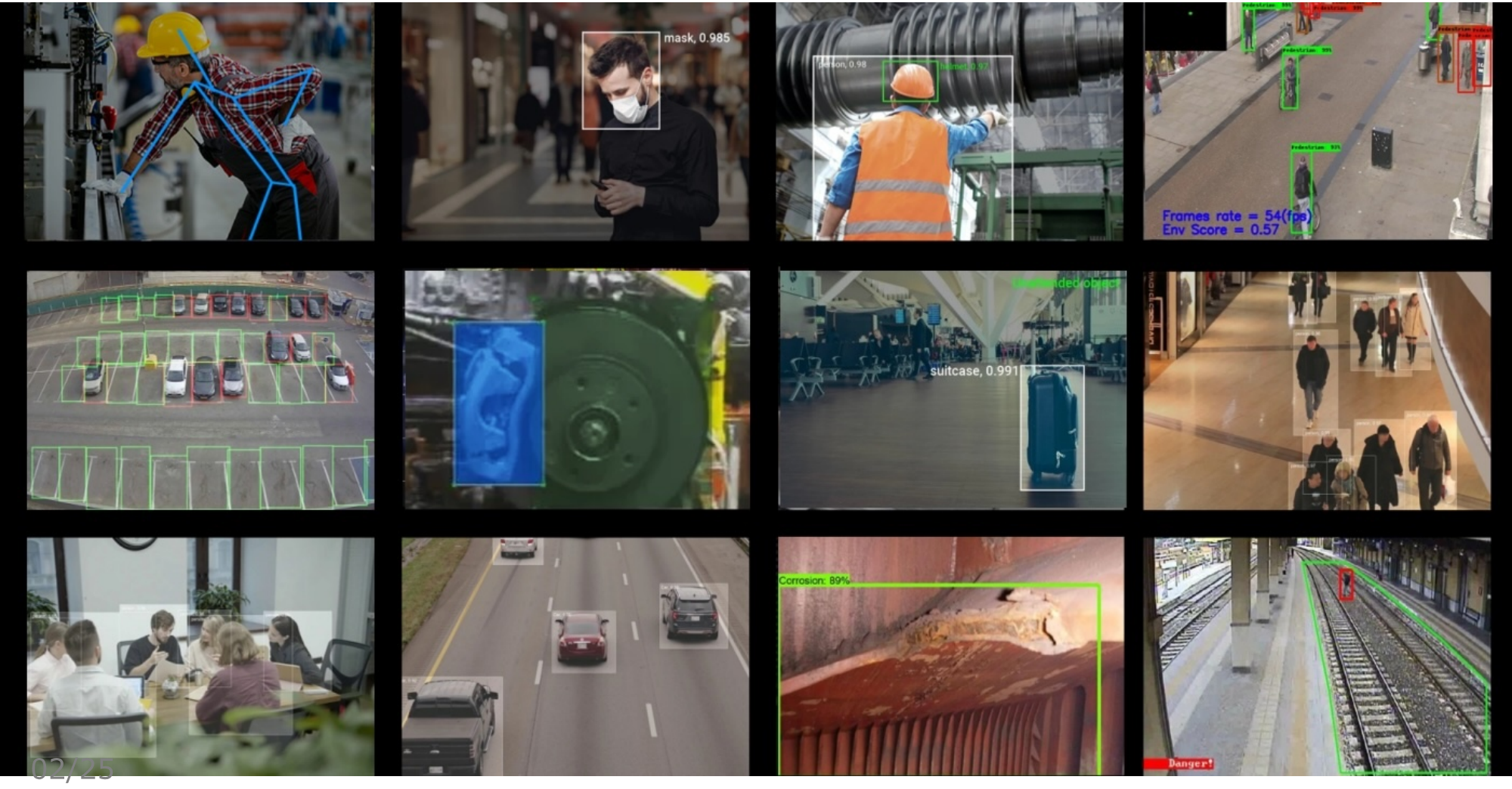
Explorations in Computer Vision

Dr Simon Lock

Some Computer Vision Applications



Different Levels of Comprehension

- Attention: is there *something* (anything) present ?
- Detection: specific *type* of thing (car, person etc) ?
- Recognition: which *individual* thing (car, person) ?
- Interpretation: what is the thing actually *doing* ?

