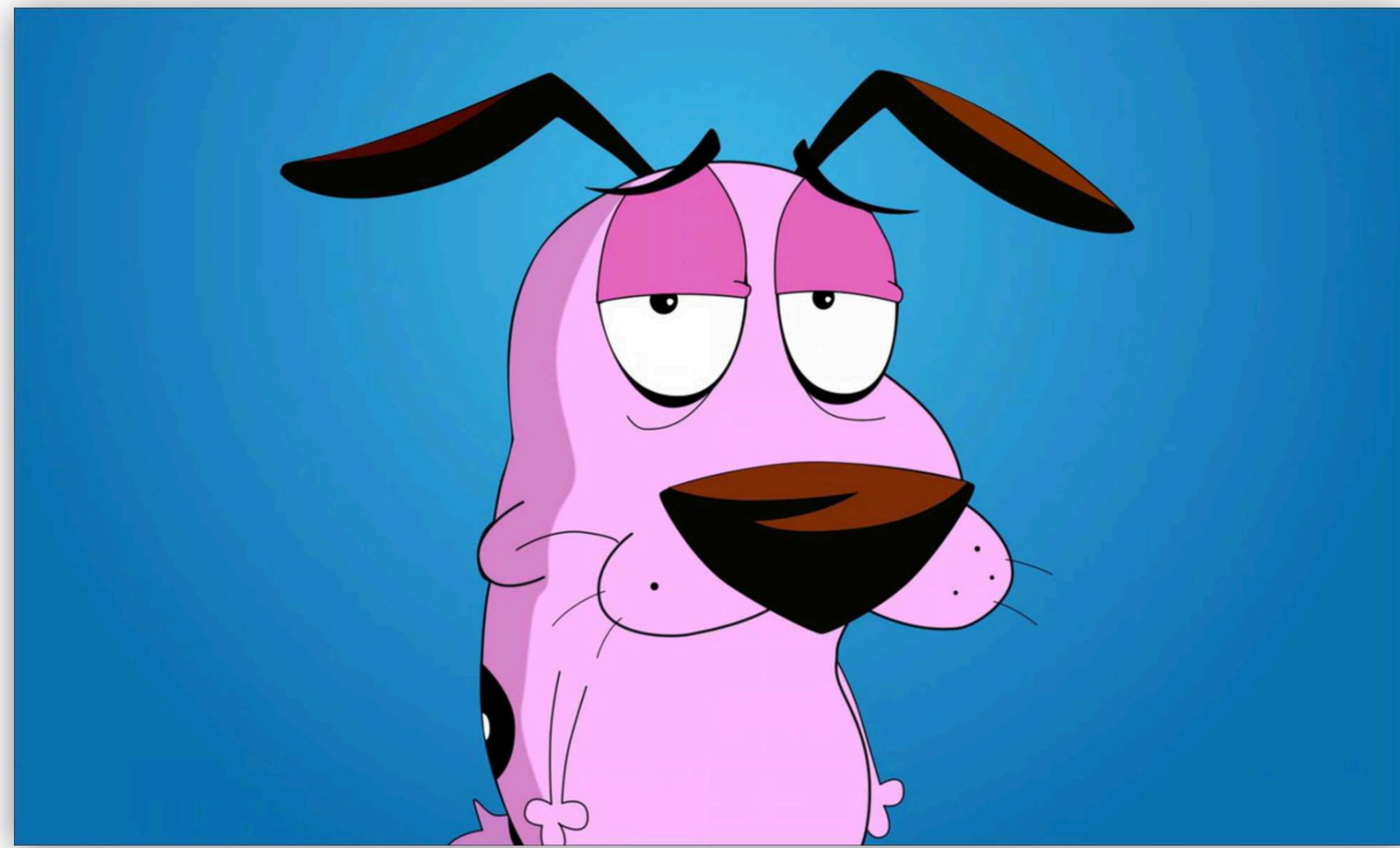
**CONVERSION TO GRayscale**

**EDGE DETECTION**

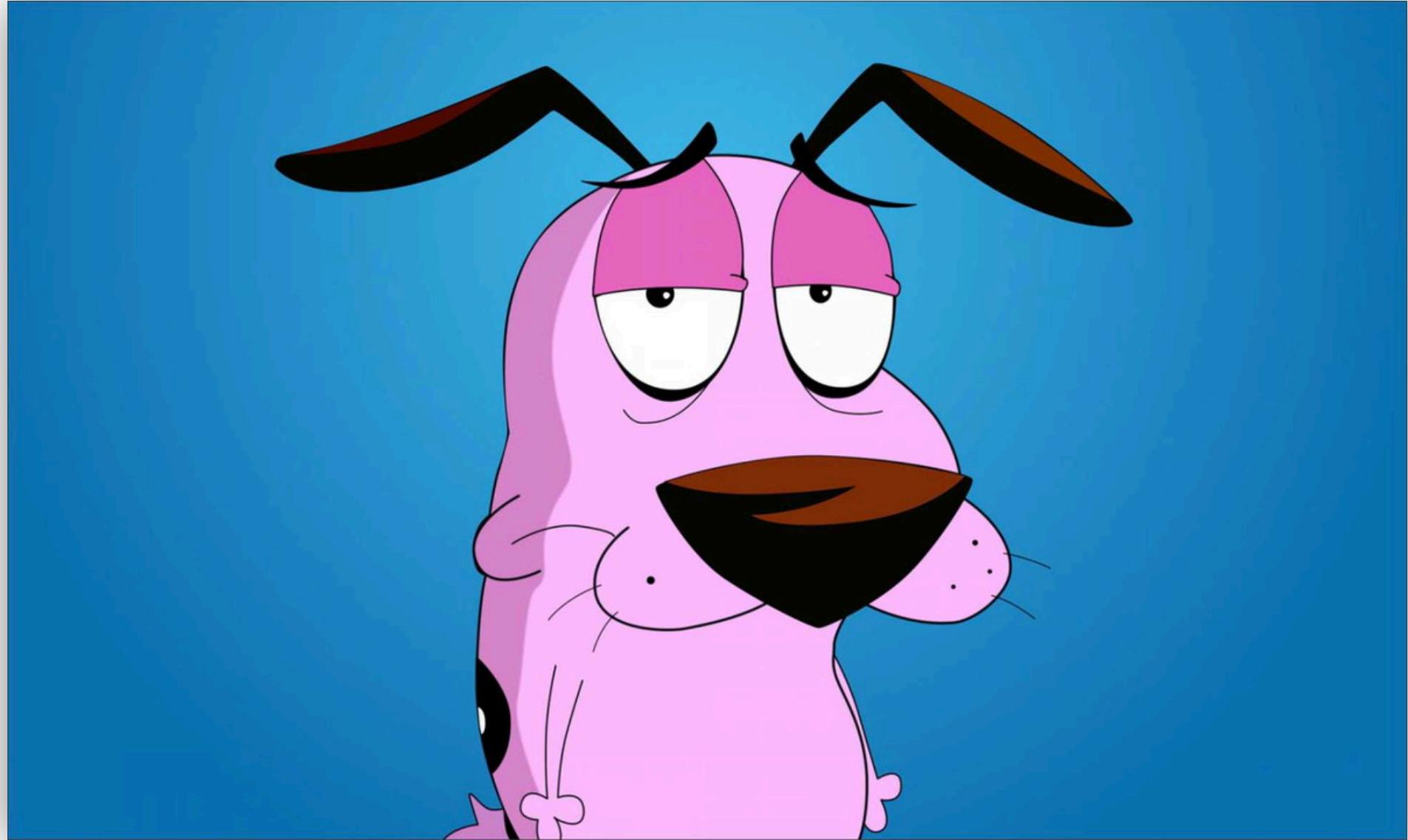
**EXTRACTING BOUNDARY**

**LSB WATERMARKING**

**TRANSFER TO ANOTHER IMAGE**



OUR HERO



$$\text{GRAY} = 0.3 \times \text{R} + 0.59 \times \text{G} + 0.11 \times \text{B}$$

