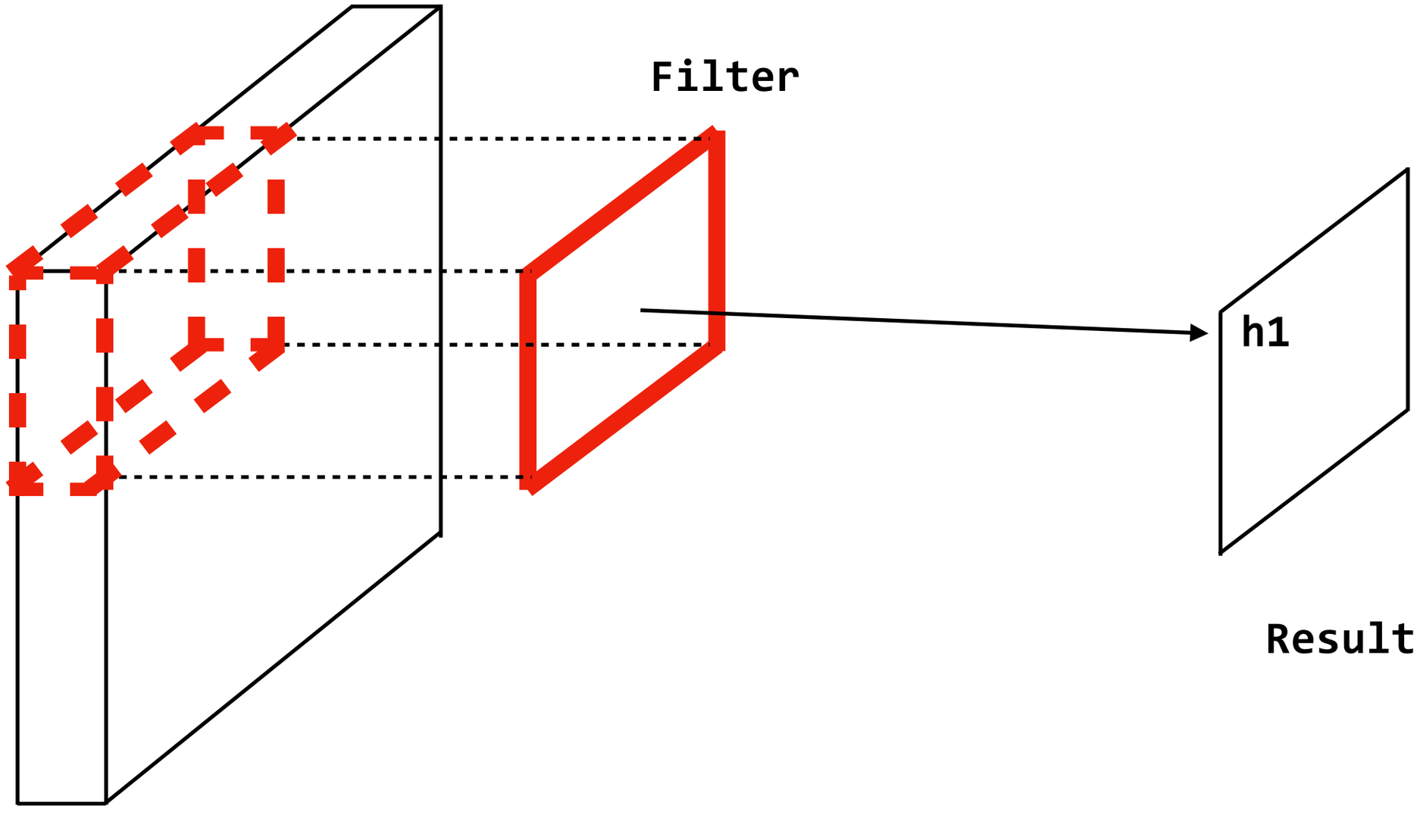
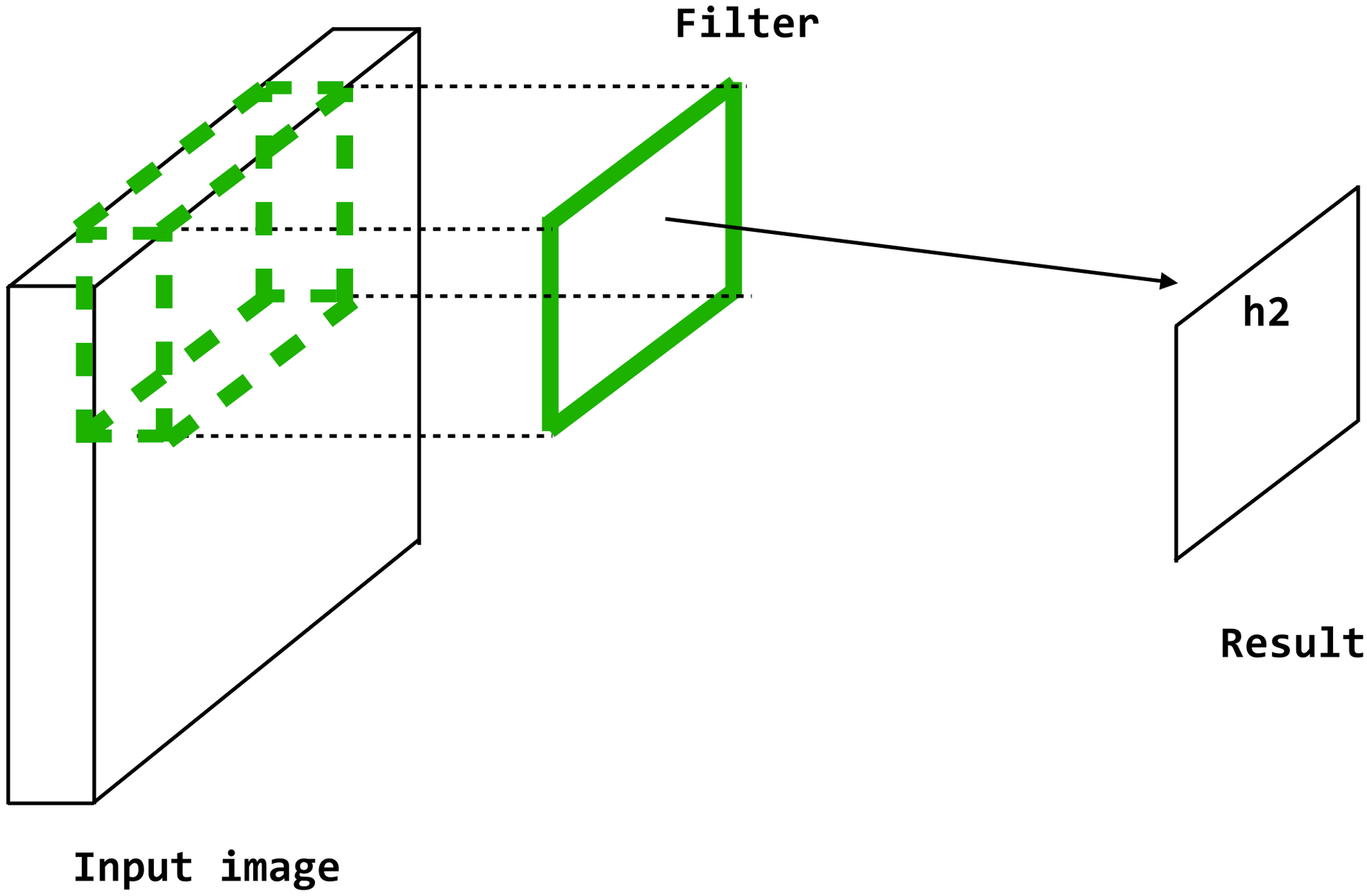
Convolutions III

