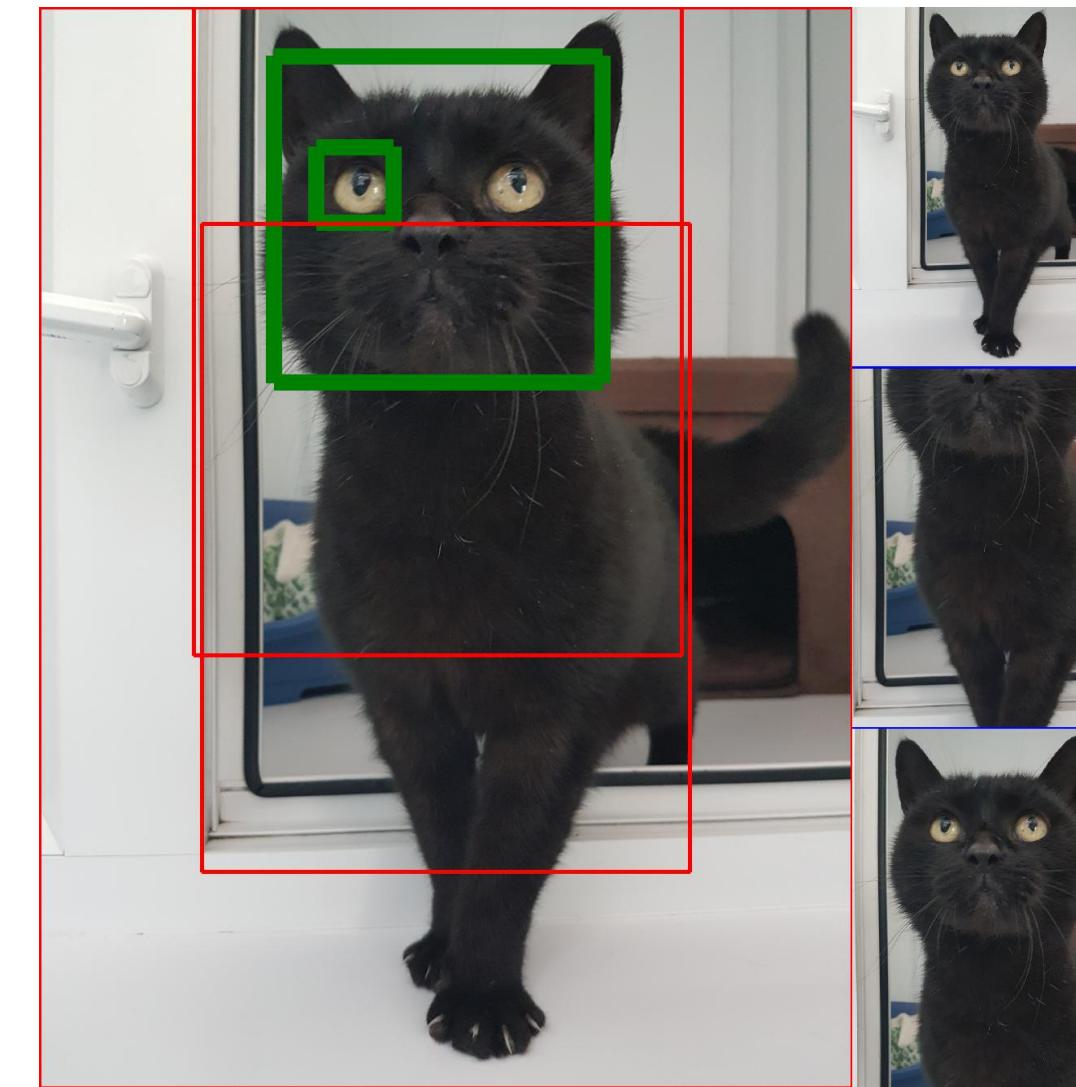


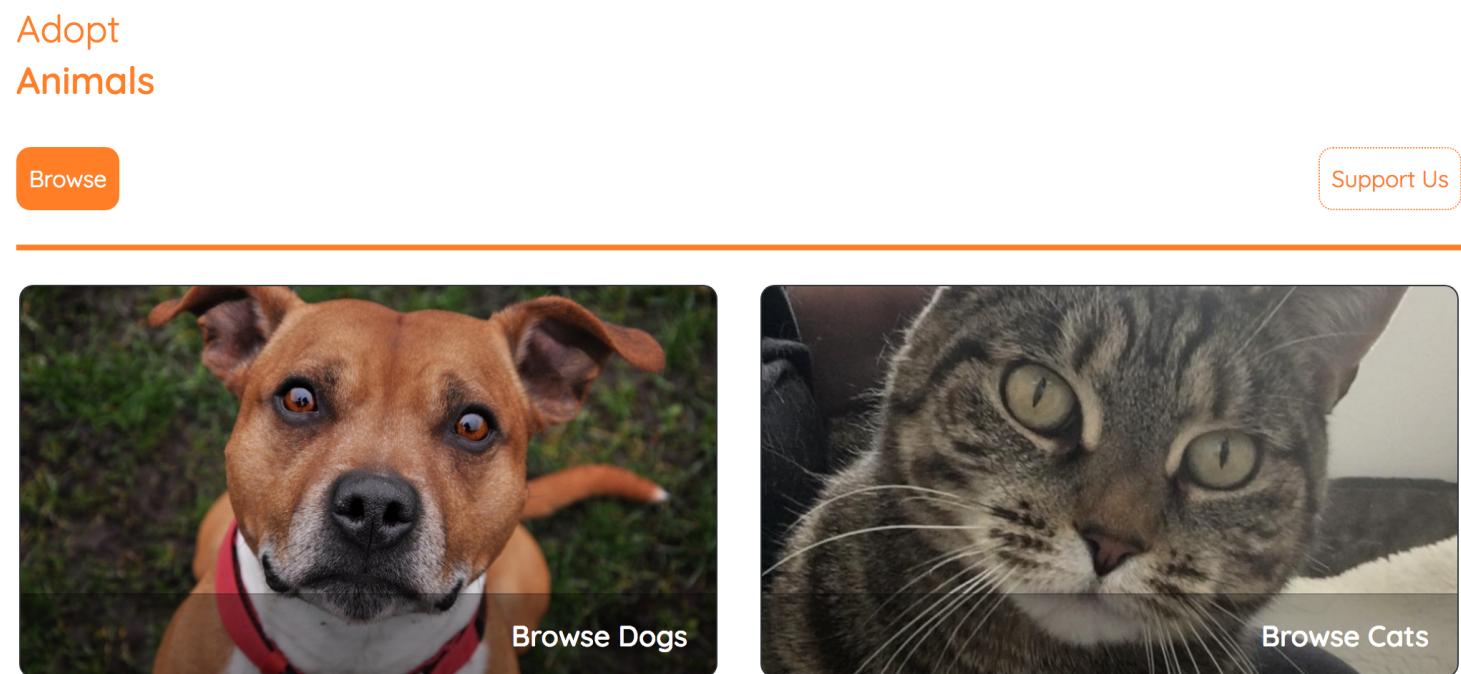
Whiskers, Ears and Puppy Dog Tails: finding out what animal pictures are made of



Who am I? Why am I doing this?