## EDGE DETECTION

LAPLACE

PREWITT

SOBEL

CANNY

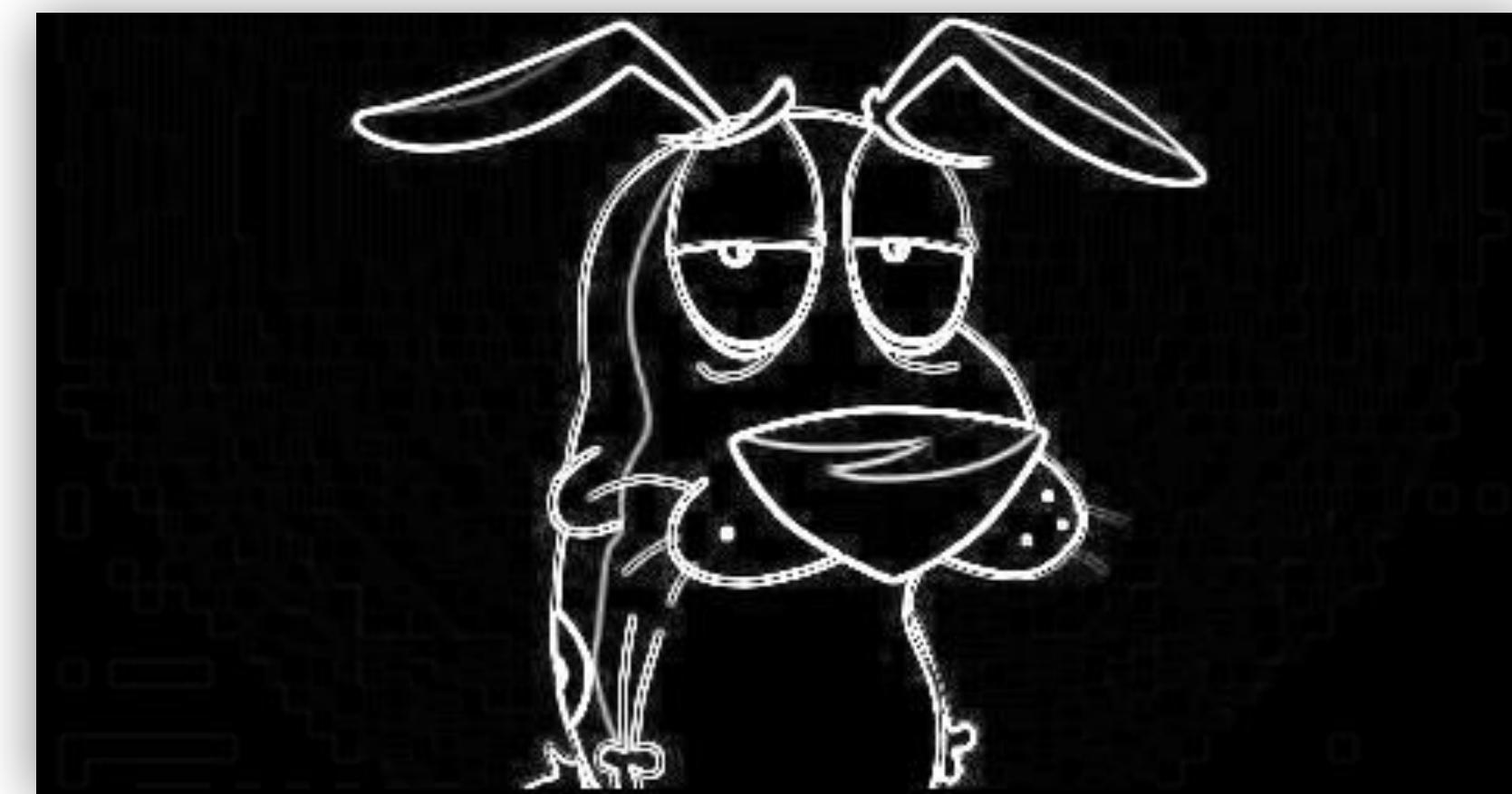
# SOBEL'S EDGE DETECTION

$$\mathbf{G}_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} * \mathbf{A} \quad \text{and} \quad \mathbf{G}_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * \mathbf{A}$$