Recap

Convolution



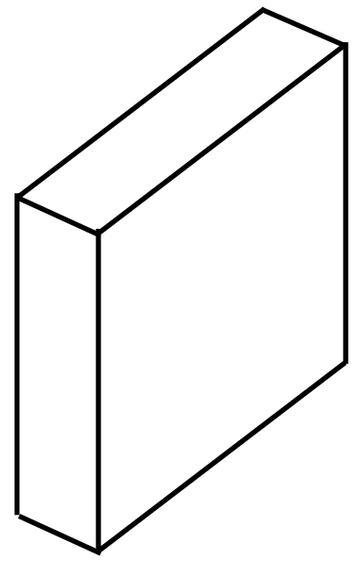
Convolution



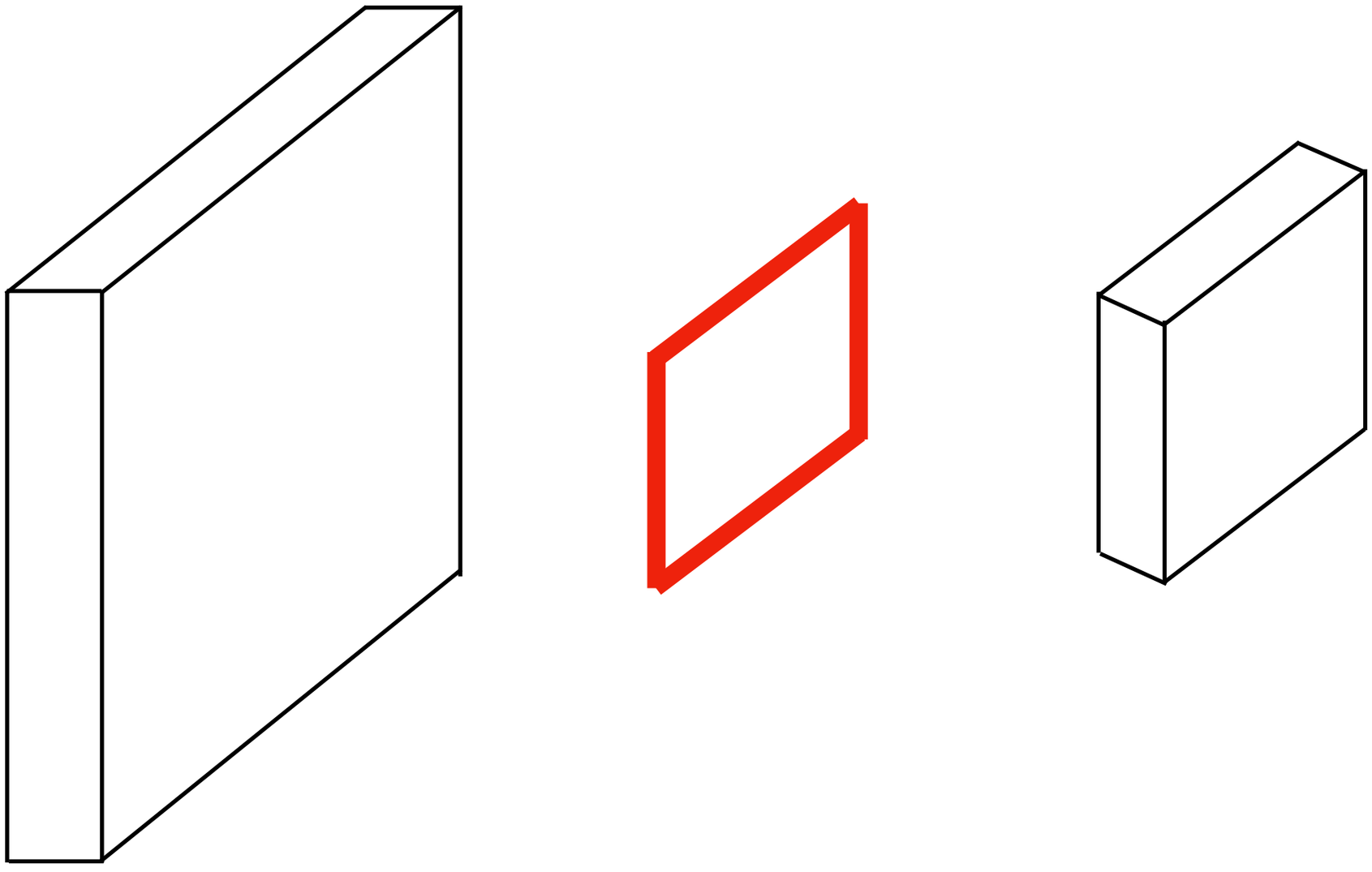