-  Mike Moran, Engineer at Skyscanner
- Adopt Animals
- Skyscanner “One Day”
- A general Machine Learning interest

Context: Adopt Animals



What is Adopt Animals?

We're building new ways to help animals in shelters find their loving forever homes.



– Adopt Animals, by Kale charity

What were we trying to do?

"Often, the images that we get are going to be slightly off-centre w.r.t. the subject, which makes the smaller previews look odd - I'll frame my thoughts as questions then statements:

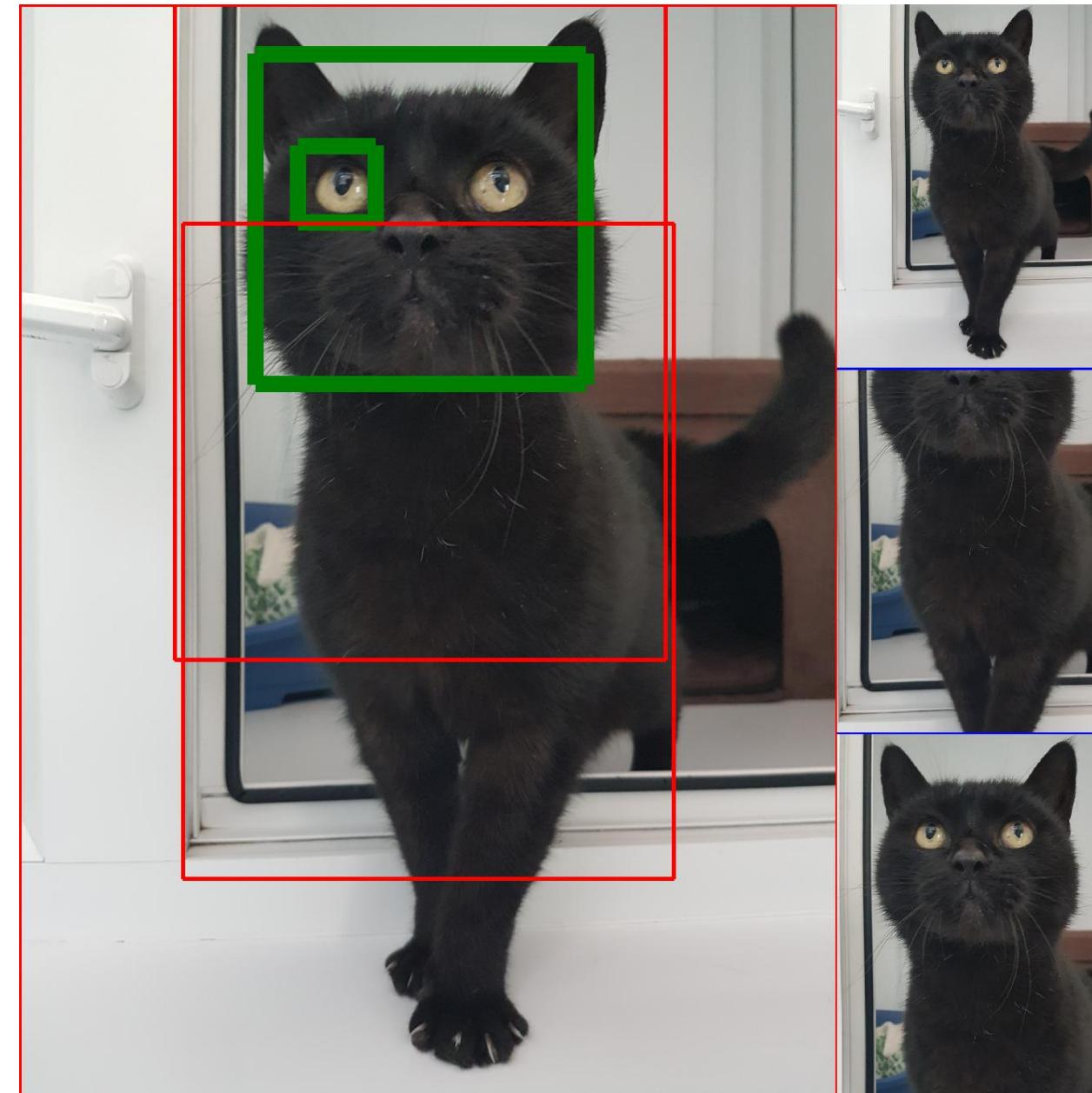
- Do ML models exist that can find the rough position of an animal's face?
- Is it per species (I'm guessing so)?
- How do I run it as part of an aggregation workflow? And what's the computational cost? Would it run on a small-ish Droplet?
- I don't need any "internal" details mapped, just where their face is, as a bounding box or something
- It doesn't have to be super perfect - more for nudging the frame in the right direction
- I can get a bunch of reference material, in the form of some snapshots of EDCH images over whatever time frame you'd like
- I can't promise to ship it immediately but it sounds like a fun side-project at least 😊"

– (Sky of Adopt Animals)

My constraints

- I ~~am lazy~~ have limited time, so an off-the-shelf solution would be good
- I am *not* a Jupyter expert (yet?)
- I have a non-uniform experience of Data-science algorithms

Ok, so how do I get to this?



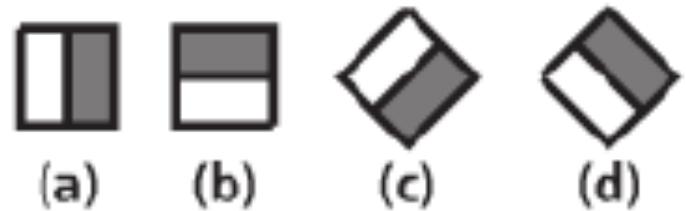


Going with something I
know: Viola Jones face
detection

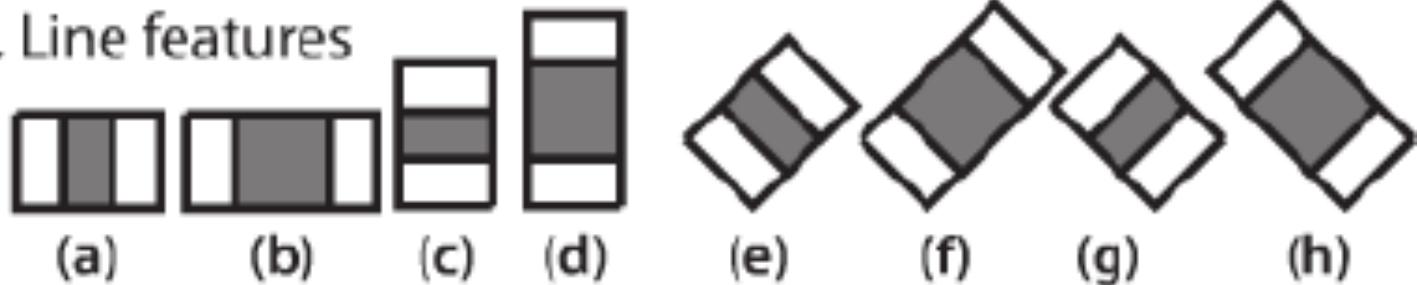
– from “Looking for Bobby but found
Paris instead”