Gx: Horizontal / Gy: Vertical

$$\mathbf{G} = \sqrt{{\mathbf{G}_x}^2 + {\mathbf{G}_y}^2}$$

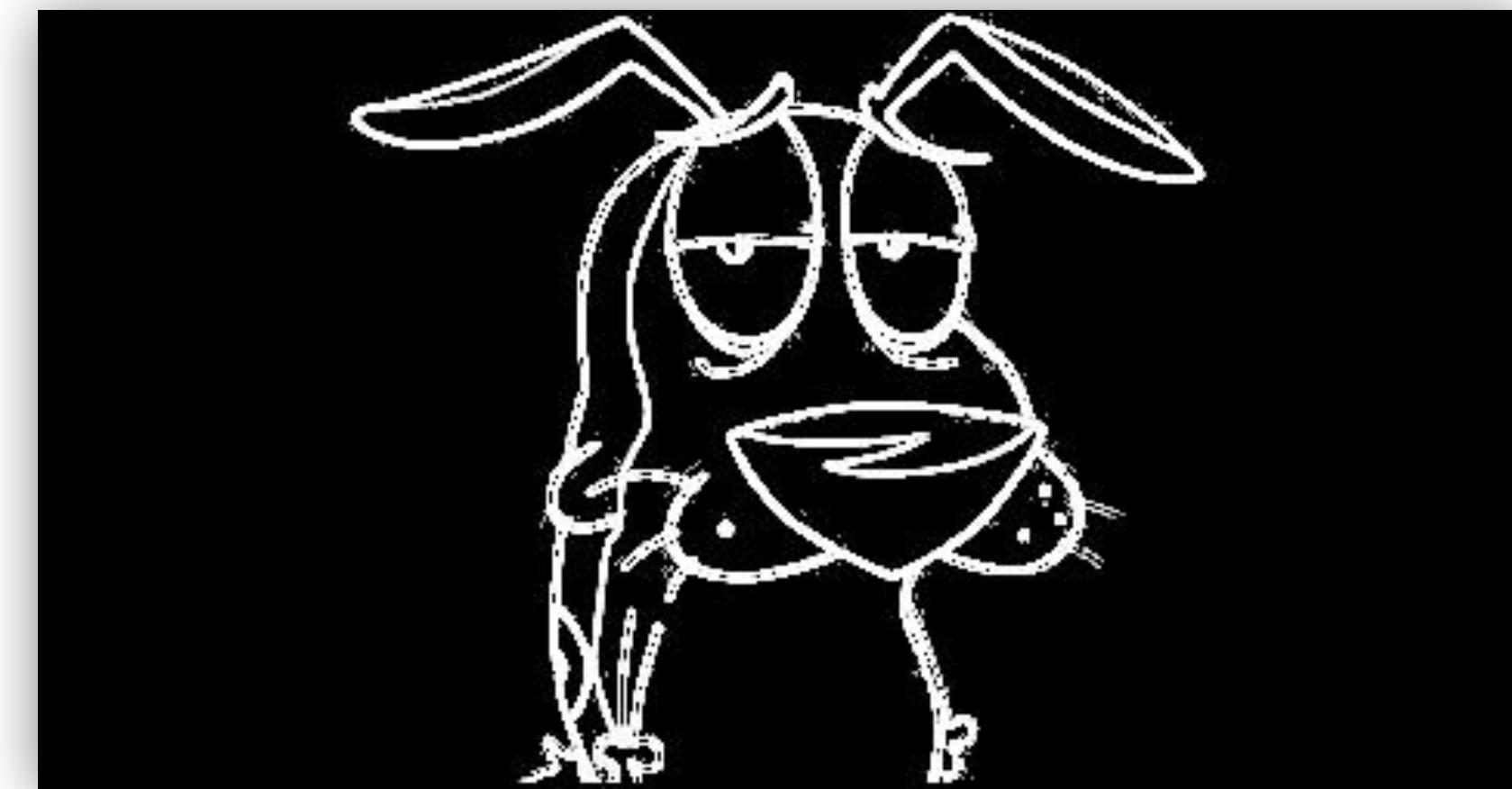
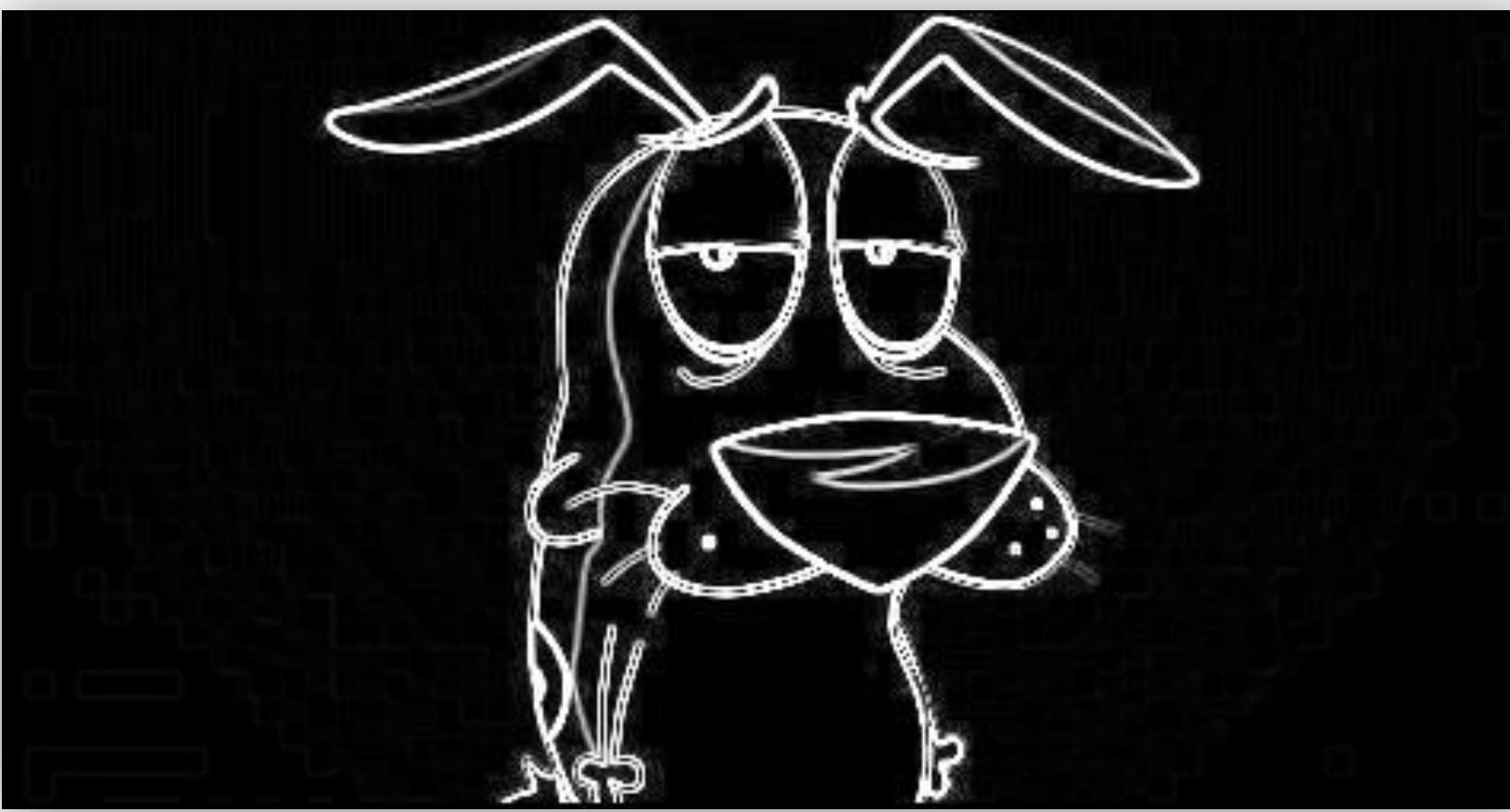
**(USED PIL, PYTHON)**