That was 1 filter

With (say) 64 Of These

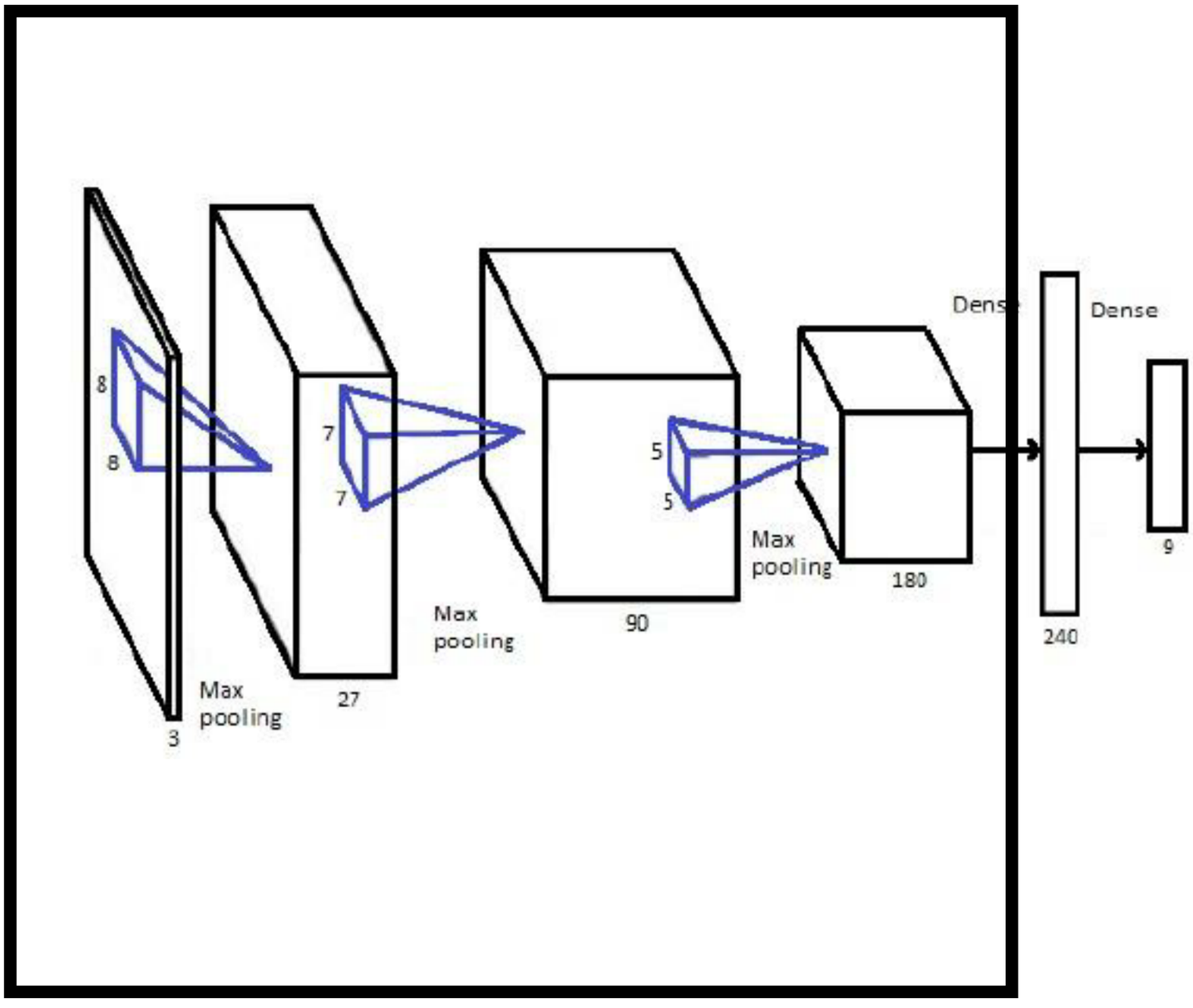
64 Such “Results”