Training: Haar features

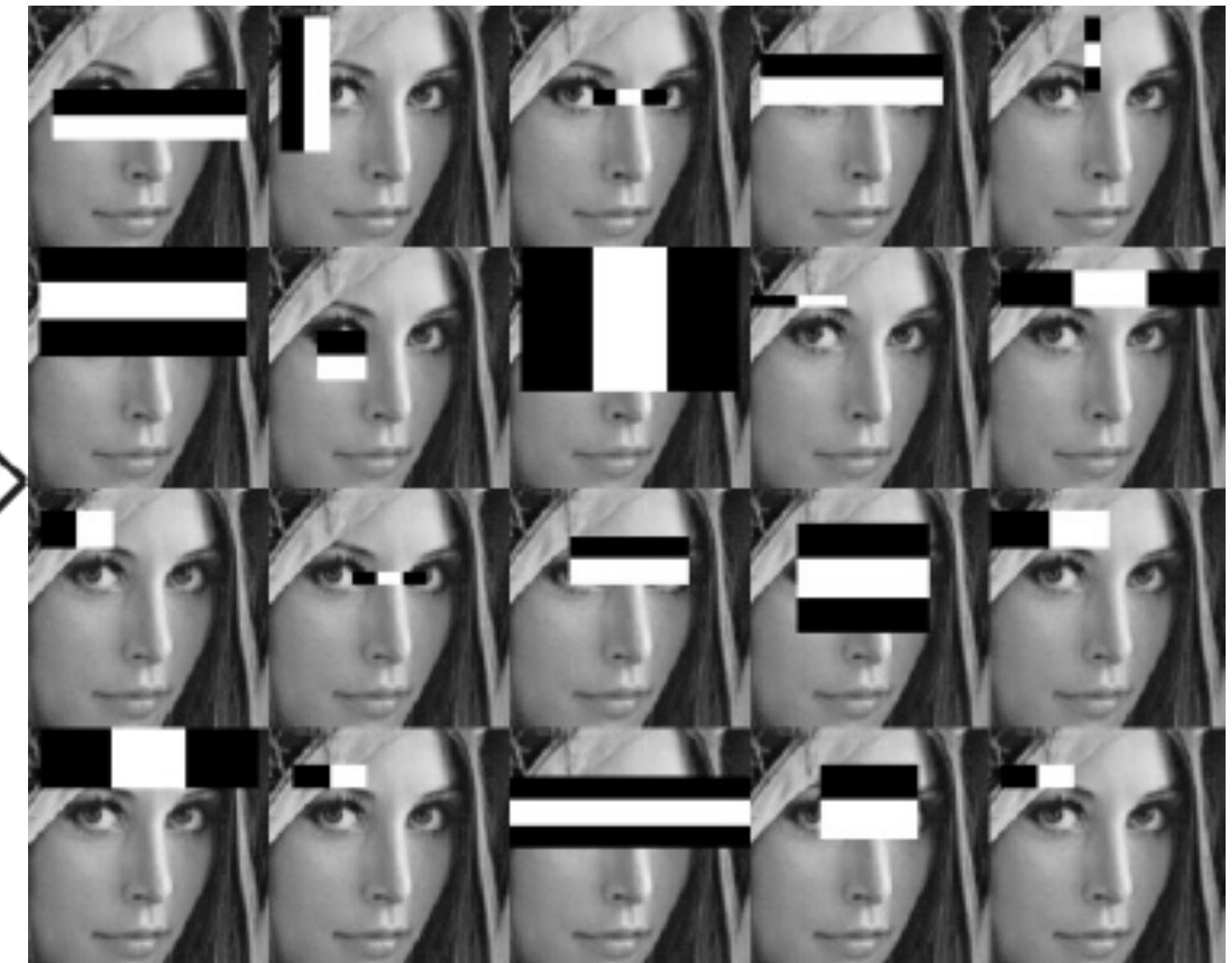
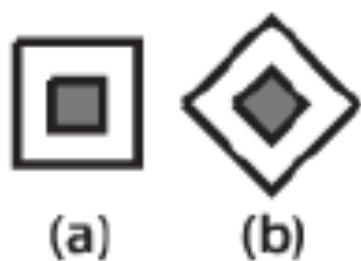
1. Edge features