**WE DID IT!**

# MAYBE THERE'S SOME NOISE IN PICTURE STRONG CONTRAST COULD HELP

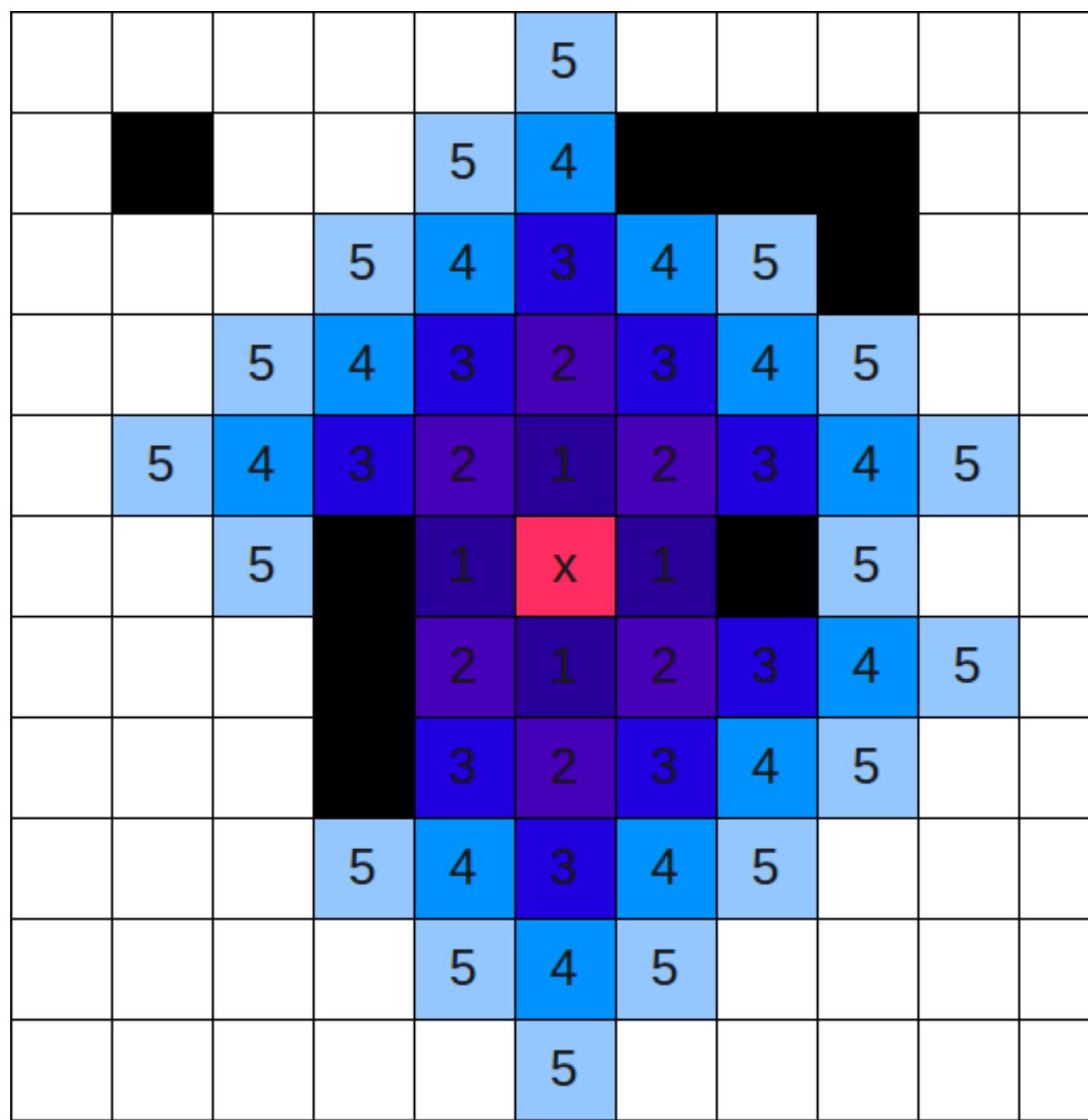


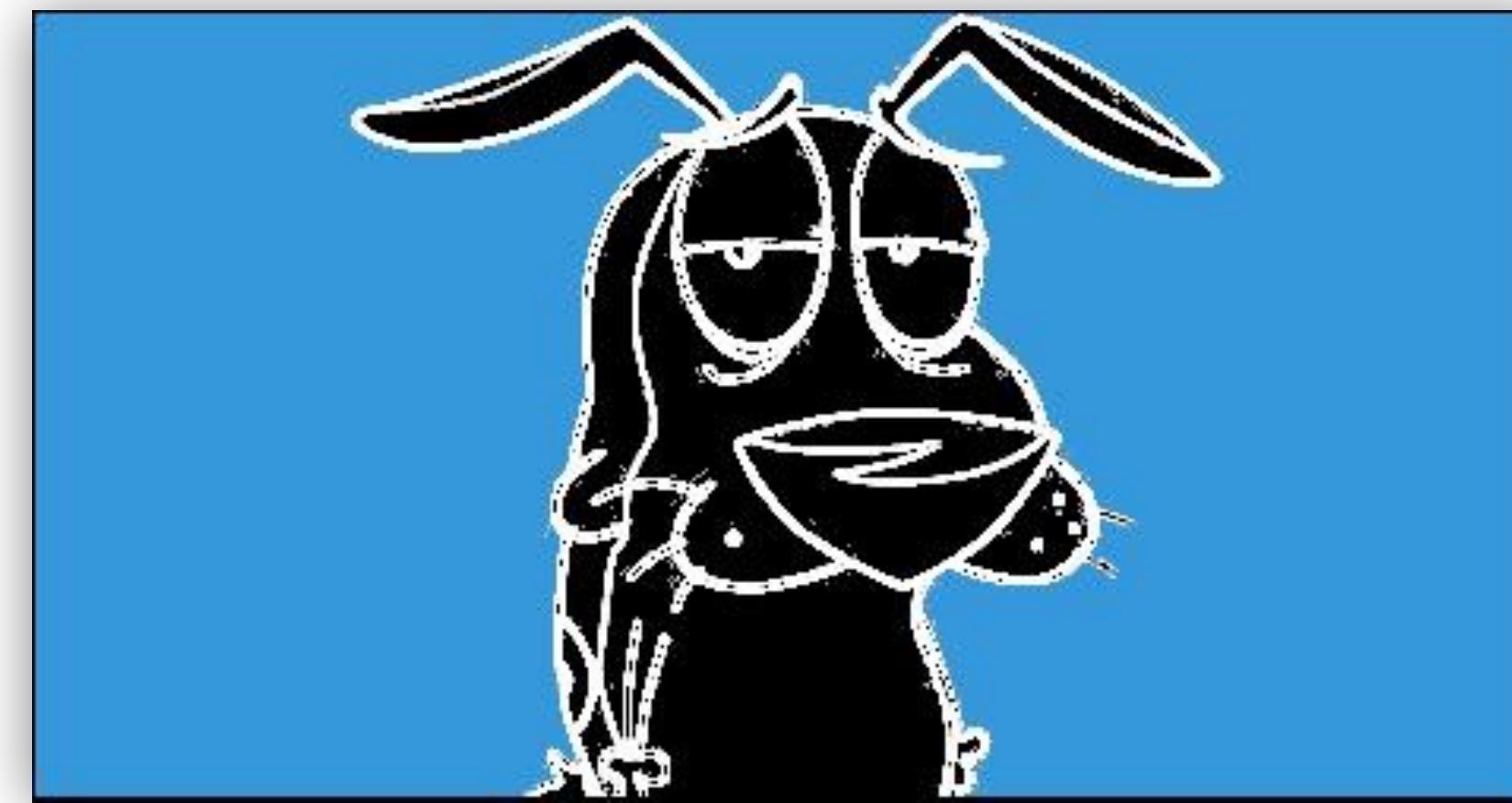
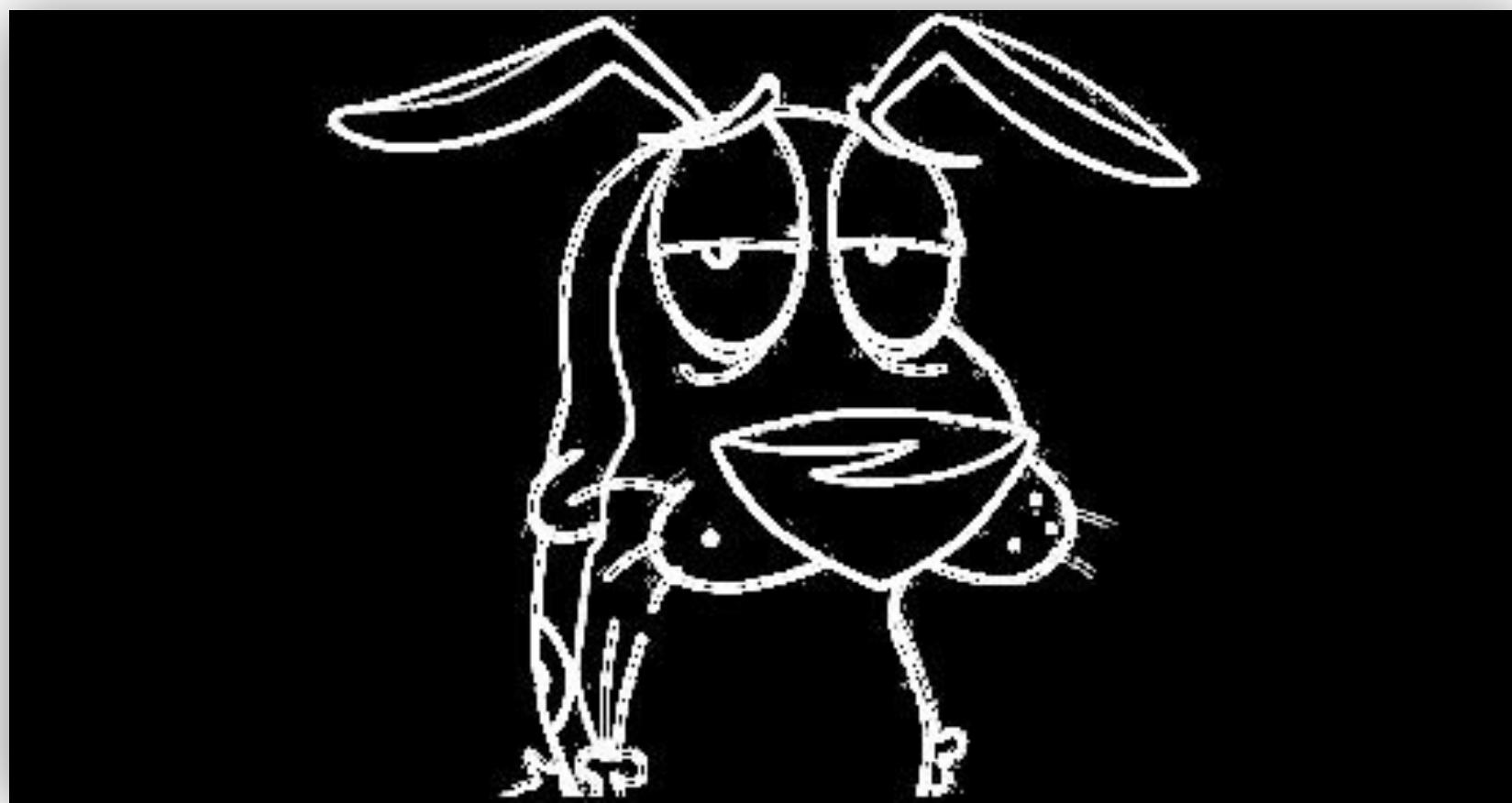