Our Approach

We *could* try to train a machine learning model
In order to identify objects/structures in an image
But this wouldn't give much insight into the process
(We'd just be training, rather than experimenting)

Instead we will take a much lower-level approach
Focusing on writing code to analyse images/video
Using a number of pixel manipulation techniques

Pixel Manipulation Techniques

- Searching for specific colours (RGB, Hue, Sat. etc.)
- Image differencing (change in Brightness, Hue etc.)
- Scanning for specific shapes (person, car, letter etc.)
- Matching relative structures (e.g. face detection)

The Application

We need some form of visual material to process

We will make use of a recording of a webcam stream

From the "Marine Biological Association" in Plymouth:

3m5!1s0x486c935311b11e0b:0xd40b5a4f7f597d10!8m2!3

It's a nice view, with lots of activity going on !

PlymouthCamSolution

Processing Template

To get you started, we provide a template project

This template illustrates how you can:

- Open up an MP4 video file
- Extract a single image ("frame") from the video
- Draw that frame onto the screen
- "Mask off" unwanted areas of the frame
- Analyse remaining pixels contained within the frame

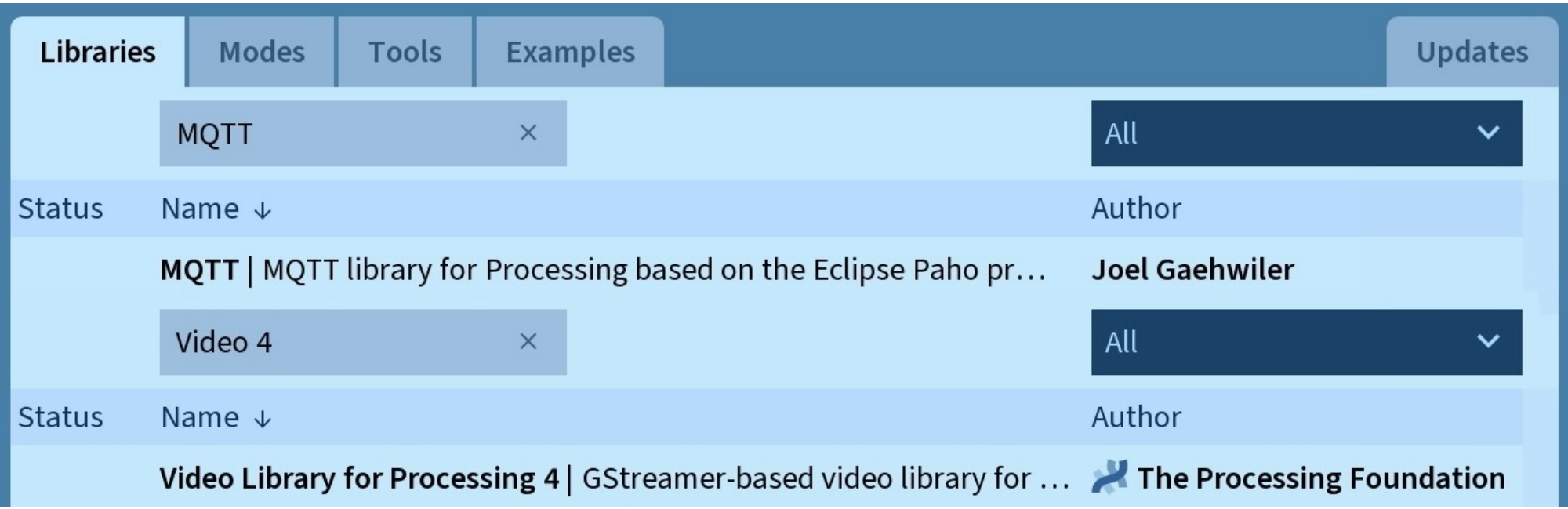
PlymouthCamTemplate

Libraries

Note that in order to run the template project

You will need to install a couple of libraries using:

`Sketch > Import Library > Manage Libraries`



Checking the Colour of Individual Pixels

We can get the colour of a particular pixel using:

```
int pixelColour = currentFrame.get(x,y);
```

Then extract various properties of the pixel using:

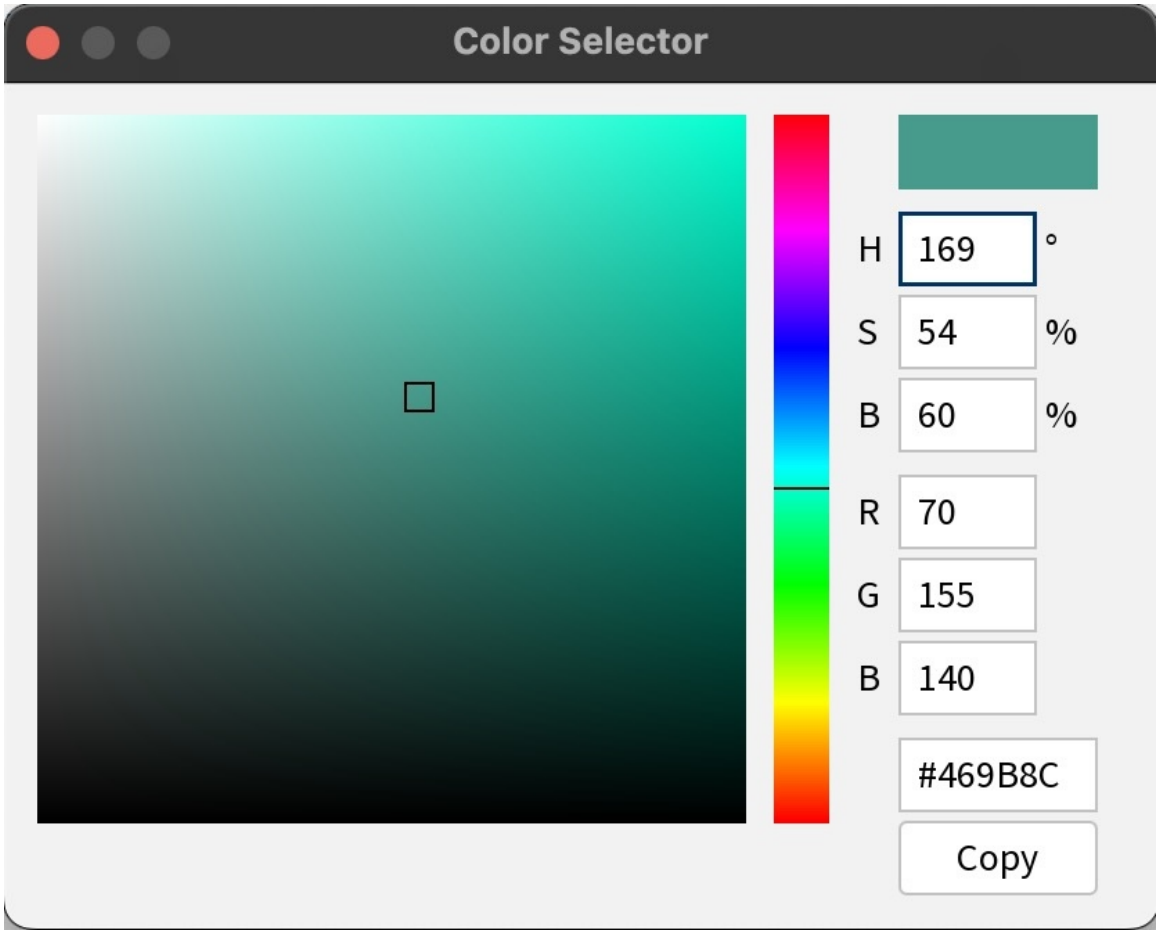
```
int redness = red(pixelColour);
int greenness = green(pixelColour);
int blueness = blue(pixelColour);
int whichColour = hue(pixelColour);
int howBright = brightness(pixelColour);
int howSaturated = saturation(pixelColour);
```