2. Line features



3. Center-surround features

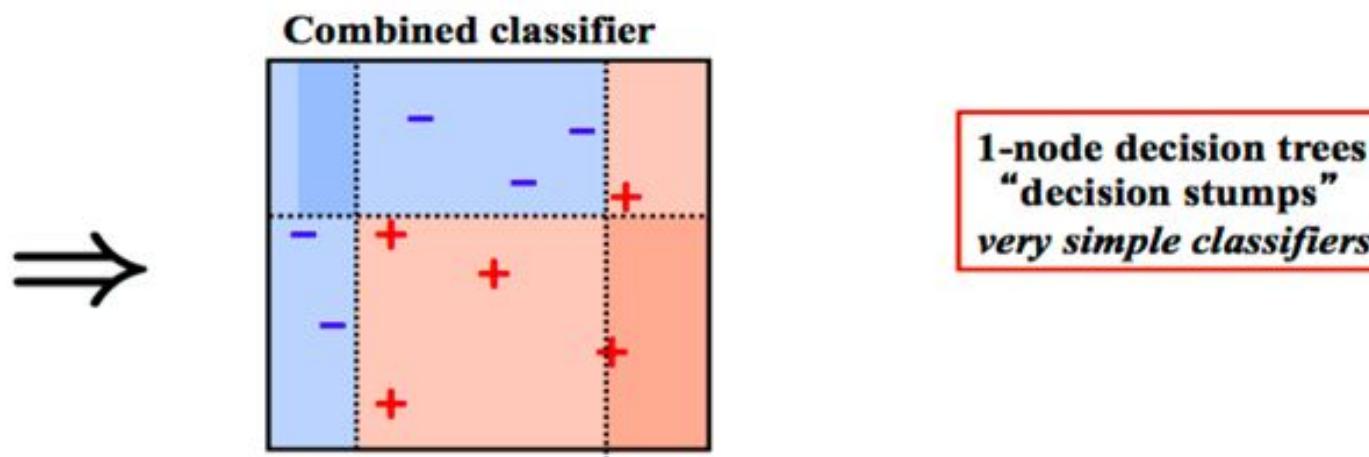


Training: Adaboost

Algorithm Adaboost - Example

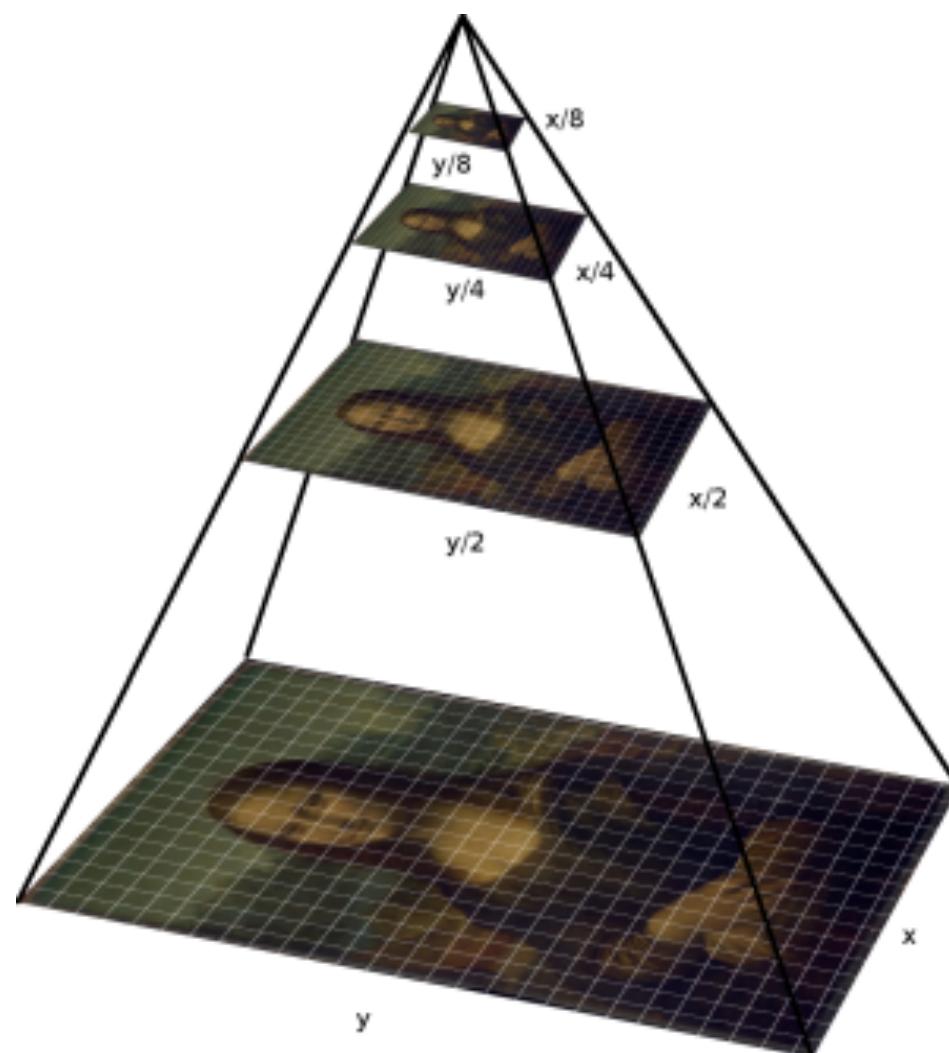
Weight each classifier and combine them:

$$.33 * \boxed{\text{blue}} + .57 * \boxed{\text{blue} \mid \text{orange}} + .42 * \boxed{\text{blue} \mid \text{orange}} \geq 0$$



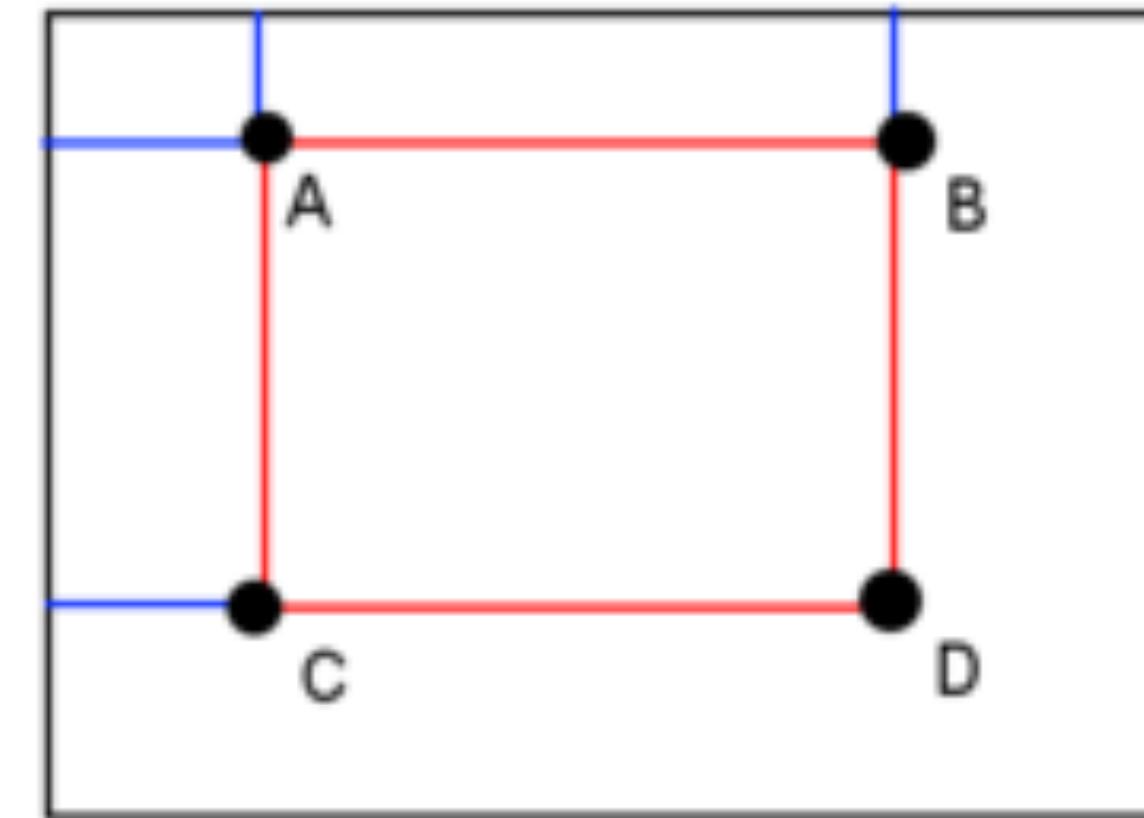
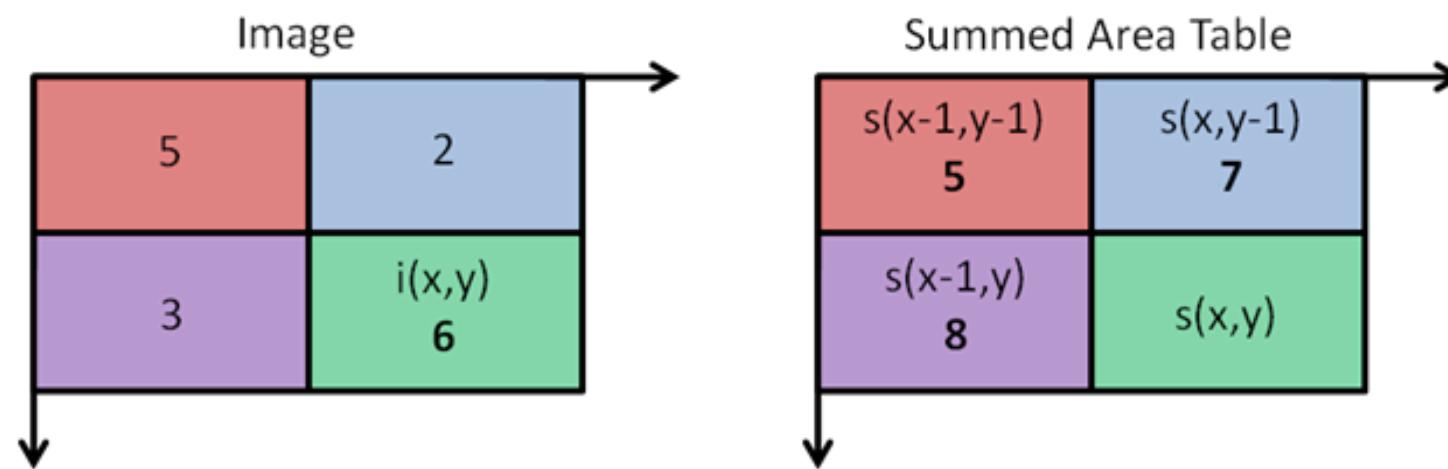
Applying: Image Pyramid