**IT LOOKS NICE (TO COMPUTER)**

**NOW WE NEED TO REMOVE “BACKGROUND”**

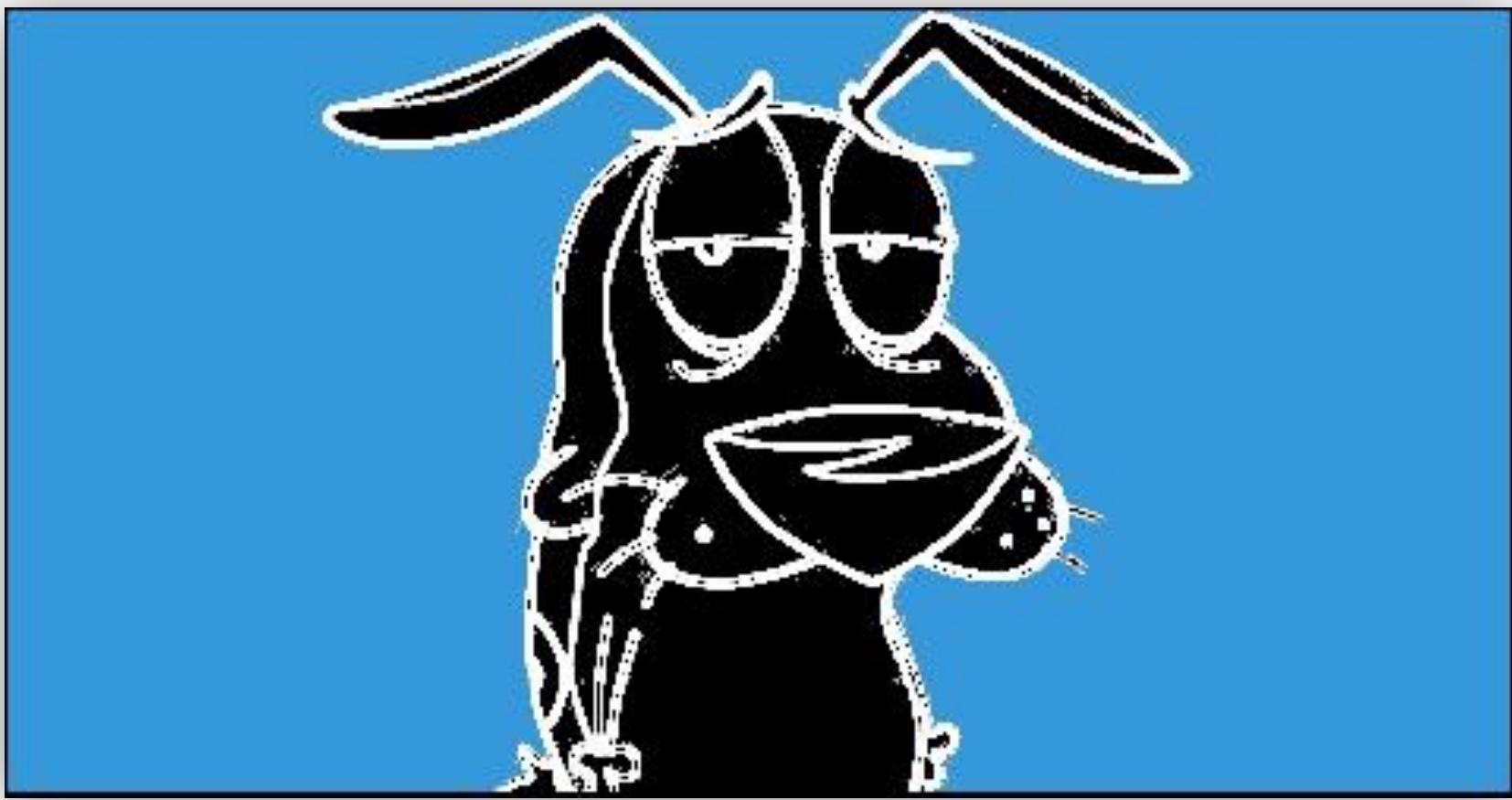
# “FLOOD-FILL ALGORITHM”





