Day 3

Object Recognition

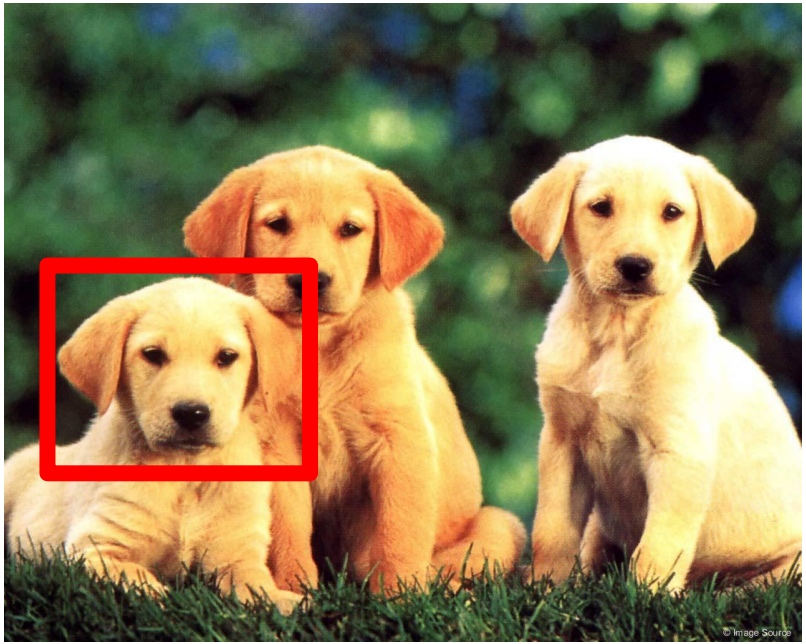
How computers can recognize.

Basic Idea - Pattern Matching

We know what we want to match, so we take it and we create a template to match against.

Basic Idea - Pattern Matching

We know what we want to match, so we take it and we create a template to match against.



Basic Idea - Pattern Matching

We know what we want to match, so we take it and we create a template to match against.

Basic Idea - Pattern Matching

We know what we want to match, so we take it and we create a template to match against.

The most basic way is we go through a sliding window approach

Basic Idea - Pattern Matching

We know what we want to match, so we take it and we create a template to match against.

The most basic way is we go through a sliding window approach

Basic Idea - Pattern Matching

We know what we want to match, so we take it and we create a template to match against.

The most basic way is we go through a sliding window approach

Basic Idea - Pattern Matching

We know what we want to match, so we take it and we create a template to match against.

The most basic way is we go through a sliding window approach

Basic Idea - Pattern Matching

We know what we want to match, so we take it and we create a template to match against.

The most basic way is we go through a sliding window approach

Basic Idea - Pattern Matching

We know what we want to match, so we take it and we create a template to match against.

1	2	3	4
2	3	4	1
3	4	2	2
4	5	3	2

3	4
4	2

Basic Idea - Pattern Matching

We know what we want to match, so we take it and we create a template to match against.

Another way we can think of this is a filter.

1	2	3	4
2	3	4	1
3	4	2	2
4	5	3	2

3	4
4	2

Basic Idea - Pattern Matching

We know what we want to match, so we take it and we create a template to match against.

Another way we can think of this is a filter.

But what operation do we want?

1	2	3	4
2	3	4	1
3	4	2	2
4	5	3	2

3	4
4	2

Basic Idea - Pattern Matching

How would I determine how similar 2 numbers were?

Basic Idea - Pattern Matching

How do I determine if two numbers are similar or equal?

Example: 120 and 50 , 34 and 31, 14 and 14

Basic Idea - Pattern Matching

How do I determine if two numbers are similar or equal?

Example: 120 and 50 , 34 and 31, 14 and 14

We would use subtraction

120 - 50 = 70, 34 -31 = 3, 14 - 14 = 0

Basic Idea - Pattern Matching

How would I determine how similar 2 numbers were or if they were equal?

Example: 120 and 50 , 34 and 31, 14 and 14

We would use subtraction

120 - 50 = 70, 34 -31 = 3, 14 -14 = 0

Basic Idea - Pattern Matching

We know what we want to match, so we take it and we create a template to match against.

Another way we can think of this is a filter.

But what operation do we want? Subtraction

1	2	3	4
2	3	4	1
3	4	2	2
4	5	3	2

3	4
4	2

Basic Idea - Pattern Matching

We know what we want to match, so we take it and we create a template to match against.

Another way we can think of this is a filter.

Basic Subtraction

1	2	3	4
2	3	4	1
3	4	2	2
4	5	3	2

3	4
4	2

-5	-5	-1	-11
-1	0	-4	-4
2	1	-4	-4
2	3	-2	-1

Basic Idea - Pattern Matching

Can anyone see any problems with this method?

Basic Idea - Pattern Matching

Can anyone see any problems with this method?

Does it work for all sizes?

Basic Idea - Pattern Matching

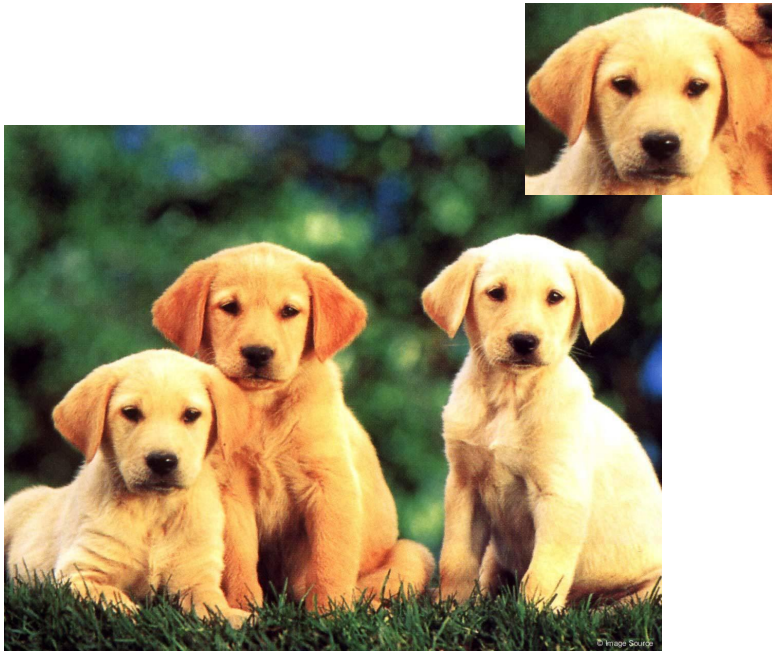
Can anyone see any problems with this method?

Does it work for all sizes?

What if I rotate the image?

Basic Idea - Pattern Matching

Can anyone see any problems with this method?



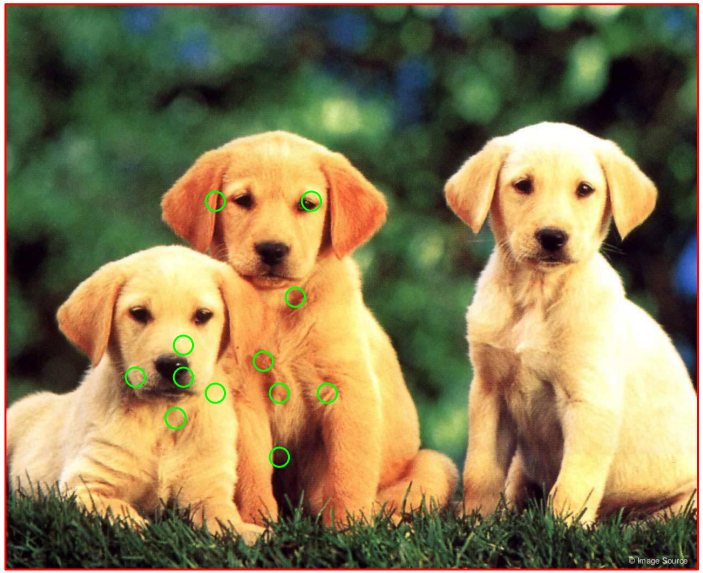
Does it work for all sizes?