The Transformation



Input **Filter** **Output**



Conv backbone

Case Studies

Creative uses of the conv backbone

Flickr Example

- Input:
  - Image
  - Attributes - user gender, user age, location etc (1 set per image)
- Output:
  - A set of tags for the image
    - (say “nature”, “wildlife”, “flowers” etc)

Strategy

- Define input shape
- Define output shape
- Define loss
- Put it together

Input

- 2 Inputs
- Image (28 x 28)
- Attribute vector (10 dimensional)

Output

- Say 100 possible tags
- Binary vector of size 100
- <0, 0, 1, 0, . . ., 1, 0>
- One binary vector per image

Loss

- We've seen so far:
  - Cross-entropy
    - Predict classes
- Quadratic loss
  - Predict real values

Loss Design

- What is produced by the model after transformations?

- What should it look like?

Classification

- 10-class classification
- Model produces 10-dim vector
- This vector must look like class probabilities

Classification

- We did it by:
    - Softmax
        - Turns into a probability
- Categorical cross entropy
    - Compares two probabilites

Regression

- Model spits our real number $\hat{y}$
- Target is a real number y
- Quadratic loss:

$$(y - \hat{y})^2$$

Our Case

- Model spits out a vector (say)

- Need to compare with target binary vector

One Option

- Quadratic loss:
- Model returns: $< \hat{y_1}, \hat{y_2}, \hat{y_3}, \hat{y_4}, \ldots, \hat{y_{100}} >$
- We want: $< y_1, y_2, y_3, y_4, \ldots, y_{100} >$

$$(y_1 - \hat{y_1}2)^2 + (y_2 - \hat{y_2})^2 + (y_3 - \hat{y_3})^2 + (y_4 - \hat{y_4})^2 + \ldots + (y_{100} - \hat{y_{100}})^2$$

Multilabel Problem

- What if sparse (i.e. lots of zeros) and few ones in the tags?
- This is normal
  - Most images have very few tags
- Model can return just 0 (i.e. no tags)
- And loss will be low

Alternate Idea

- Target is binary vector
- Each entry thus 0 or 1
- So each entry is a tag probability
- We have a vector of probabilities

So

- For each tag run a classification problem
- So 1 cross-entropy loss per tag
- Add these losses up