**SELECT SOME PIXELS TO DEFINE A BACKGROUND**

AND  
FLOOD-FILL  
AGAIN



**SELECT SOME PIXELS TO DEFINE AN OBJECT**

NOW IT'S TIME FOR EMBEDDING

# LSB TECHNIQUE, by W.M.WEI

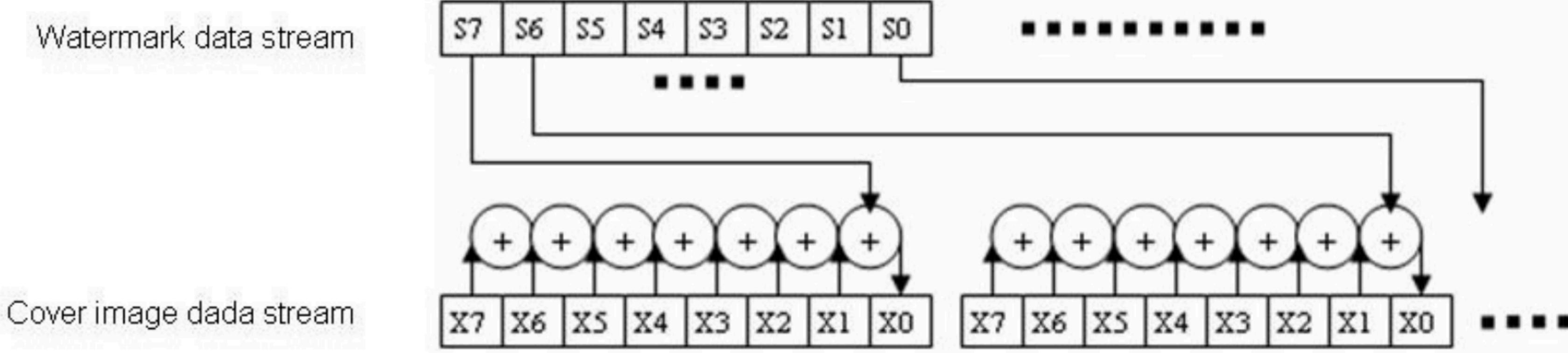
COLOR BITS

$X_7X_6X_5X_4X_3X_2X_1X_0$

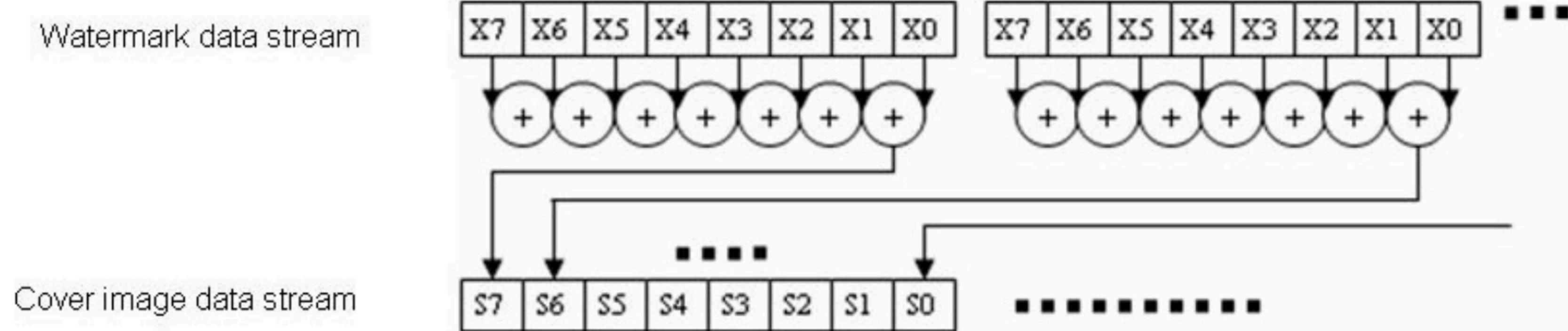
MSG BIT

S

$$X_0' = X_7 \oplus X_6 \oplus X_5 \oplus X_4 \oplus X_3 \oplus X_2 \oplus X_1 \oplus S$$



(a)



(b)

**ORIGINAL IMAGE**



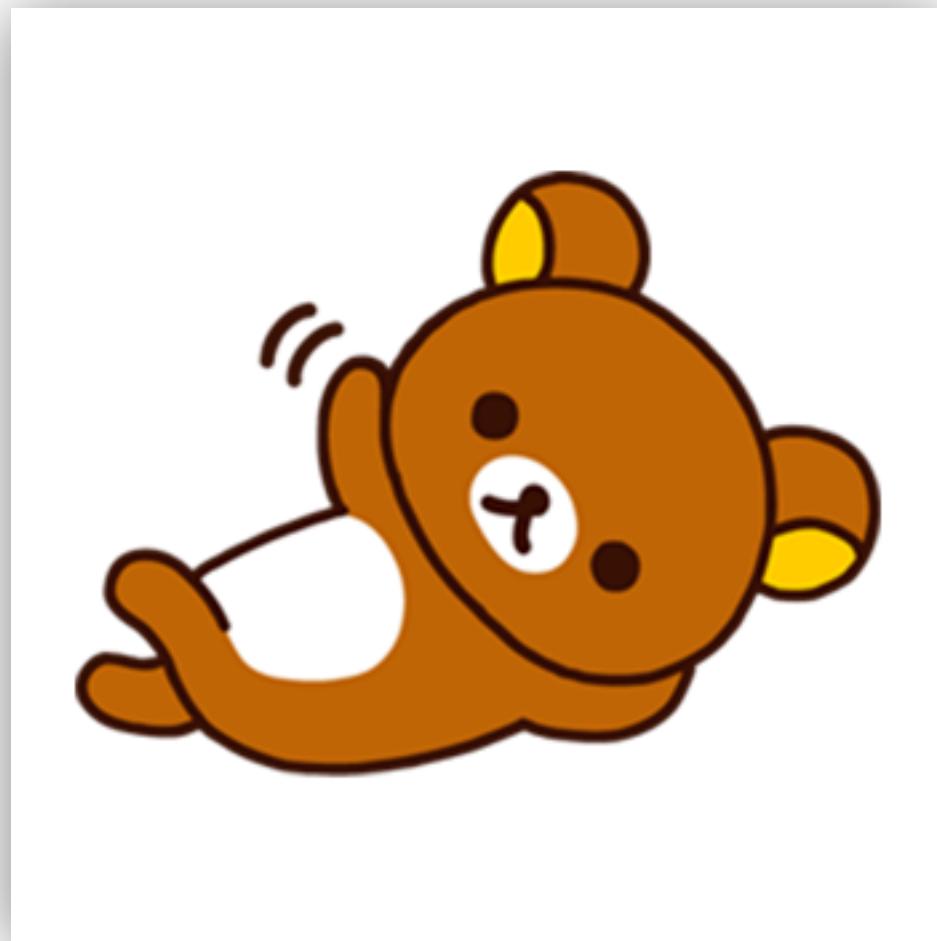
**WATERMARKED IMAGE**



**NOT SO DIFFERENT**

```
x gaonnarae@gaonnalaeui-MacBook-Pro ~ ~/Github/CartoonWatermarking master •+ >M< python3 decoder.py dog_embed.bmp  
(480, 251) RGB  
Give me a coordinate: 230 217  
Decoded! String is: mynameisdog
```