What if I rotate the image?

What if there are very similar things in the image?

Feature Descriptors

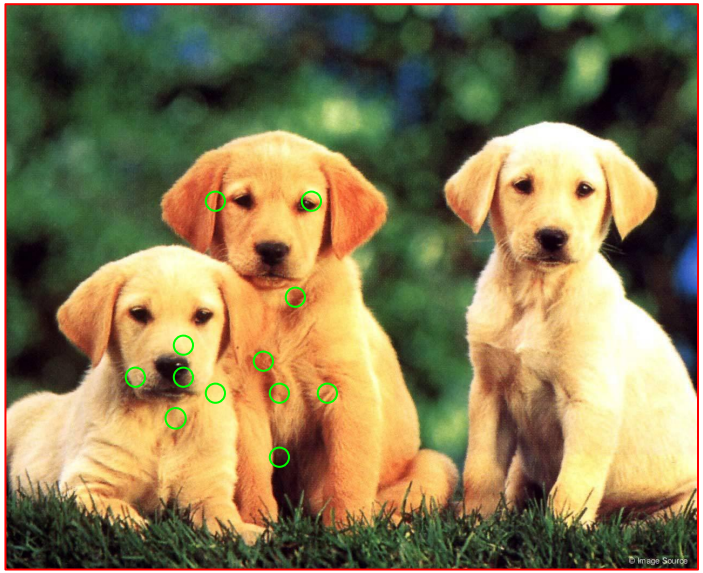
What features can we use to describe a dog?



Breed?

Feature Descriptors

What features can we use to describe a dog?

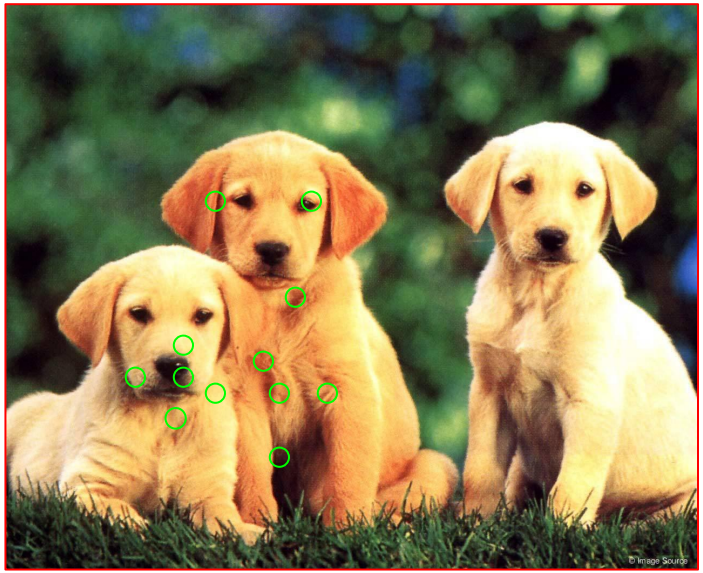


Breed?

Size?

Feature Descriptors

What features can we use to describe a dog?



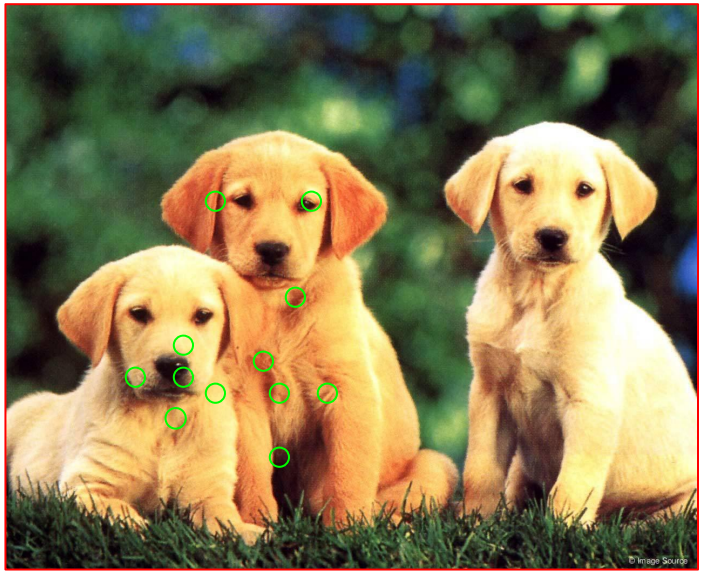
Breed?

Size?

Color?

Feature Descriptors

What features can we use to describe a dog?



Breed?

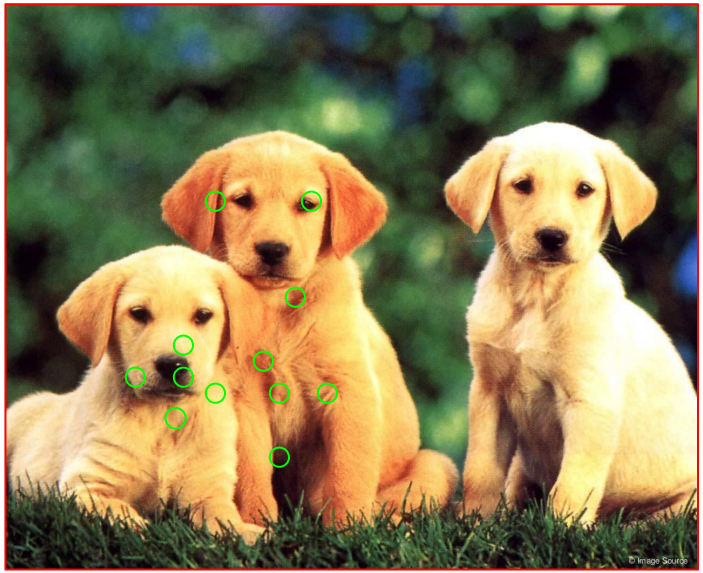
Size?

Color?

Type of hair/fur?

Feature Descriptors

How would a computer understand these features?



Breed?

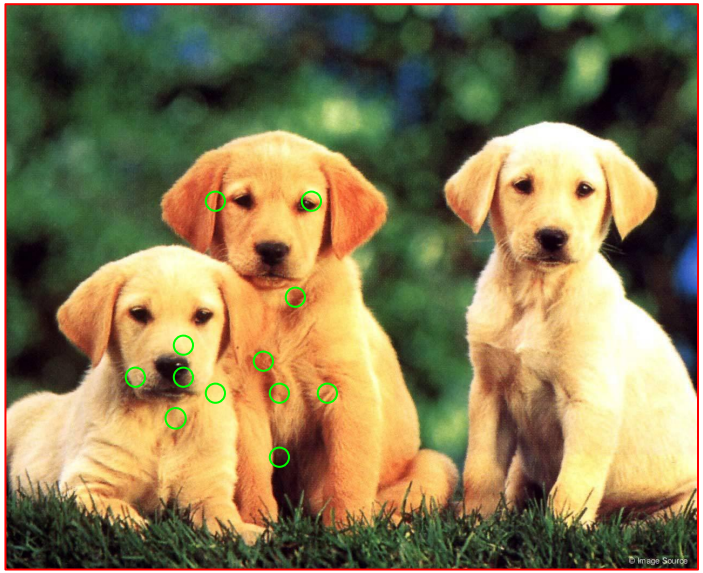
Size?

Color?

Type of hair/fur?