Drawing Pixels onto a Frame

- Before any drawing, call "beginDraw" on the frame
- To mask off areas set "fill" colour to black
- Mask off Rectangle, Triangle or Polygon areas
- To draw individual pixels set "stroke" colour...
- Then draw an individual "point" (pixel) in that colour
- After drawing, be sure to call "endDraw" on Frame

Hue / Saturation / Brightness / RGB

Warning: All values in Processing are in range 0-255



Your Objective

Select an aspect of the scene to analyse/monitor

You are free to choose any element that you like...

But try to choose something with a clear "purpose"

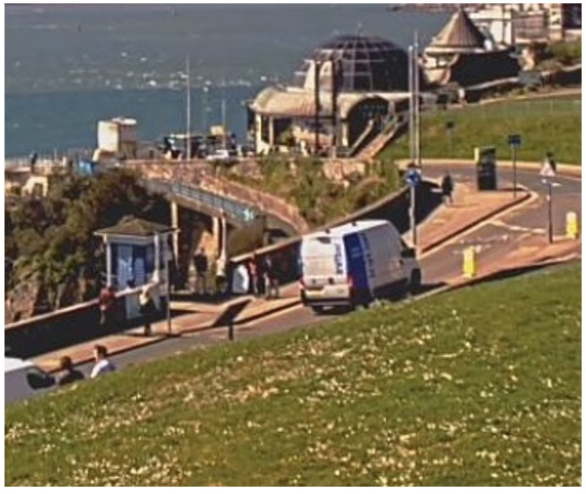
(Something it might actually be useful to monitor !)

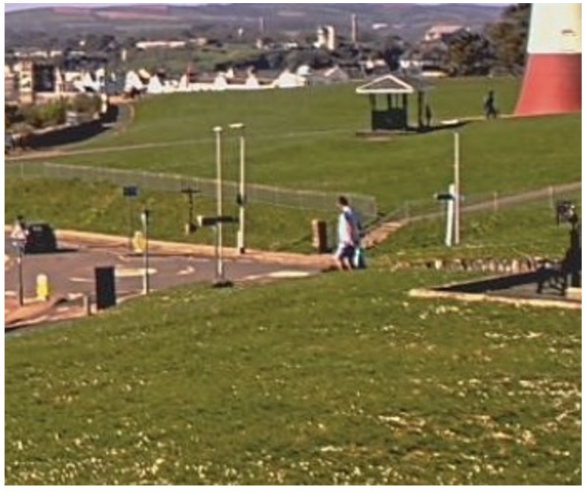
Later, we will be send someone a notification

When a particular situation occurs in the scene

What Will You Choose ?





To Work !

Blob Detection

Blob Detection is a more advanced technique

(But something still achievable within this session)

Involves scanning for coherent clusters of pixels

(Based on the pixel brightness or colour)

Narrowing down the bounding box to get a tight fit

BlobDetection

Sending Notifications

It's no good being able to detect something...

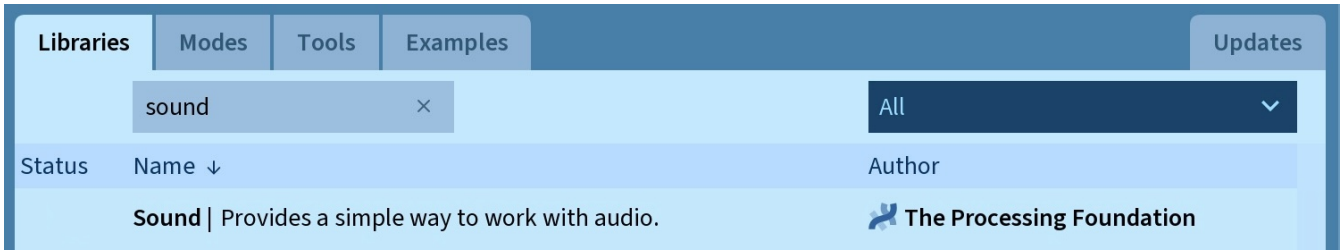
If nobody actually realises the event has occurred !