- deals with varying feature size



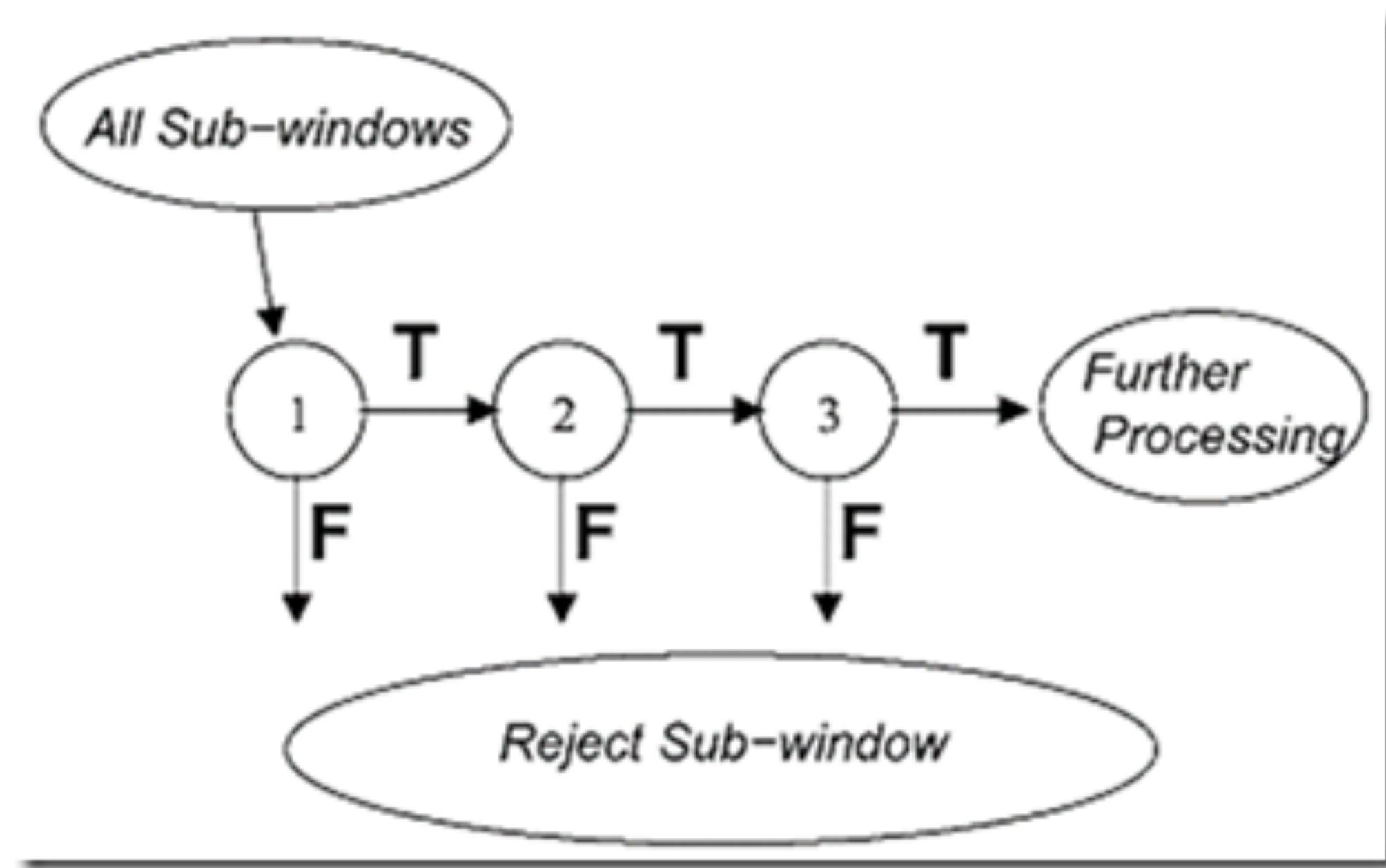
Applying: Integral Image

- efficiently evaluates hear features; $O(n^2) \rightarrow O(1)$



$$\text{Sum} = D - B - C + A$$

Applying: Haar cascade



Visualising applying to image

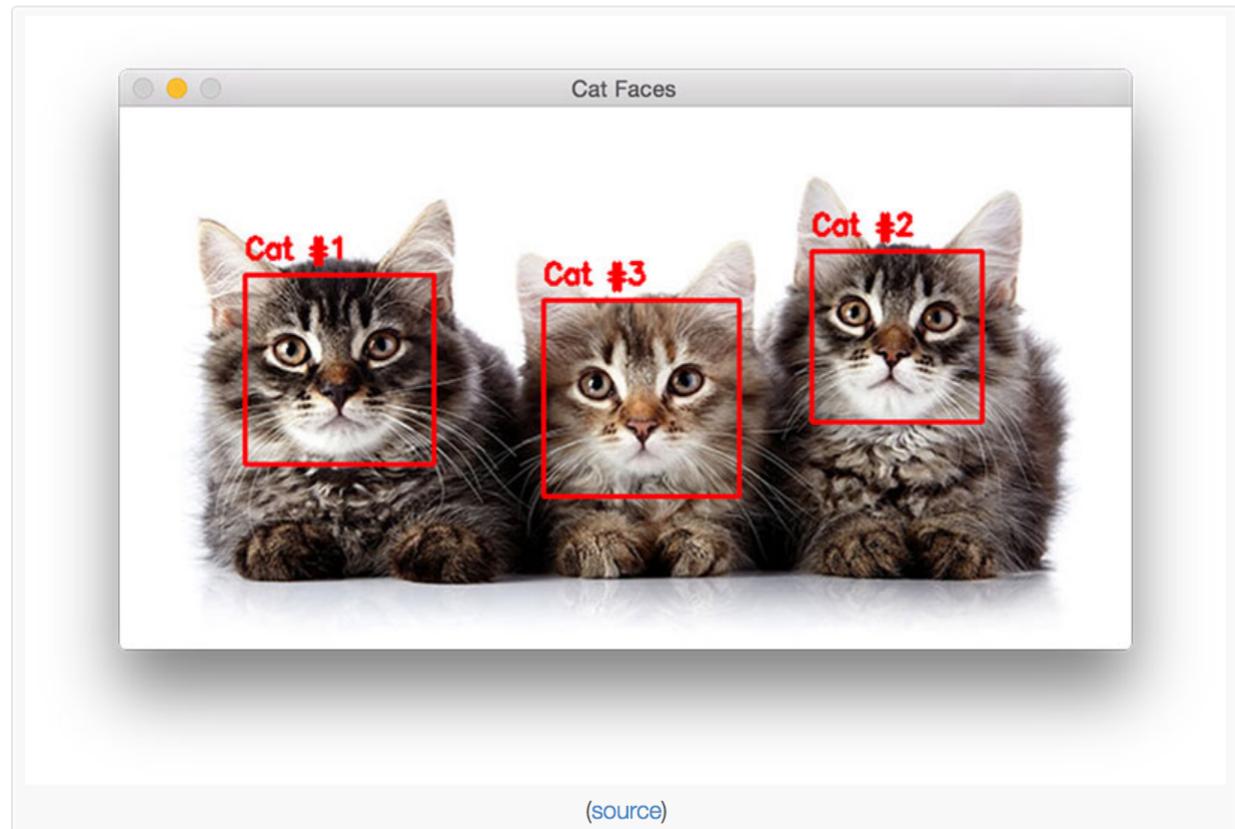


[YouTube visualisation by Adam Harvey](#)