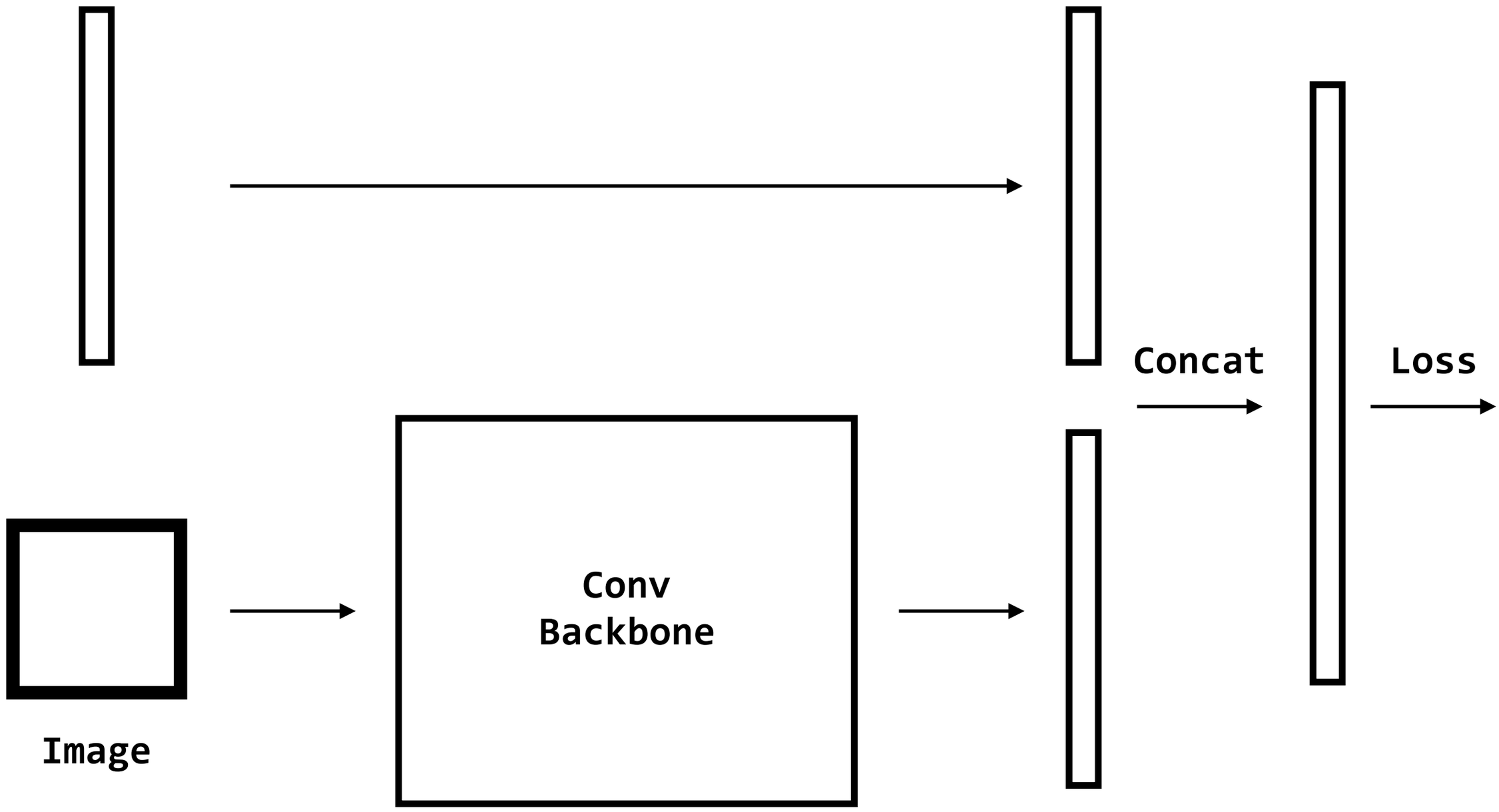
The Model

- Need to combine
- Image
- Attribute vector

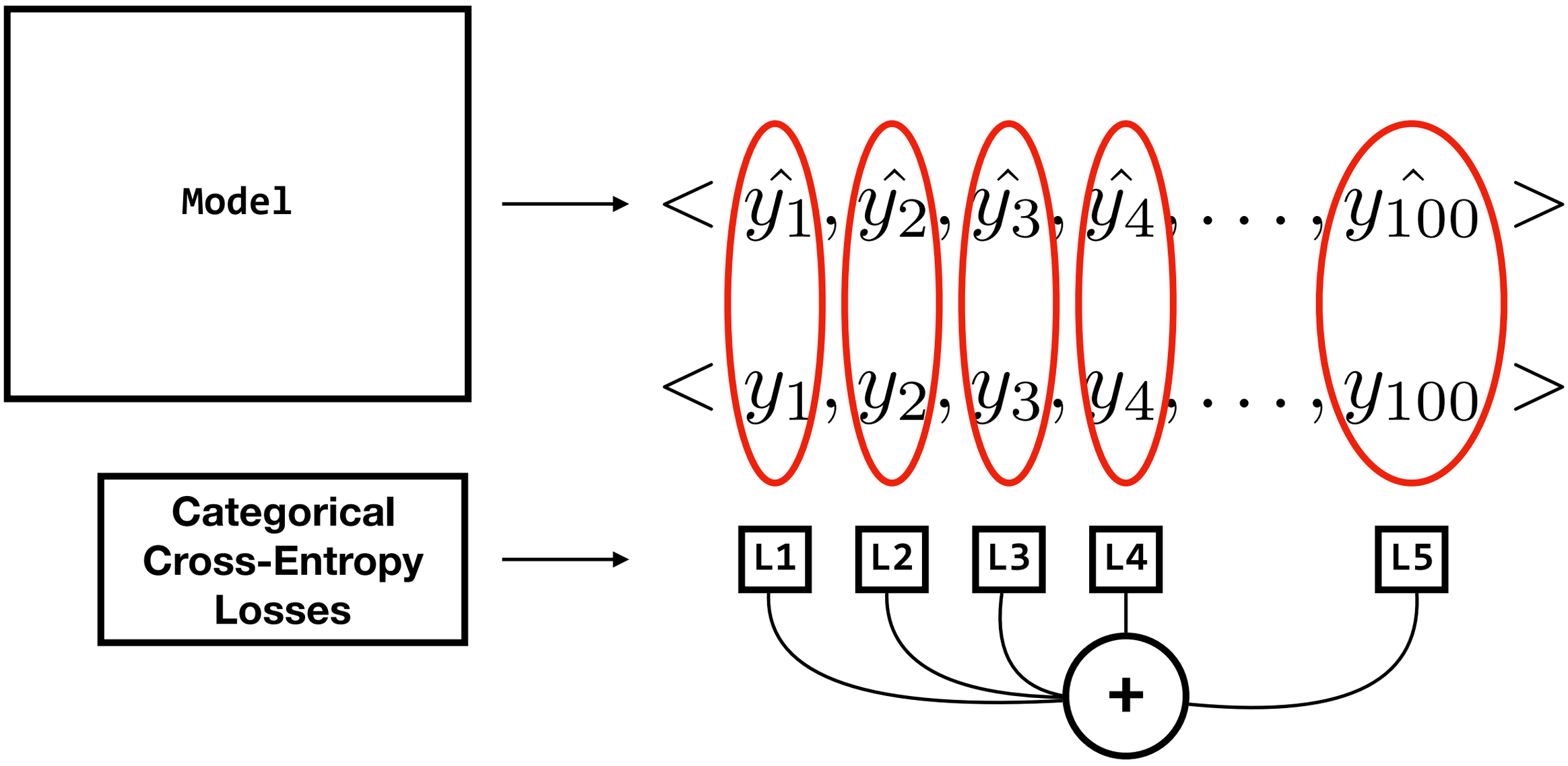
Simple

- Send image through conv backbone
- Flatten the result
    - Vector representation of the image
- Concatenate with attribute vector
- This is sent through MLP

Model Diagram



Loss Diagram



Reference

- https://keras.io/layers/merge/#concatenate