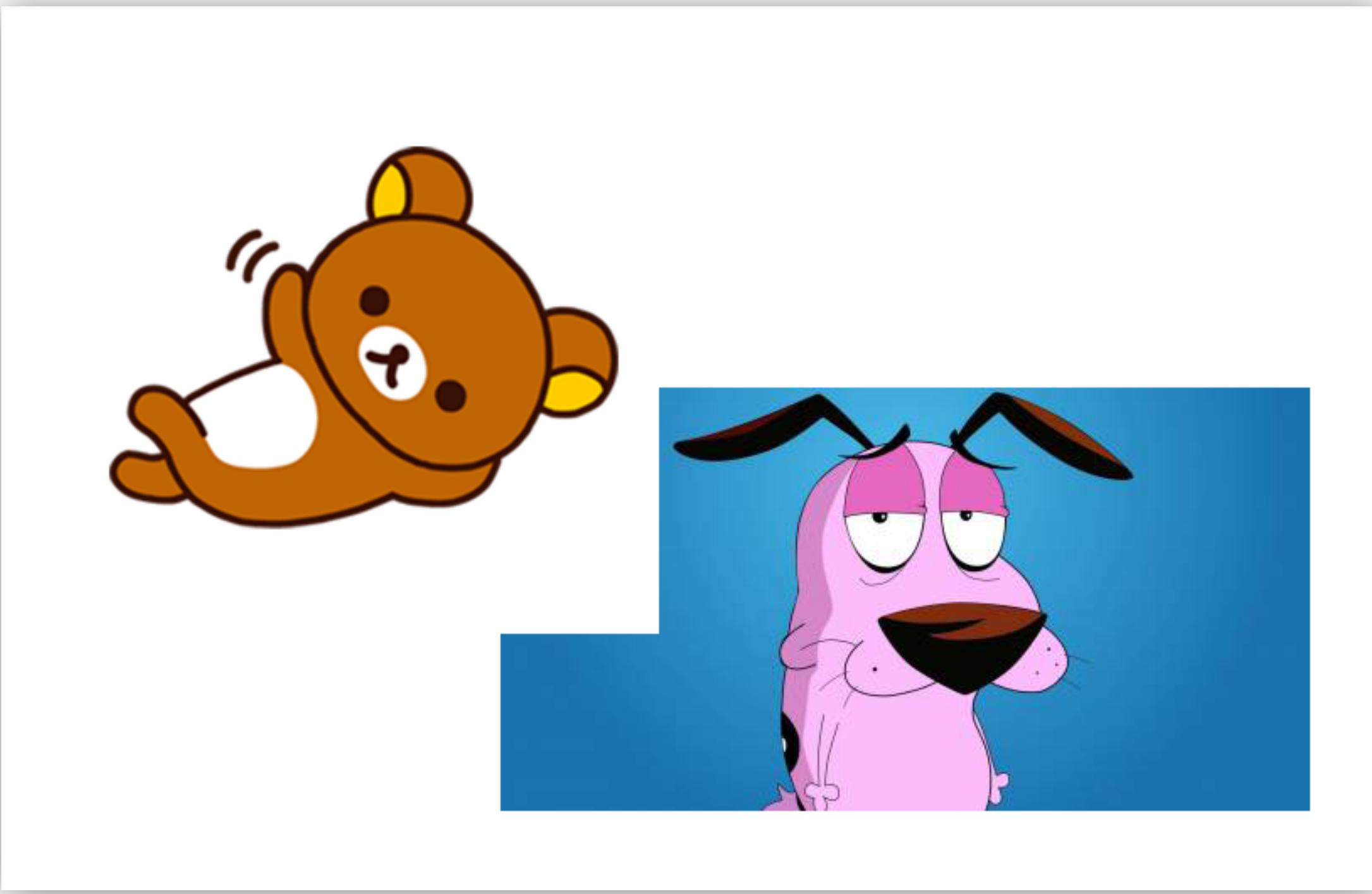
# BUT THERE'S A HIDDEN MESSAGE!