Feature Descriptors

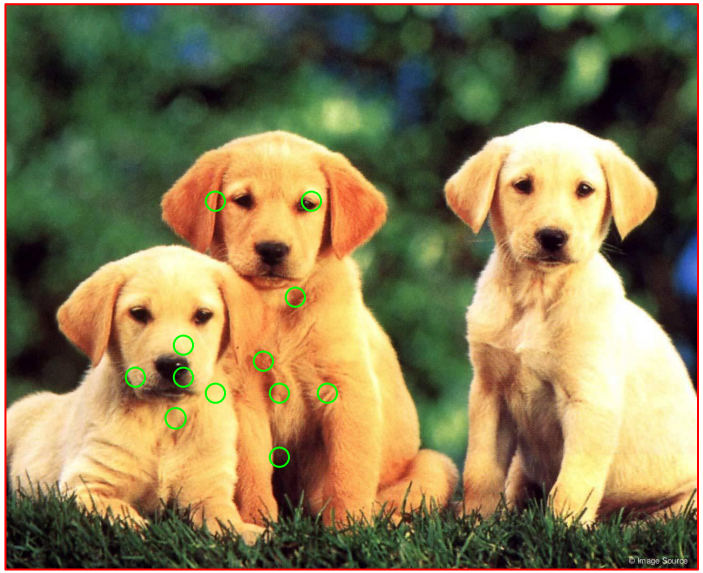
How would a computer understand these features?



Computers use math!

Feature Descriptors

How would a computer understand these features?

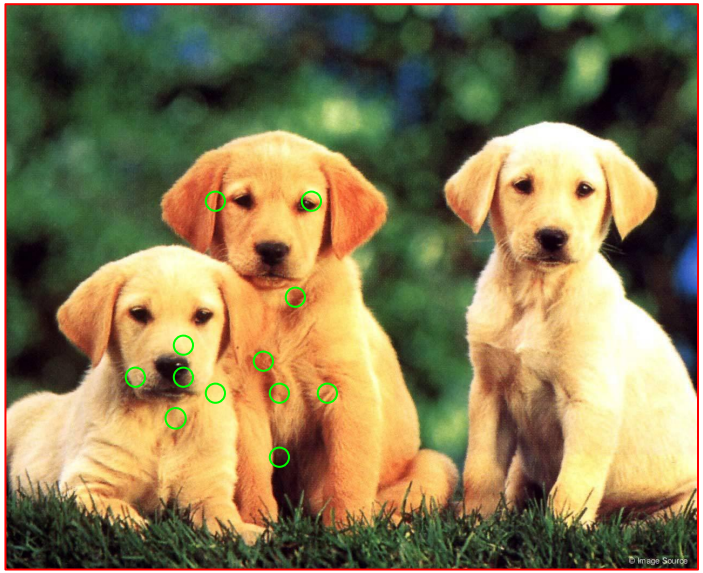


Computers use math!

Let’s have the computer create an equation to describe the things we want.

Feature Descriptors

How would a computer understand these features?



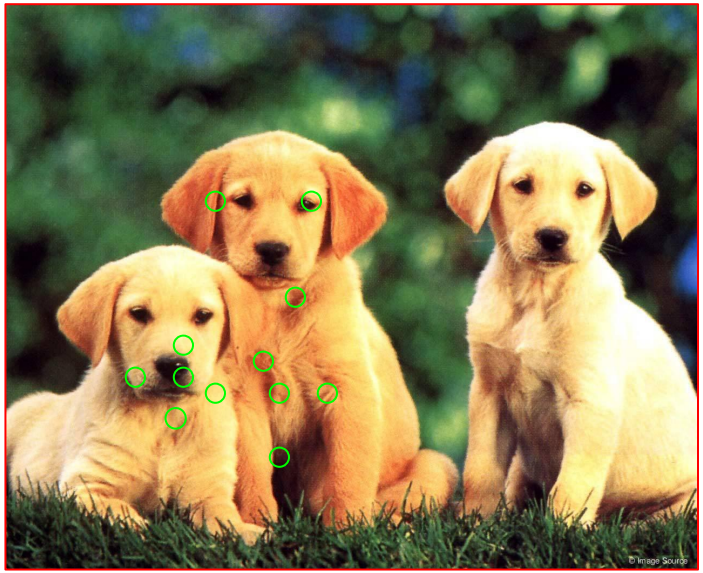
Computers use math!

Let's have the computer create an equation to describe the things we want.

This equation we will call a **…?**

Feature Descriptors

How would a computer understand these features?



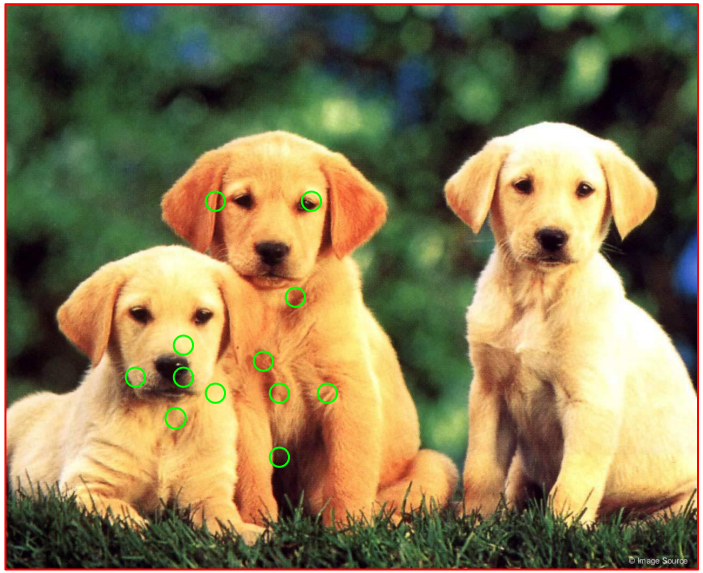
Computers use math!

Let's have the computer create an equation to describe the things we want.

This equation we will call a **FEATURE DESCRIPTOR**

Feature Descriptors

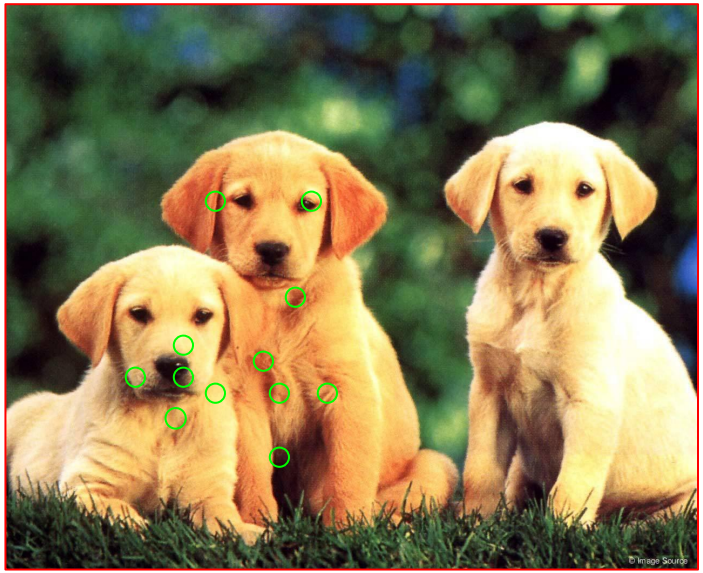
A mathematical description of a interest point.



19w + 24x + 43y + 2z etc..

Feature Descriptors

A mathematical description of a interest point.