- https://keras.io/losses/#binary_crossentropy
  - One liner for the loss we implemented

Siamese Net Example

- Given:
  - 2 images of faces (close up etc. etc.)
- Need to say if the faces belong to the same person

Parts

- Input:
  - 2 Images
- Output:
  - Yes / No
  - Yes - same person in both faces
  - No - different person in both faces

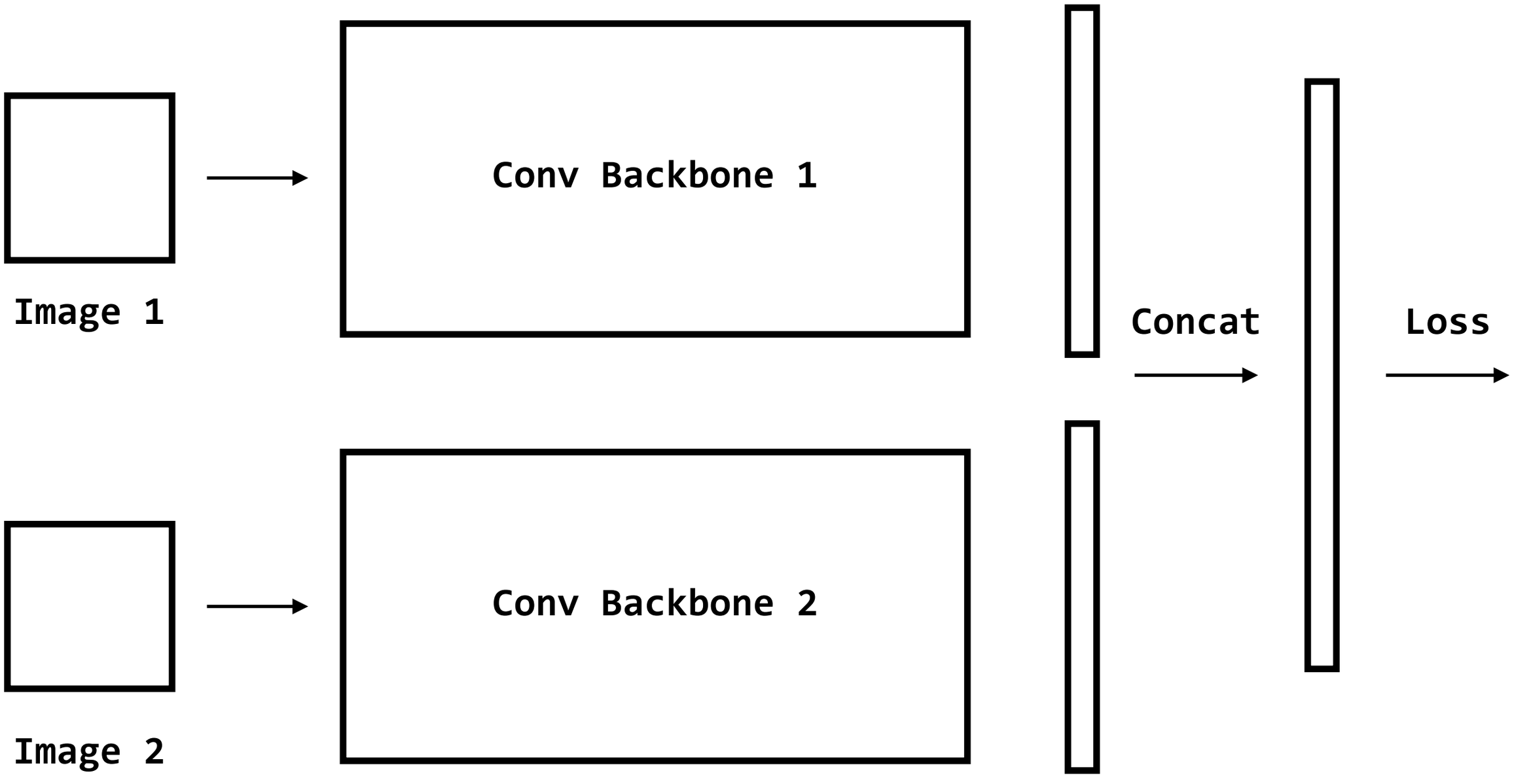
Ideas?