**“mynameisbear”**

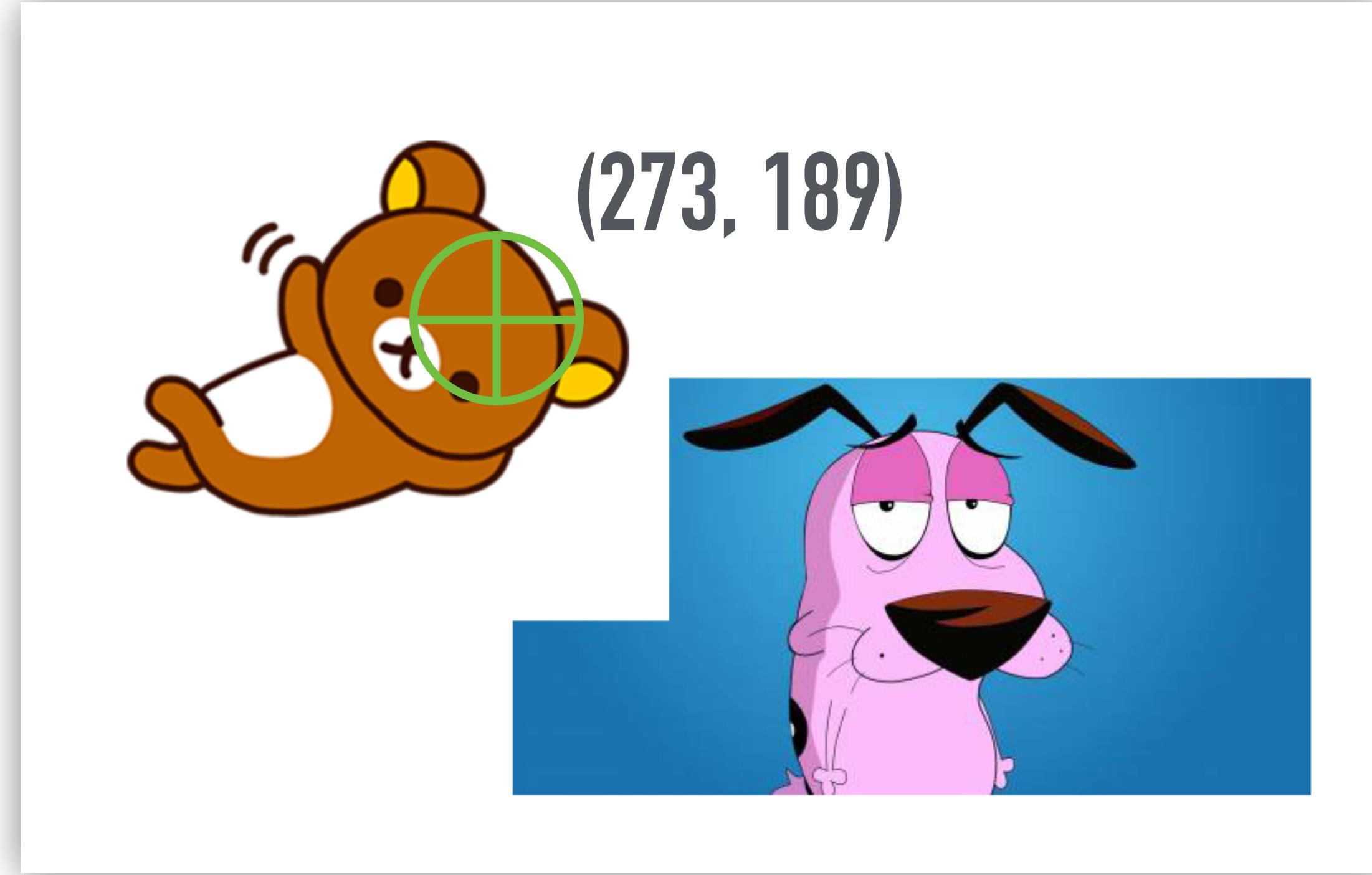


**“mynameisdog”**



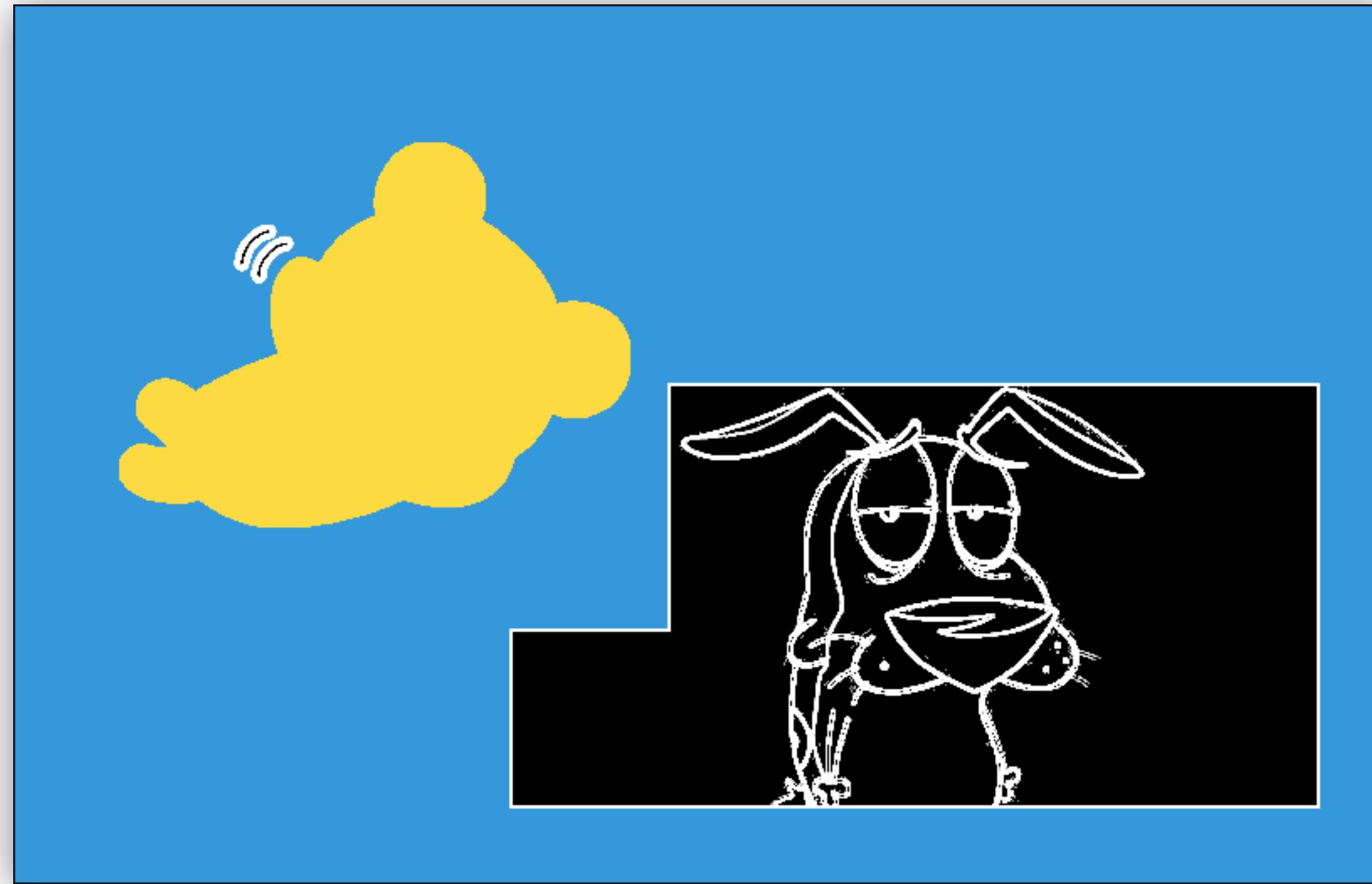
**WHAT WILL COME OUT?**

**DEPENDS ON THE PIXEL WHAT YOU CHOOSE**



(273, 189)

```
gaonnarae@gaonnalaehui-MacBook-Pro ~/Documents/Github/CartoonWatermarking master •+ >M< python3 decoder.py total.png  
(821, 524) RGB  
Give me a coordinate: 273 189  
Decoded! String is: mynameisbear
```



“mynameisbear”

# DEMO

ALL SOURCES ARE AVAILABLE AT  
[https://github.com/GAONNR/Cartoon\\_Watermark](https://github.com/GAONNR/Cartoon_Watermark)

# REFERENCE

Wanchun Luo, "Object-Related Illustration Watermarks in Cartoon Images", 2004  
[\(http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.91.4161&rep=rep1&type=pdf\)](http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.91.4161&rep=rep1&type=pdf)

VISHWANATH1991, "CANNY EDGE DETECTION", 2012  
[\(https://cyroforge.wordpress.com/2012/01/21/canny-edge-detection/\)](https://cyroforge.wordpress.com/2012/01/21/canny-edge-detection/)

DAVID COLLIEN, "EDGE DETECTION", OPENLEARNING  
[\(https://www.openlearning.com/courses/COMP087390/Modules/AmazingScaling/Content/EdgeDetection\)](https://www.openlearning.com/courses/COMP087390/Modules/AmazingScaling/Content/EdgeDetection)

PILLOW LIBRARY, PIL FOR Python3  
[\(https://python-pillow.org/\)](https://python-pillow.org/)

A dense, overlapping crowd of The Simpsons characters in various poses and expressions, serving as the background for the text.

**THANK YOU!**