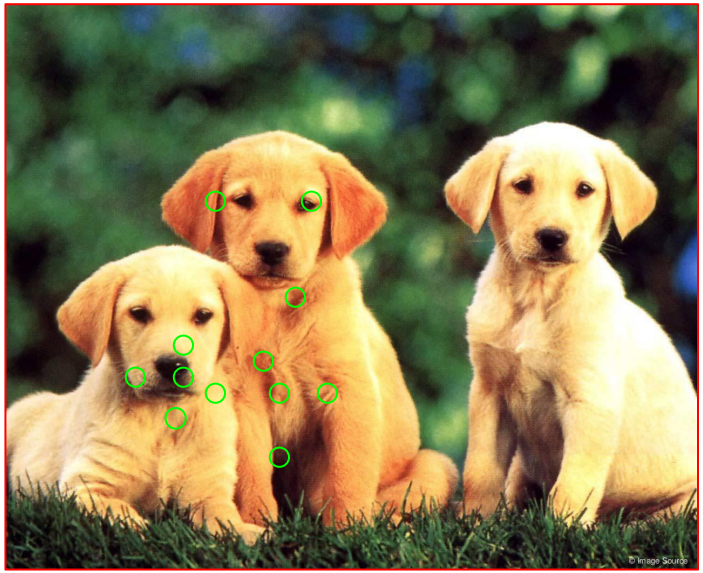


19w + 24x + 43y + 2z etc..

But we will make it really long, 64/128 items.

Feature Descriptors

A mathematical description of a interest point.



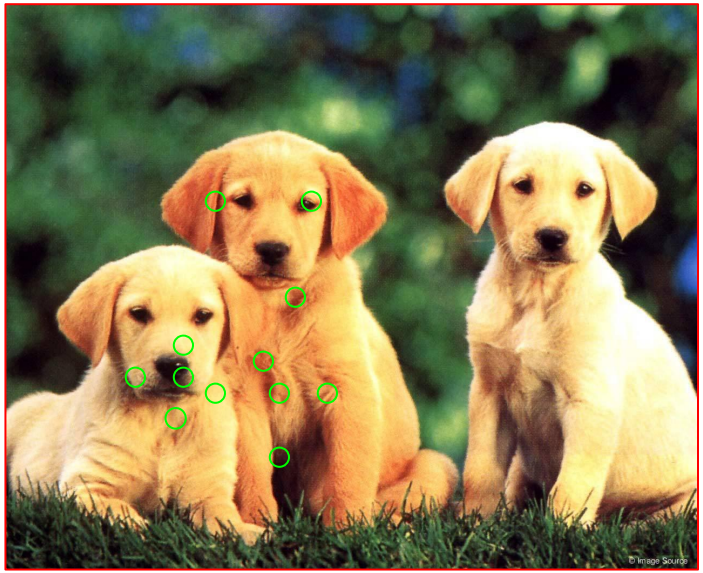
19w + 24x + 43y + 2z etc..

But we will make it really long, 64/128 items.

20a + 34b + 56c + 25d+ 46e + 78f+ …. + 19w + 24x + 43y + 2z

Feature Descriptors

A mathematical description of a interest point.



19w + 24x + 43y + 2z etc..

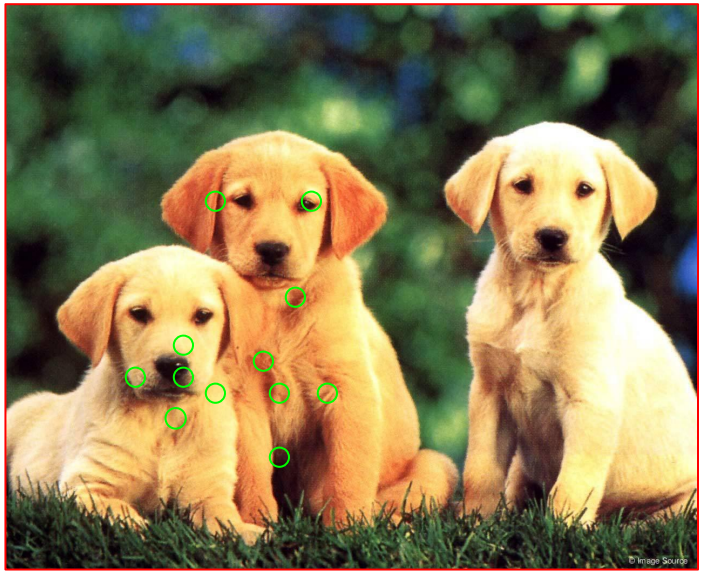
But we will make it really long, 64/128 items.

20a + 34b + 56c + 25d+ 46e + 78f+ …. + 19w + 24x + 43y + 2z …

And hope to have more than 1.

Feature Descriptors

A mathematical description of a interest point.



19w + 24x + 43y + 2z etc..

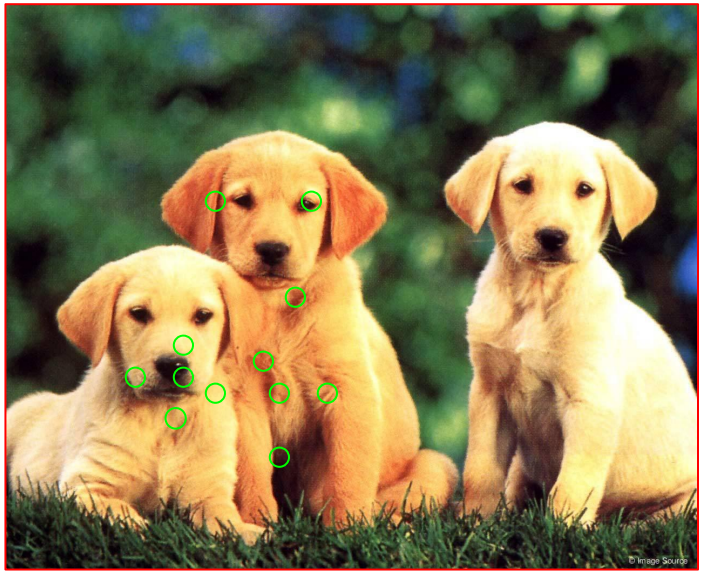
But we will make it really long, 64/128 items.

20a + 34b + 56c + 25d+ 46e + 78f+ …. + 19w + 24x + 43y + 2z …

And hope to have more than 1. Tens?

Feature Descriptors

A mathematical description of a interest point.



19w + 24x + 43y + 2z etc..

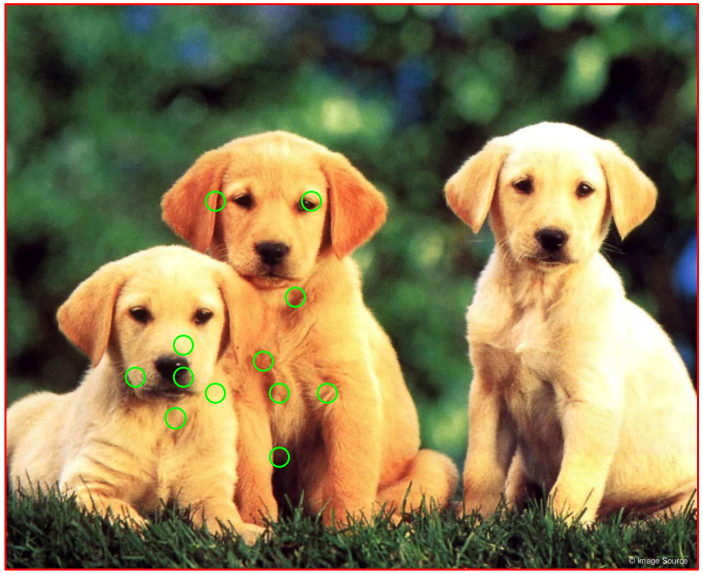
But we will make it really long, 64/128 items.

20a + 34b + 56c + 25d+ 46e + 78f+ …. + 19w + 24x + 43y + 2z …

And hope to have more than 1. Tens? hundreds?

Feature Descriptors

A mathematical description of a interest point.



19w + 24x + 43y + 2z etc..

