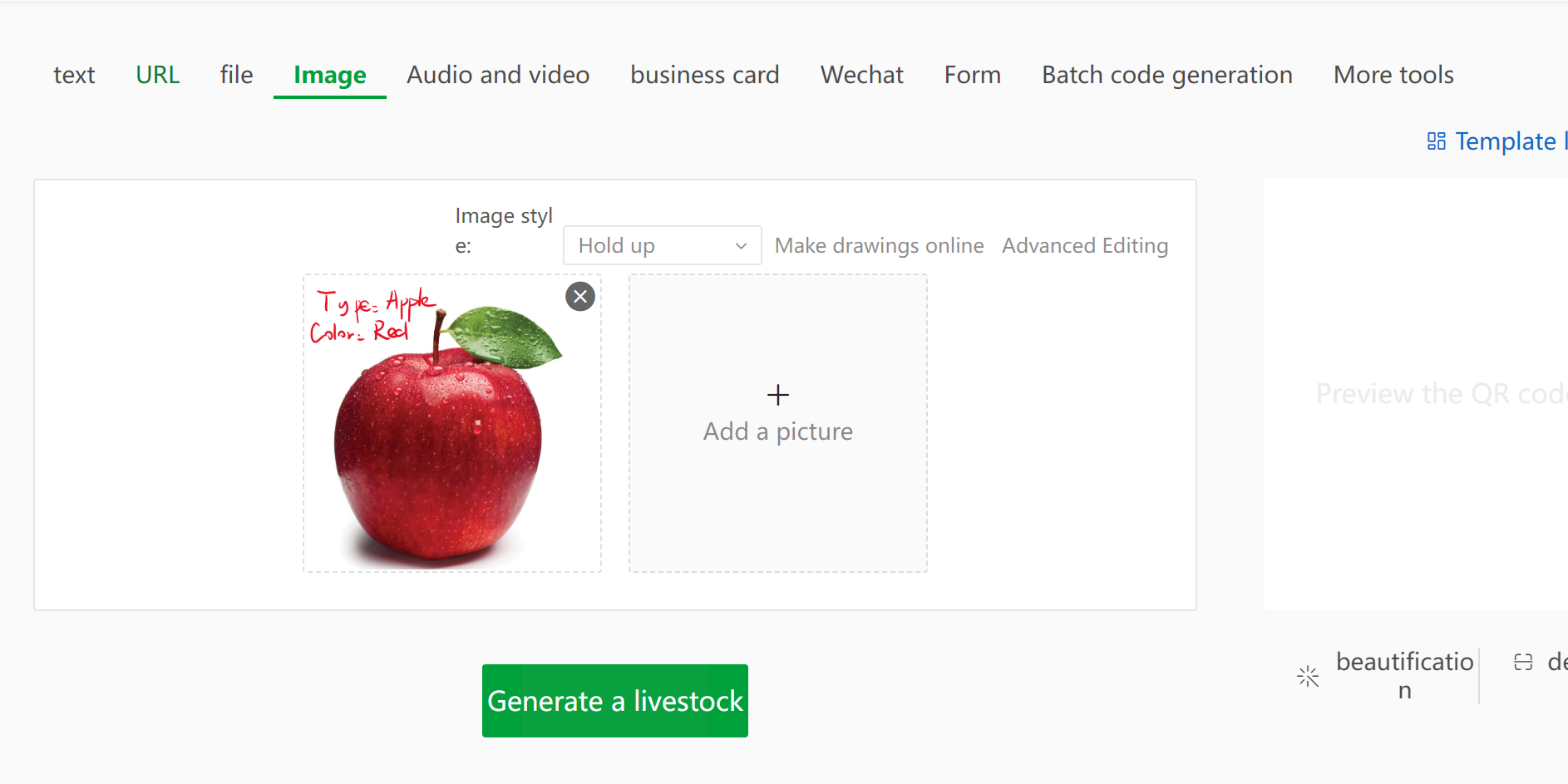
This week we try to find a good QR code generator and try to put as much as information we want when we scan the QR code.

So there are open source QR Code Generator everywhere and we can put different types of information in the QR code.



We made an QR Code example just like the Code below. The code itself can also be various, different colors, etc..



And the result from the QR code is below.



This week we tried to write a script with the API from Open CV that we can get the picture/frame of the videos that we shot by cameras. And also, it’s very important to get the QR Code from different angles, positions and brightness.

Before you run the program, you need to first install the cv2&numpy&pyzbar with pip in windows.

# This script is used to scan the QR code in the video with openCV

import cv2

import numpy as np

from pyzbar.pyzbar import decode

global cap

cap = cv2.VideoCapture('3d07299619f03fa26b9f6fd6ff7c40f5.mp4') #Open the video

# Scan the code from the picture（decode）

def Read\_Decode\_Pic(image):

# Iteration

for code in decode(image):

# print("QR Code：", code)

data = code.data.decode('utf-8')

print("QR Code：", data) #decode the data

# Polygon acquisition (box of rectangles)

pts\_poly= np.array(code.polygon, np.int32) #Get polygon coordinates

cv2.polylines(image, [pts\_poly], True, 5) #draw polygon frame

# Display data (get the upper left corner of the rectangular box as the coordinates of Text (left coordinate), display data)

pts\_rect = code.rect

# print(pts\_rect)

cv2.putText(image, data, (pts\_rect[0],pts\_rect[1]), cv2.FONT\_HERSHEY\_SIMPLEX, 0.7, 2)

# Display data Rectangle coordinates Font type Font size Color Thickness

if not cap.isOpened():

print("Cannot open camera")

exit()

cv2.imshow('image', image) # Wait for all rectangles to be drawn and then display

# Scan the code from the video（decode）

def Read\_Decode\_Cam():

cap.set(3, 1000) #The width of the frame

cap.set(4, 800) #The height of the frame

while True:

success, image = cap.read() #Get each frame

if not cap.isOpened():

print("Cannot open camera")

exit()

if image is None:

exit()

cv2.imshow('image', image)

#quit if not be able to use camera

# if not success:

# break

if image is None:

continue

image = Read\_Decode\_Pic(image) #Detection for each frame

# cv2.imshow('result', image)

cv2.waitKey(1) #delay 1ms

if \_\_name\_\_ == '\_\_main\_\_':

# img = cv2.imread('cacd048312468152f0ffb6e7041b318.png')

# Read\_Decode\_Pic(img) # Scan the code from the picture（decode）

Read\_Decode\_Cam() # Scan the code from the video（decode）

cv2.waitKey(0)

sSo we gave an example right here.

**Video** :An example of scanning QR Code from Video in different angles and positions

<https://www.youtube.com/watch?v=CLSybKP6h-M>

We successfully finished the first milestone and we will tried to test this methodology in real situation next week.