- Design loss
- Design model

One Solution

- 2 Convolution backbones
  - 1 per input image
- Get representations
- Concatenate and use for 2-class classification

Diagram



Loss

- Cross entropy (2 categories)

Possible Problems?

Symmetry

- Can’t guarantee symmetry

- It is important since results **must not** be different for swapped inputs

Either

- Show it both pairs as input
  - Less likely to work (why)
- **Share weights**

Weight Sharing

- https://keras.io/getting-started/functional-api-guide/#shared-layers

- https://www.tensorflow.org/programmers_guide/variables

Metric Learning Setup

- Previous problem loss value
- 0 <- “Not Same Person”
- 1 <- “Same Person”

i.e.

- Model emits a real number between 0 and 1
- Closer to 1
  - Higher confidence “Same Person”
- Closer to middle
  - Not so sure
- Closer to 0
  - Higher confidence “Not Same Person”

You've Computed

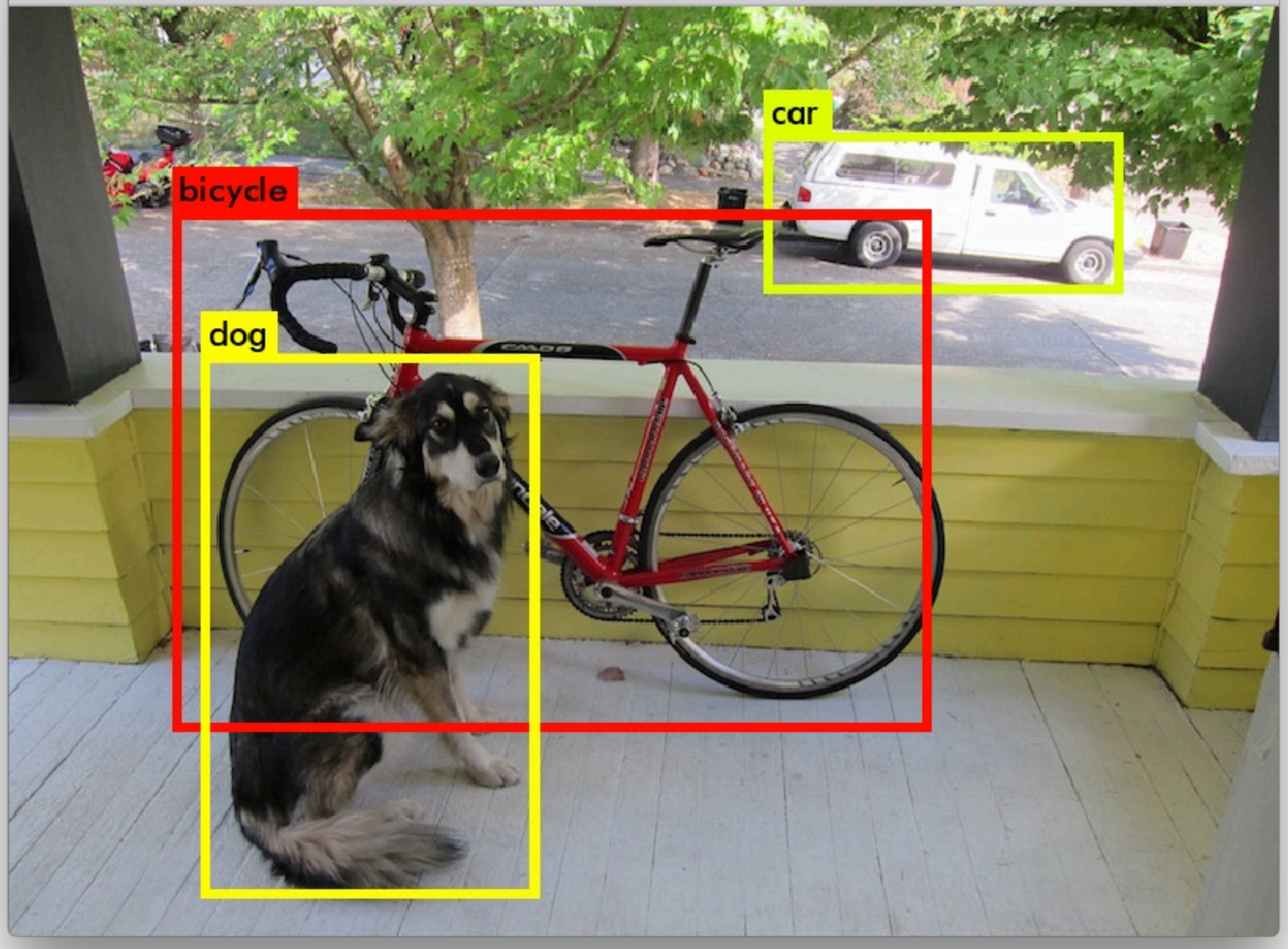
- Distance
- Similarity
  - (Same Thing Really)
- Given 2 faces
  - how close (far apart) are they

Object Detection

- Input:
  - Image
- Question:
  - Where is the object in this image?
- Given label:
  - A bounding box

predictions.jpg

Search



Ideas?