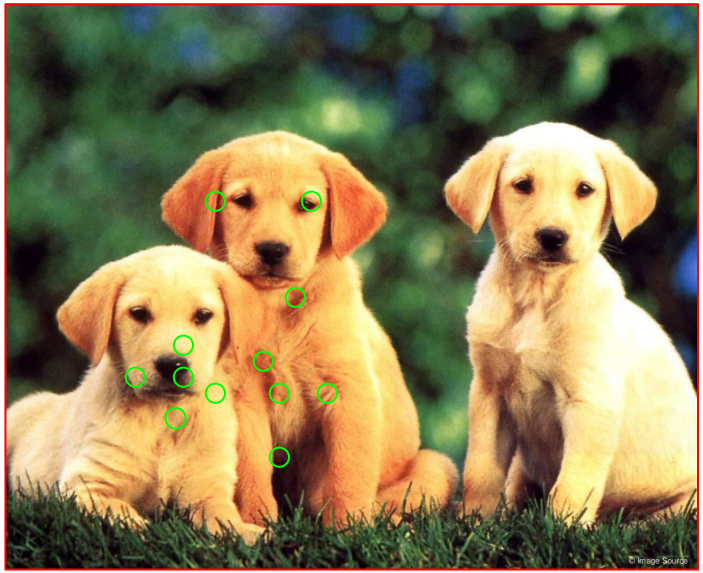
But we will make it really long, 64/128 items.

20a + 34b + 56c + 25d+ 46e + 78f+ …. + 19w + 24x + 43y + 2z …

As many as we can

Feature Descriptors

What if we can create a mathematical representation of various interesting points?



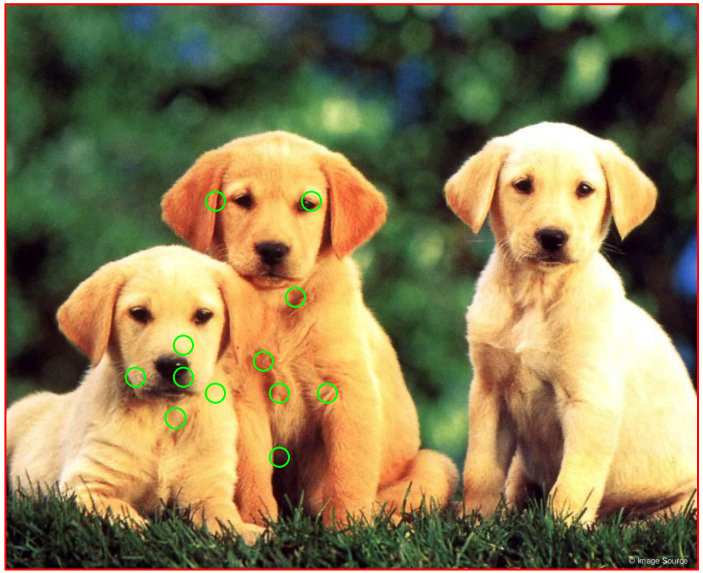
19w + 24x + 43y + 2z etc..

But we will make it really long, 64 items.

Now we can do the same type of differencing operation, but not have to worry about the minor things like, scale, rotation, etc....

Feature Descriptors

What if we can create a mathematical representation of various interesting points?



19w + 24x + 43y + 2z etc..

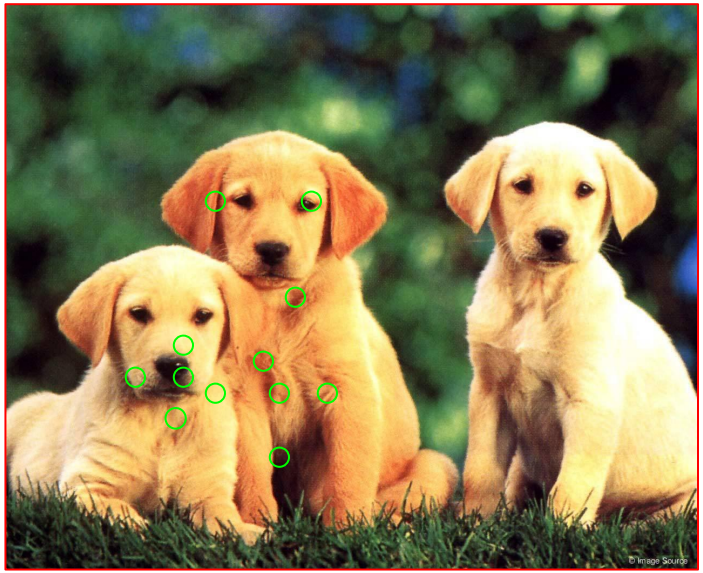
But we will make it really long, 64 items.

Now we can do the same type of differencing operation, but not have to worry about the minor things like, scale, rotation, etc....

Similarity is still a problem though.

Feature Descriptors

What if we can create a mathematical representation of various interesting points?



19w + 24x + 43y + 2z etc..

But we will make it really long, 64 items.

Now we can do the same type of differencing operation, but not have to worry about the minor things like, scale, rotation, etc....

Similarity is still a problem though. Or is it?

Feature Descriptors - Some Example of SURF

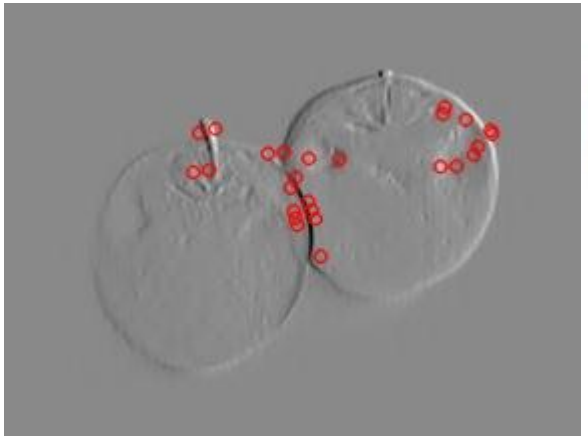
See if you can tell me what type of objects feature description is good on?

Feature Descriptors - Some Example of SURF

Feature Descriptors - Some Example of SURF

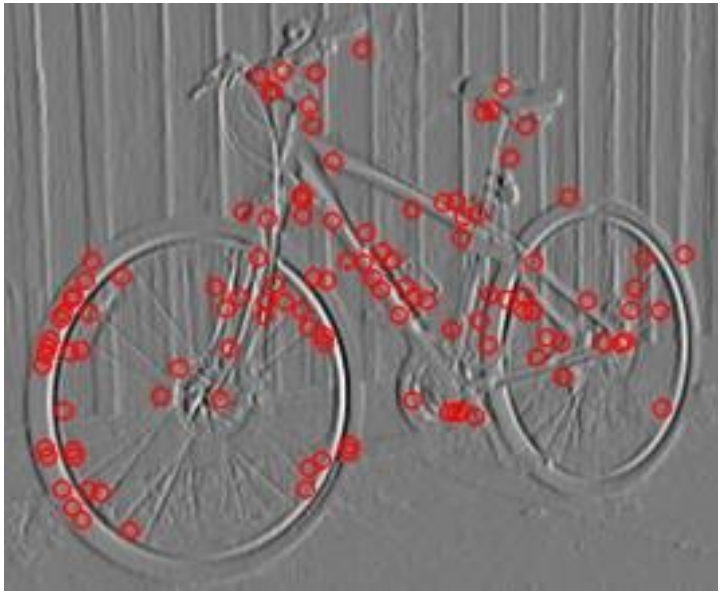
Feature Descriptors - Some Example of SURF