Add an audible alert using the "SoundFile" class:

tps://processing.org/reference/libraries/sound/SoundFile.ht

Note that you will need to install the "sound" library:

`Sketch > Import Library... > Manage Libraries...`



Remote Notifications

What if we aren't in the same room as analyser ?
We won't be able to actually hear the alert !

We might even want to notify multiple remote users
In case many people are interested in the event

There are various mechanisms available to do this
We use a communication mechanism called MQTT...

Message Queue Telemetry Transport (MQTT)

Message passing protocol (ideal for notifications !)
You can "publish" messages to "topics" (channels)
Other people "subscribe" to receive notifications

Connect to broker by adding this to setup method:

```
mqtt = new MQTTClient(this);
mqtt.connect("mqtt://broker.hivemq.com:1883");
```

Then to publish message/notification to topic call:

```
mqtt.publish(topic, parameter);
```

Topic must start with "plymouth/" for example:

```
mqtt.publish("plymouth/birds", "3");
```

MQTT Demo

PlymouthCamSolution

NotificationMonitor

Monitoring Published Messages

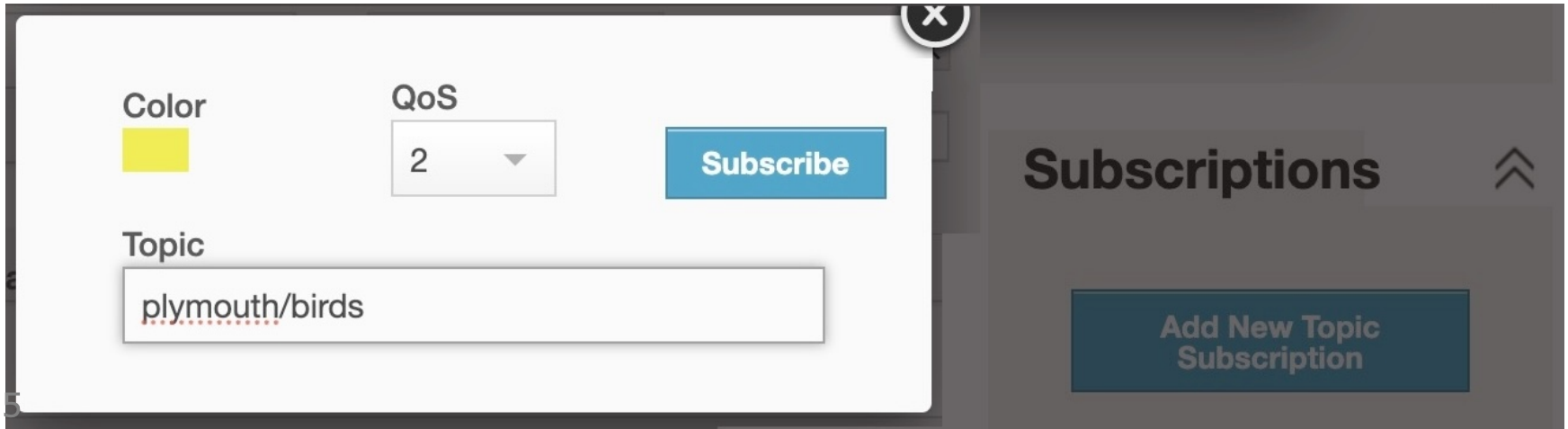
Monitor published messages using the web client:

https://www.hivemq.com/demos/websocket-client/

Connect using the default settings

"Subscriptions" > "Add New Topic Subscription"

Then add "topic" of interest (e.g. "plymouth/birds")



To Work !