Loss

- 2-category classification per pixel
- (dog or not-dog)
- (car or not-car)
- Etc

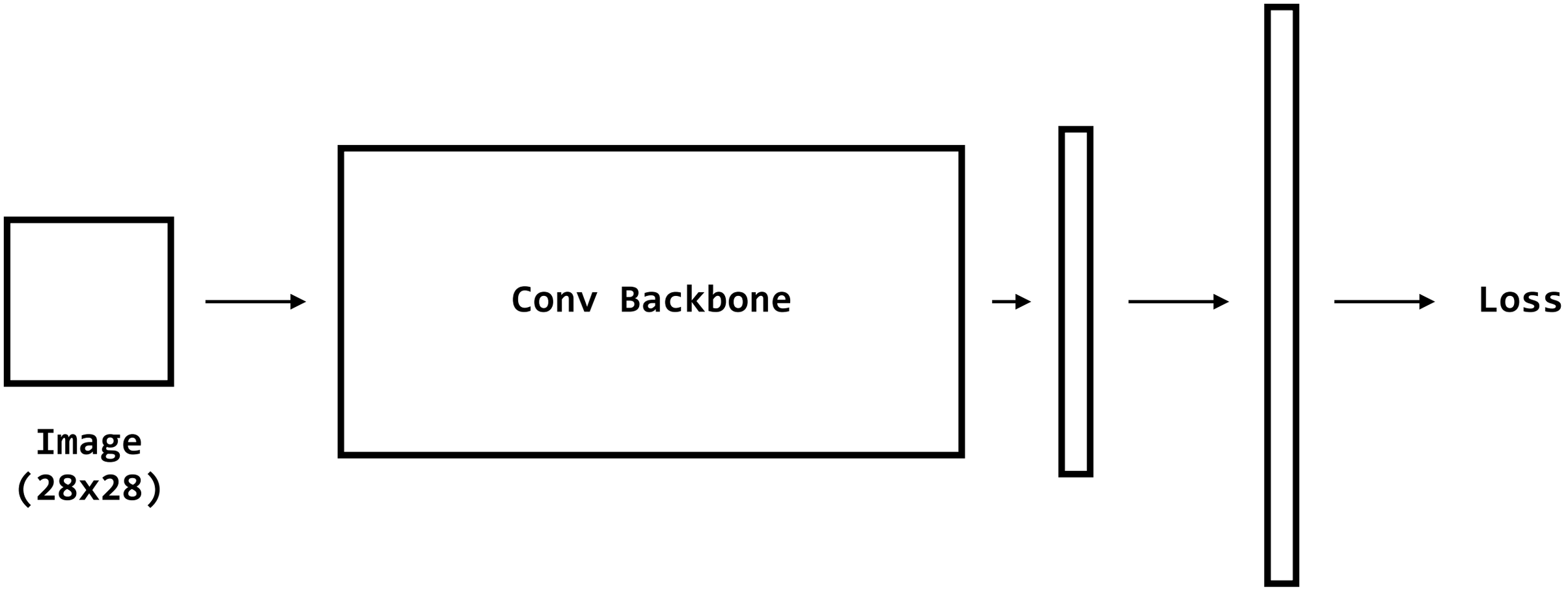
Input

- Need to be a little creative
- Take candidate regions
- Do predictions on candidate regions

Loss

- Looks like the loss we saw in the Flickr example

Loss



Convolution Wins

- Big wins recently in
  - Machine translation
  - Audio synthesis
  - Video
  - etc.

Debugging

Recap

- Fully Connected Layer

- ReLUs fold the space

What Are The Convolutions Doing?

- Typical approaches are empirical
- Plot filter responses
- Visualize embeddings (representations)
- Visualize weights

Visualizing Responses

Deep Visualization Toolbox

yosinski.com/deepvis

#deepvis

Jason Yosinski

Jeff Clune

Anh Nguyen

Thomas Fuchs

Hod Lipson

Cornell University

University of Wyoming

NASA Jet Propulsion Laboratory
California Institute of Technology

https://www.youtube.com/watch?v=AgkfIQ4IGaM

MNIST TSNE

- Demo

TSNE

- Dimension Reduction
- Good for visualizing what the network has done

TSNE

- Reduced dimension preserves pairwise distances

Deepdream

- https://github.com/google/deepdream/blob/master/dream.ipynb

Intuition

- Pick some layer you want to “enhance”
- Backprop all the way to the image
- Result = trippy images

Deepdream

- Enhance responses

- how?

Acknowledgements

- Ruslan Salakhudtinov - http://www.cs.cmu.edu/~rsalakhu/10707/Lectures/Lecture_Conv1.pdf
- Some figures from CS231N @ Stanford
- Good vibes from the Keras and TF teams
- Vincent Dumoulin - https://github.com/vdumoulin/conv_arithmetic