Feature Descriptors - Some Example of SURF



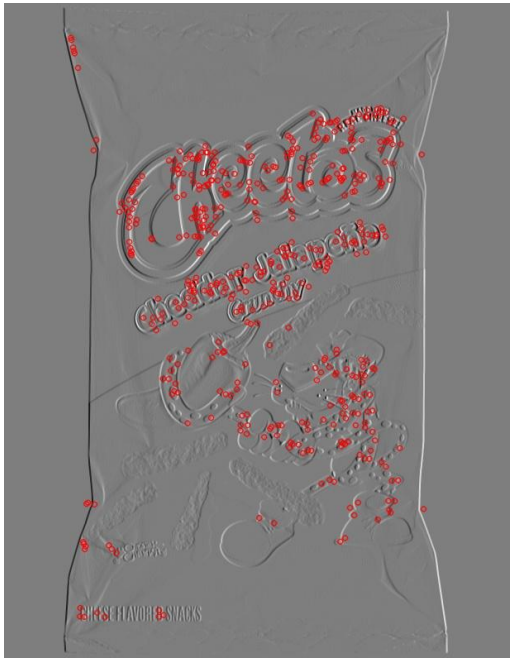
Feature Descriptors - Some Example of SURF

Feature Descriptors - Some Example of SURF



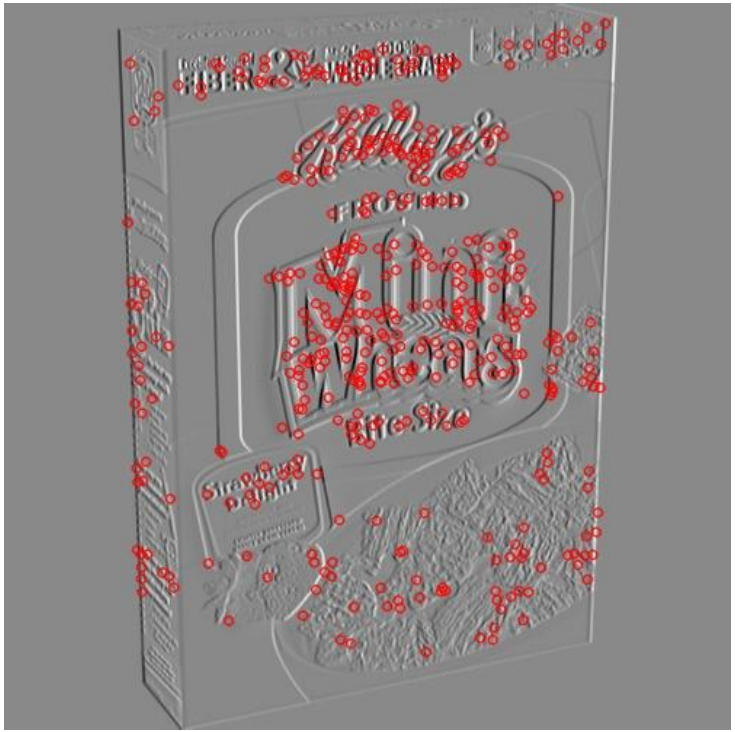
Feature Descriptors - Some Example of SURF

Feature Descriptors - Some Example of SURF



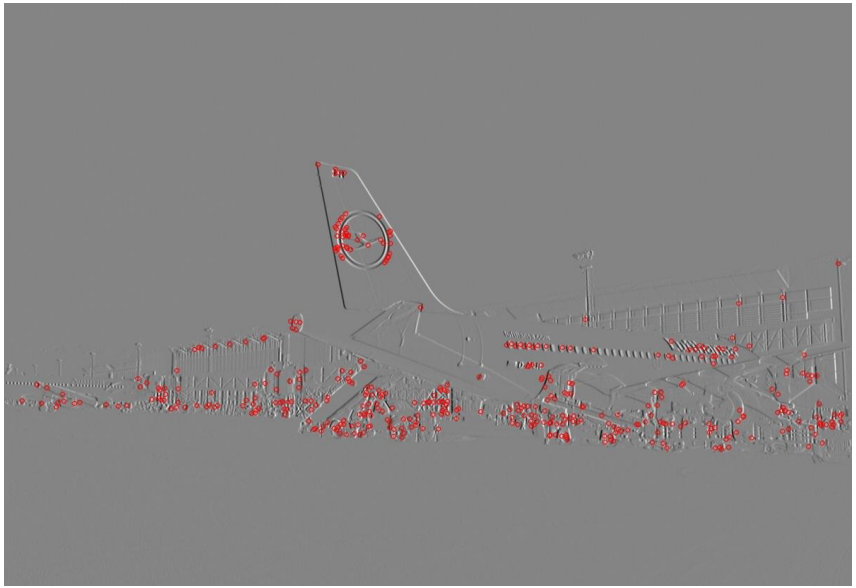
Feature Descriptors - Some Example of SURF

Feature Descriptors - Some Example of SURF



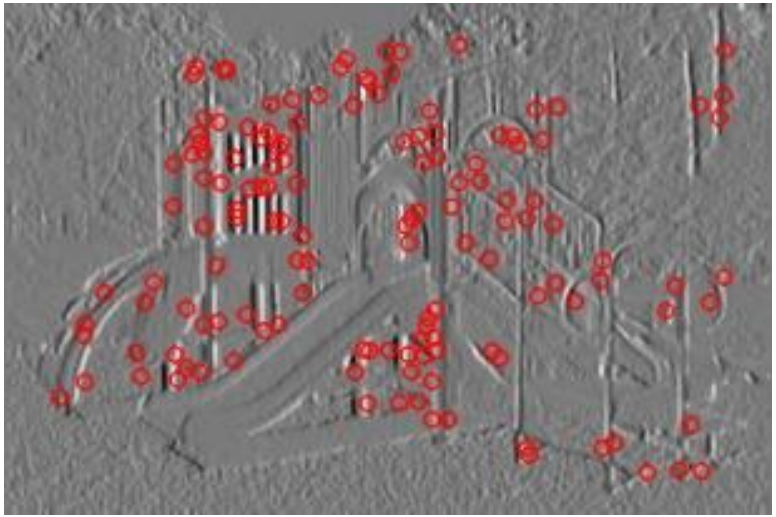
Feature Descriptors - Some Example of SURF

Feature Descriptors - Some Example of SURF



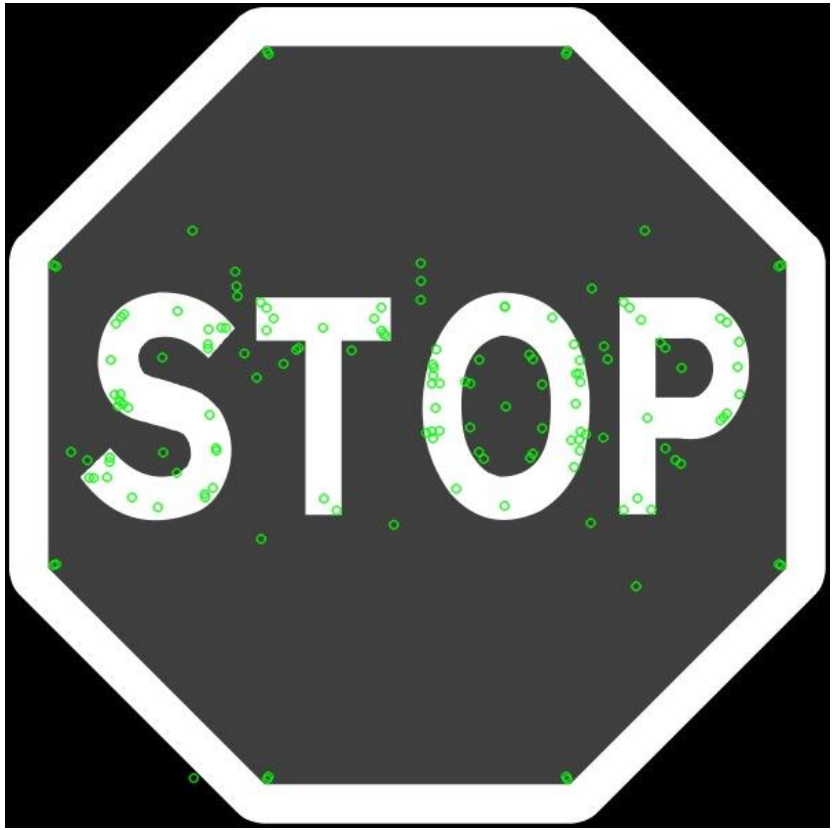
Feature Descriptors - Some Example of SURF