Alternative Servers

A server that gathers messages is called a "Broker"
(Arbitrates between message senders and receivers)

For this activity we are using a free public broker
Since it's free there are no guarantees of availability
Sometimes the server is not operational :o(

That's fine - there are other alternatives available:

`mqtt://broker.hivemq.com:1883`

`mqtt://test.mosquitto.org:1883`

`mqtt://public-mqtt-broker.bevywise.com:1883`

Alternative Video

Your code may work with the original video, but...
Is it flexible enough to operate in other conditions ?
What if it is a cloudy day ? Or even raining ?

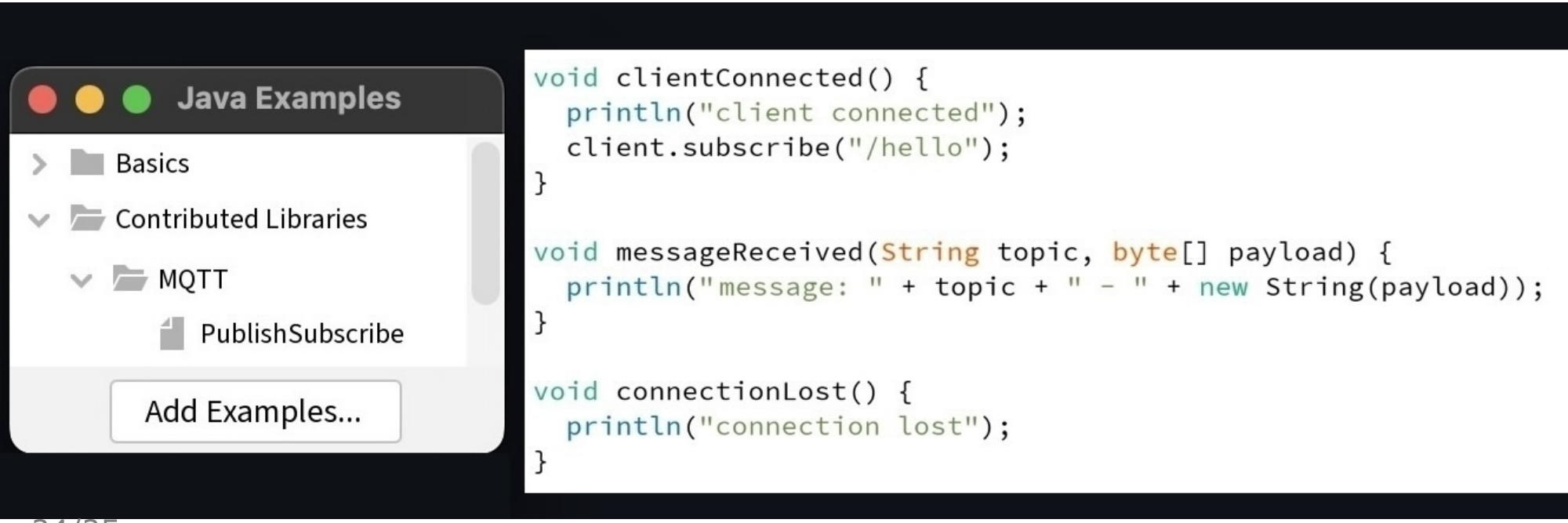
Try your analysis on an alternative video stream
If it doesn't work, you will need to adapt your code
(But make sure it still works with the original video !)

What else is going on ?

Do you want to see what everyone else is doing ?

You can create a new Processing project to find out !

Take a look at the MQTT examples (File>Examples...)



What is happening right now ?

https://www.mba.ac.uk/mbawebcam/

Want test your code analysing the live video feed ?
Find all instances of: drawMovieOntoFrame
And replace them with: drawLiveFeedOntoFrame

For example, the following line:

```
drawMovieOntoFrame(currentFrame);
```

Would now become:

```
drawLiveFeedOntoFrame(currentFrame);
```

Be careful - try not to behave like a DoS attack !