Re-using an off-the-shelf model

Detecting cats in images with OpenCV

by Adrian Rosebrock on June 20, 2016 in Object Detection, Tutorials

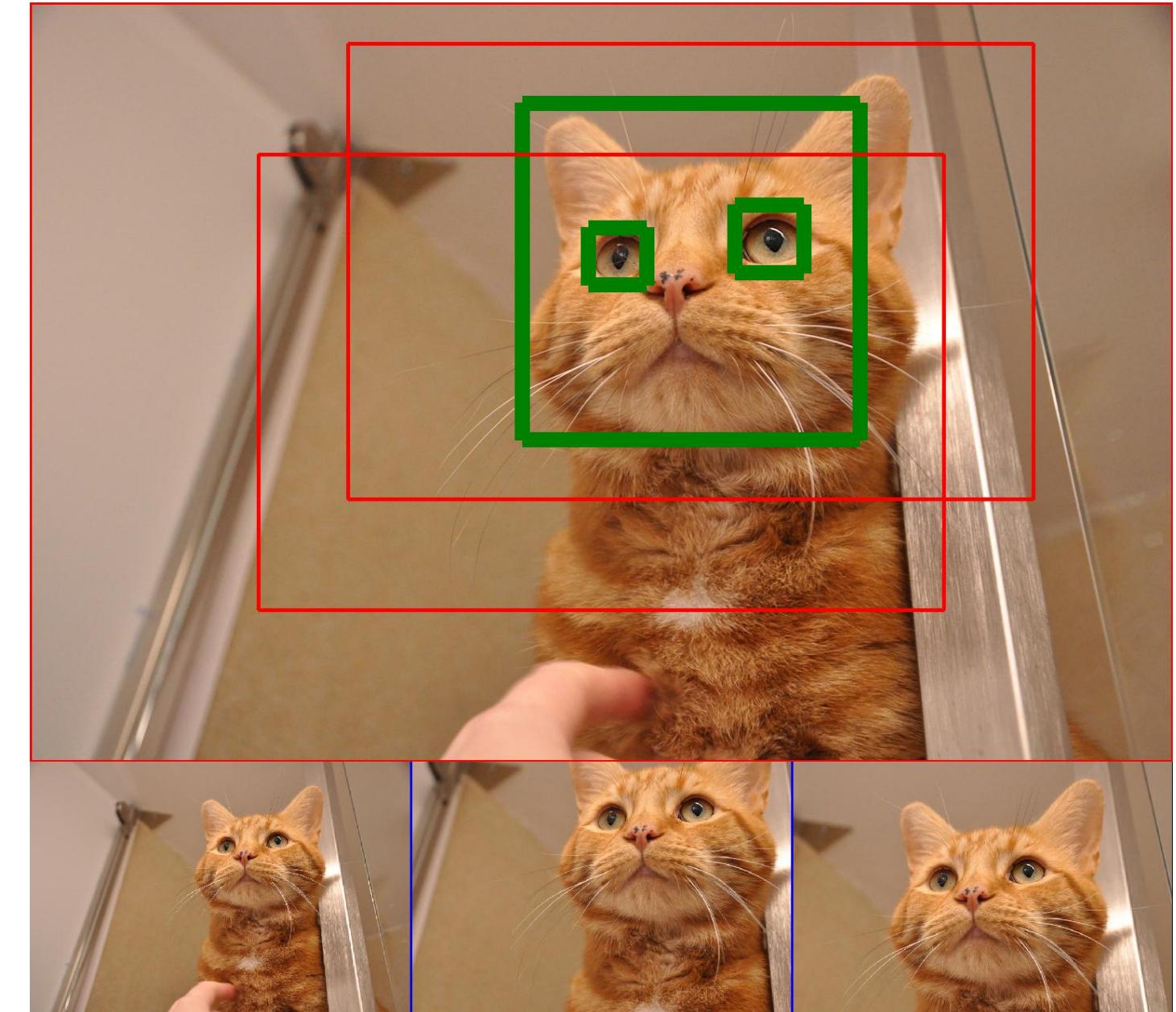
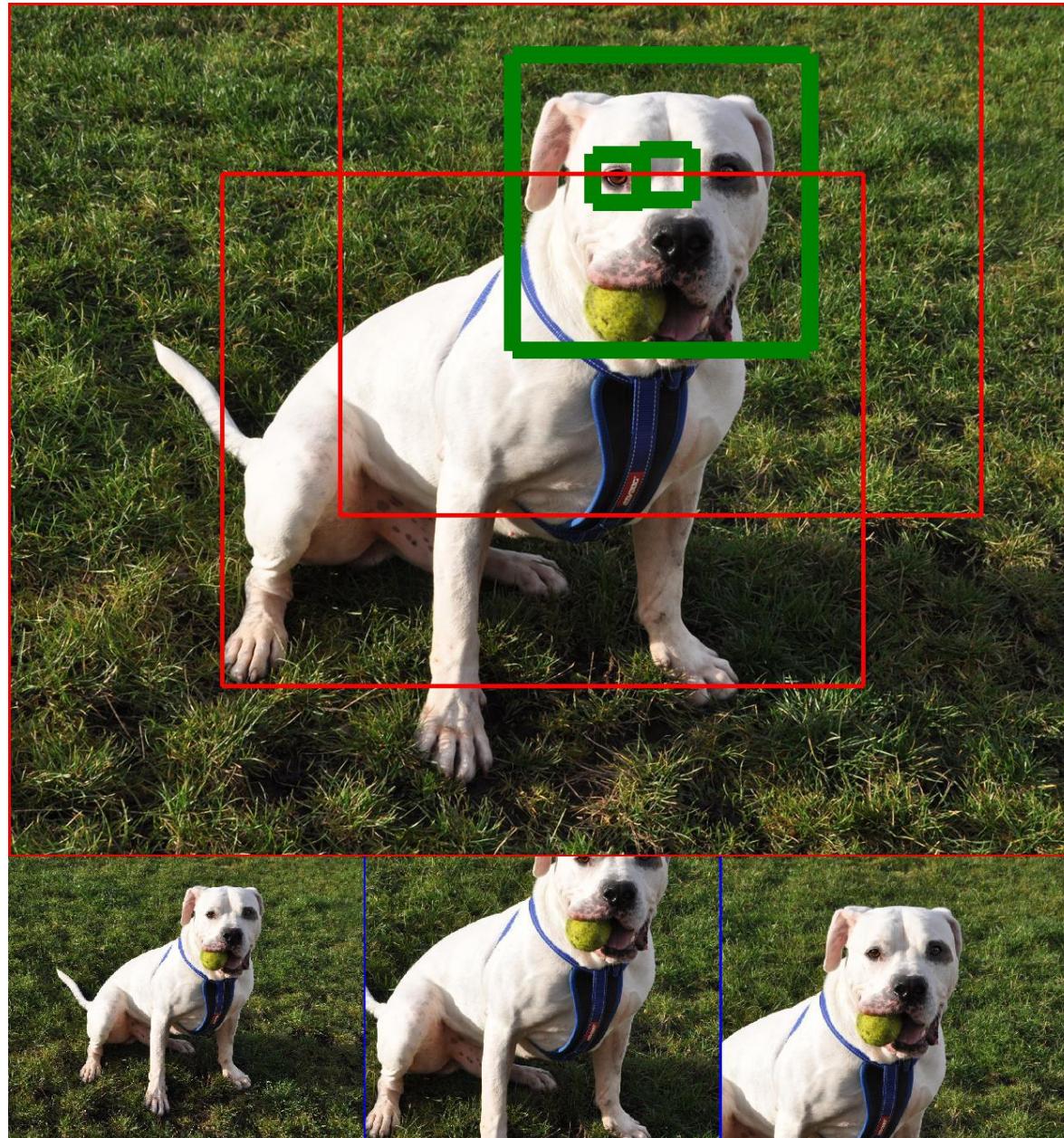


(source)