Feature Descriptors - Some Example of SURF

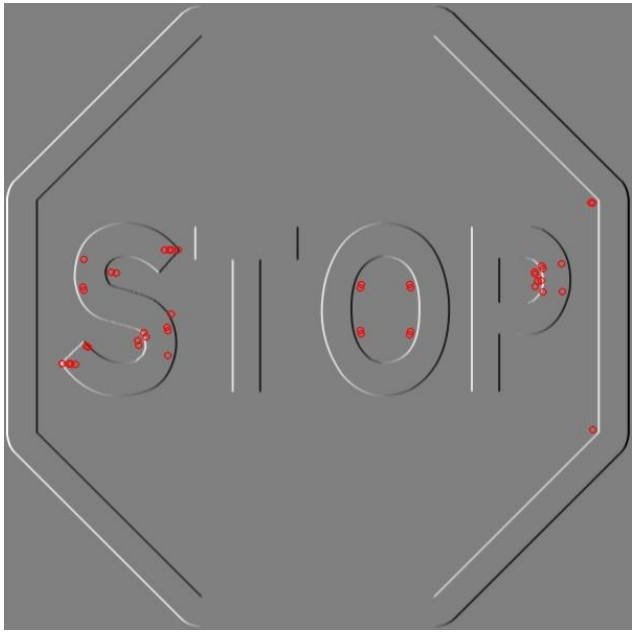


Feature Descriptors - Some Example of SURF

Feature Descriptors - Some Example of SURF



Feature Descriptors - Some Example of SURF



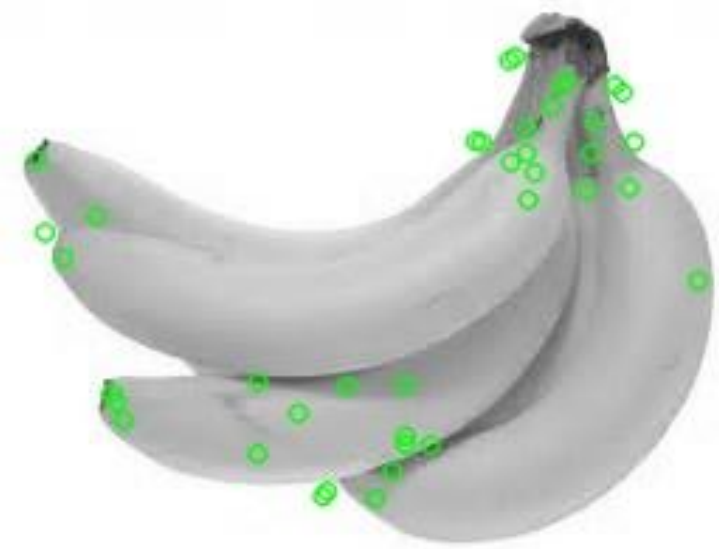
Feature Descriptors - Some Example of SURF

Feature Descriptors - Some Example of SURF

Feature Descriptors - Some Example of SURF

Feature Descriptors - Some Example of SURF

Feature Descriptors - Some Example of SURF



Feature Descriptors - Where do they work?

How do we find these interesting points?

Feature Descriptors - Where do they work?

How do we find these interesting points?

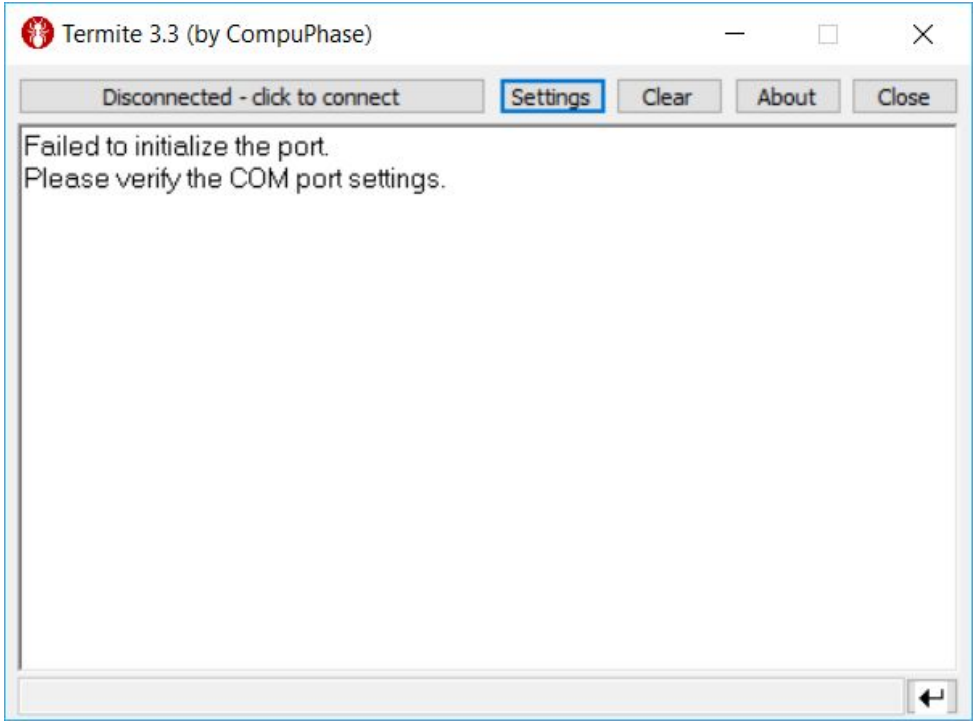
Edges!

Activity

1. Plug in your small cameras into the computers
2. On the desktop double-click the AMCap
3. Also go into the termite folder and in the bin folder open termite.exe
4. In termite.exe

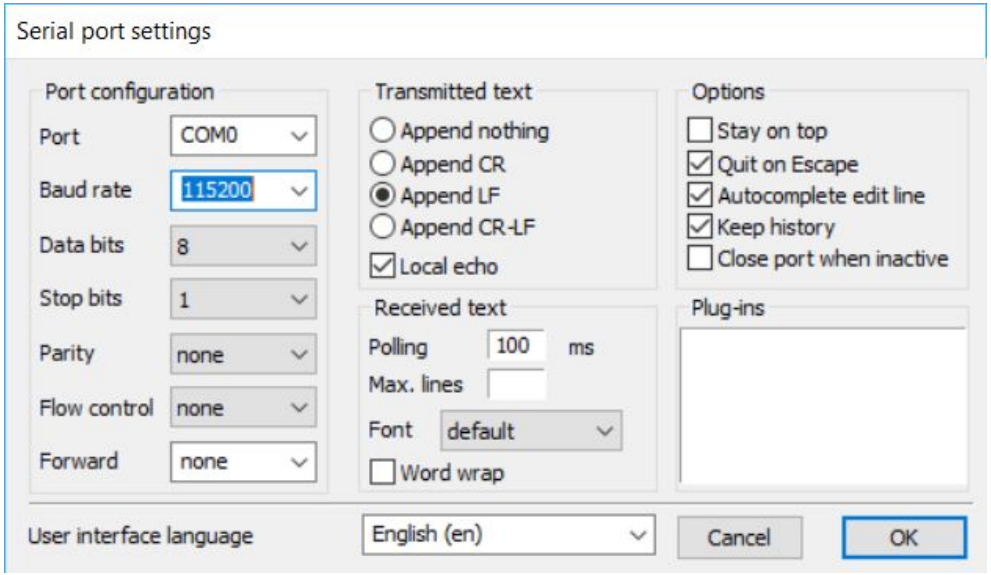
Activity

1. Click Settings
2.



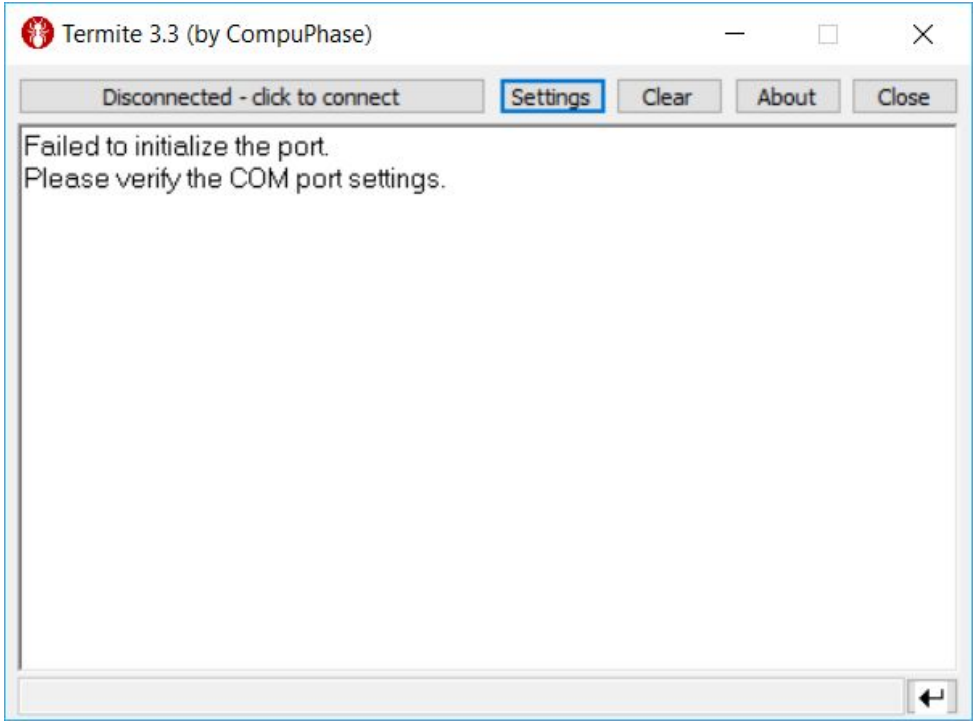
Activity

1. Change the COM Port to be whatever number shows up (shouldn't be COM0)
2. Change the Baud Rate to 115200



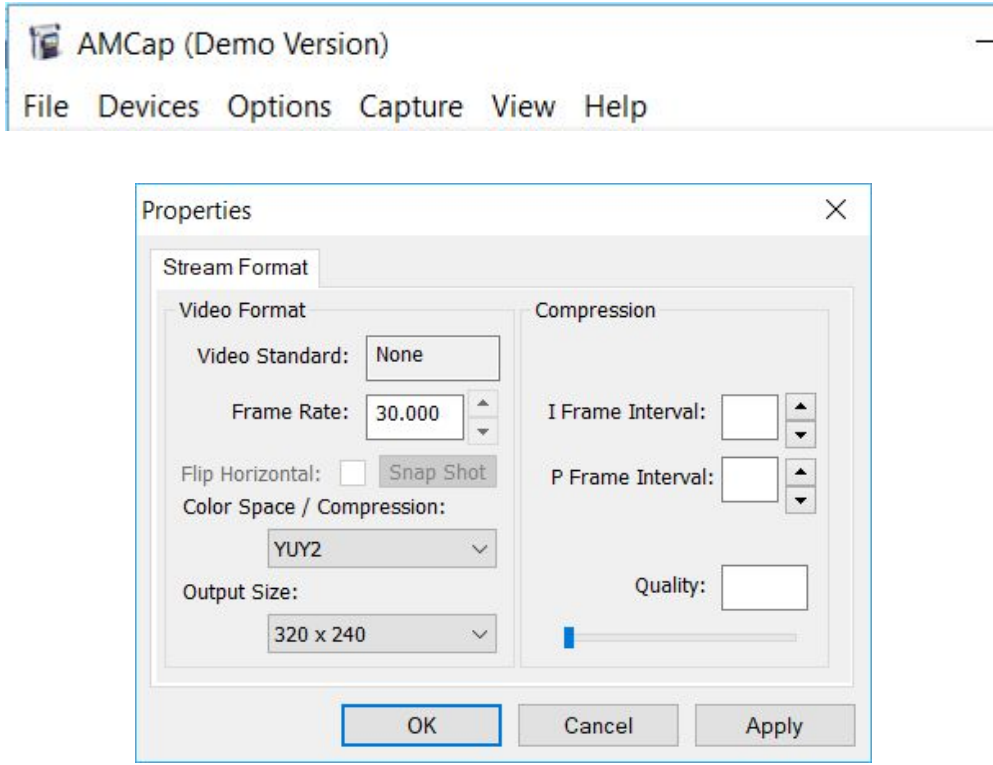
Activity

1. If not connected click to button to connect.



Activity

1. In AMCap
2. Under the Devices the second device
3. Under Options > Video Device > Capture Format
4. Set FPS to 30.00
5. Color Space / Compression to YUY2
6. Output size to 320x252



Commands to run in Termite

In the text box window

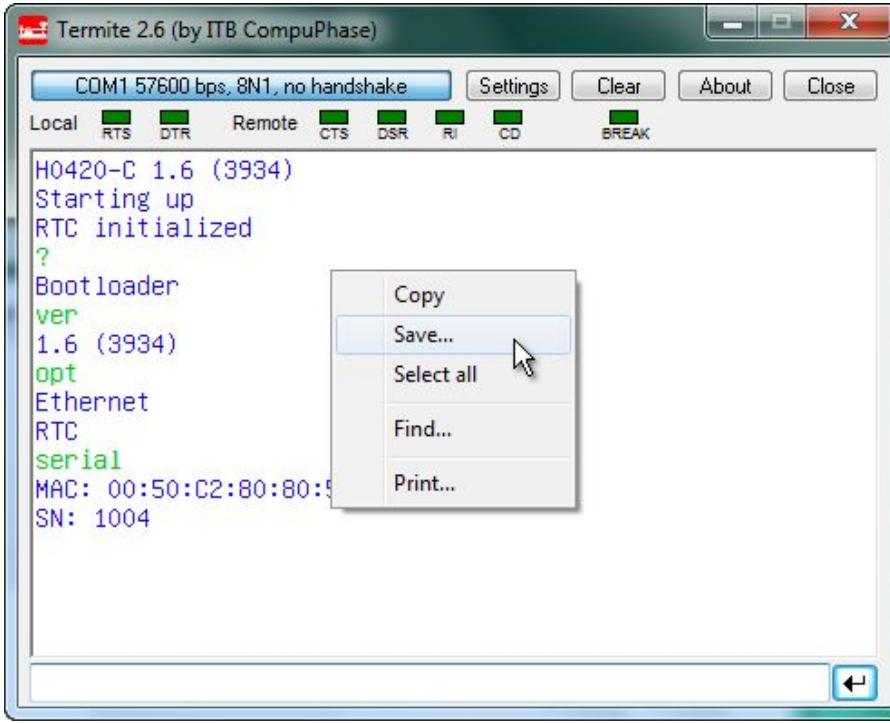
Type

```
setpar showwin true
```

To save a model/Template :

```
save model-name
```

Save various objects, items, and see what it will now recognize?



Activity Recap

What types of objects worked best?

What things affect accuracy?

What happened to the speed when you trained more objects?

What to do when we don’t know what features will work!