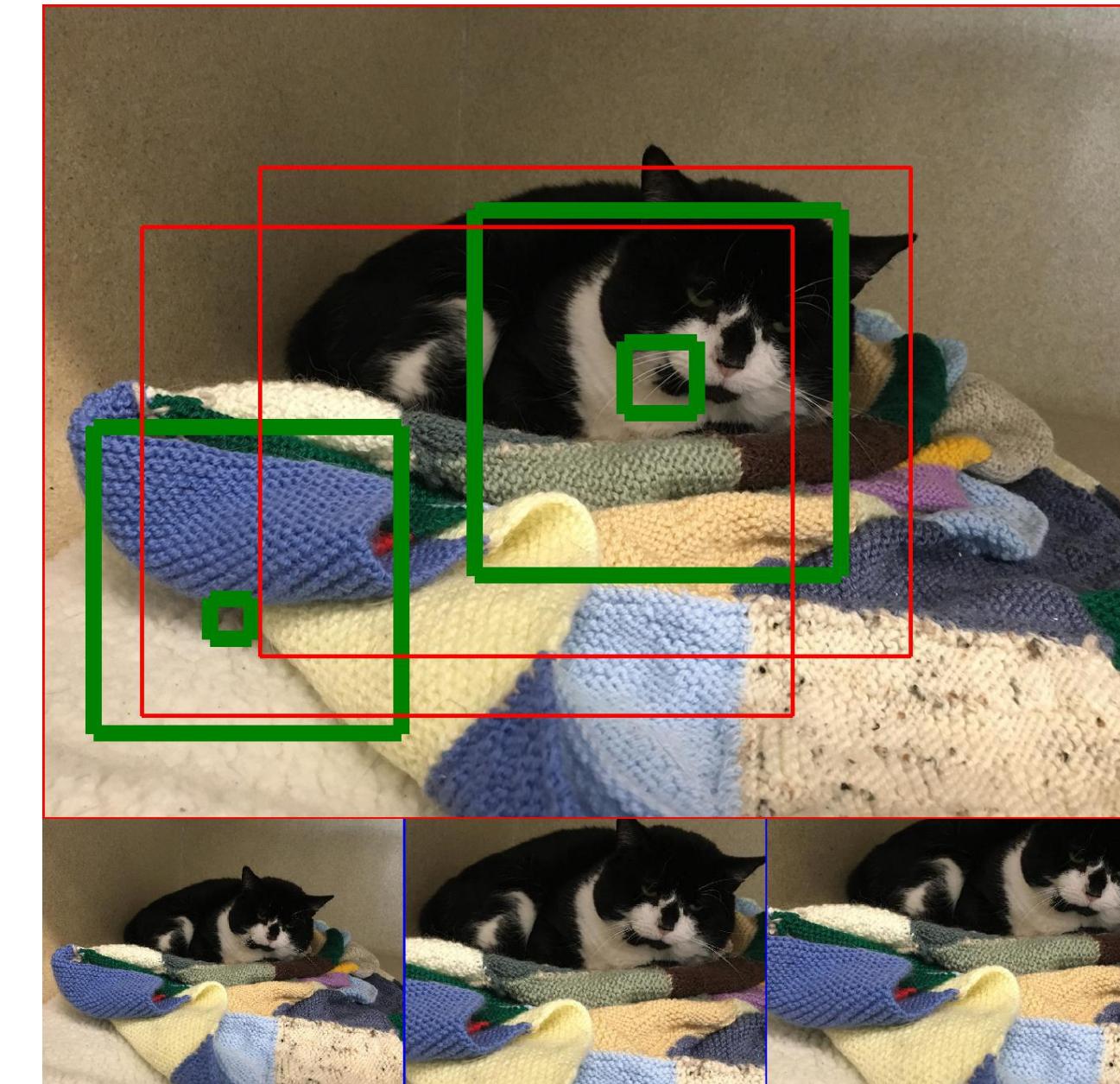
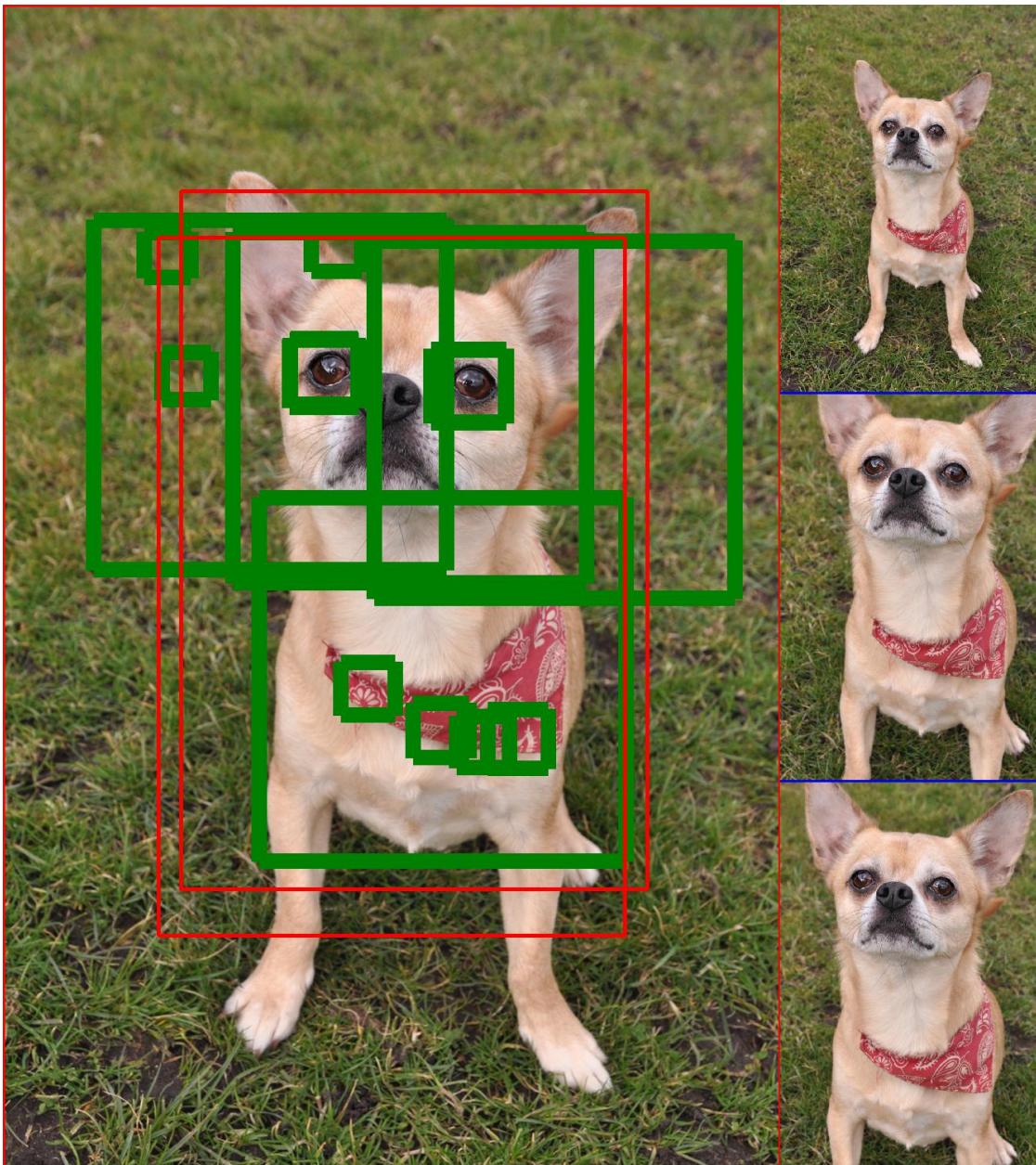
Did you know that OpenCV can detect cat faces in images...**right out-of-the-box with no extras?**

from [pyimagesearch](#), using a pre-trained cat model from [Joseph Howse](#)

Where have I gotten to in this? Job done, ye?



Not so fast



Visualising varying min_size

Learnings / Reinforcements

- Maybe too lazy in not focussing on understanding e.g. playing with scale-size instead of max-size parameters. Became clear once I actually spent some time on it.
- Some parameters still seem like magic (min neighbours = 5)
- You don't have to be an expert to be helpful

Backing up: what next?

- Productionising the cats-only thumbnails
- Pre-learned Dogs?
- Auto-learning of parameters on top of same algorithm
- #HASHTAG DEEP LEARNING?
 - Deep learning is expensive, both in terms of cpu cost and collecting of training data, is it worth it?

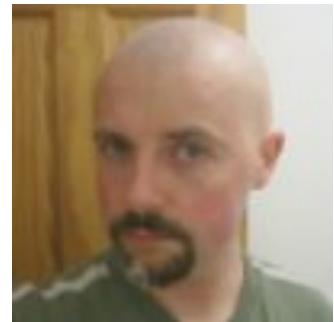
Thanks / Contact details / Questions



Sky of Adopt Animals, <https://twitter.com/carrotcodes>



Kale Charity, <https://www.kale.org.uk/>



(that's me), https://twitter.com/mike_moran