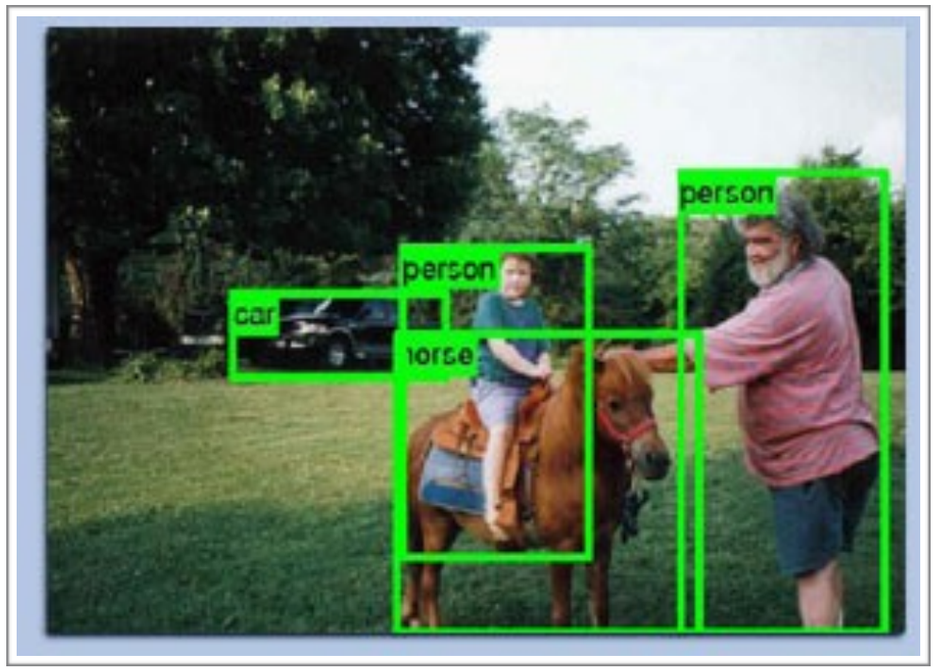
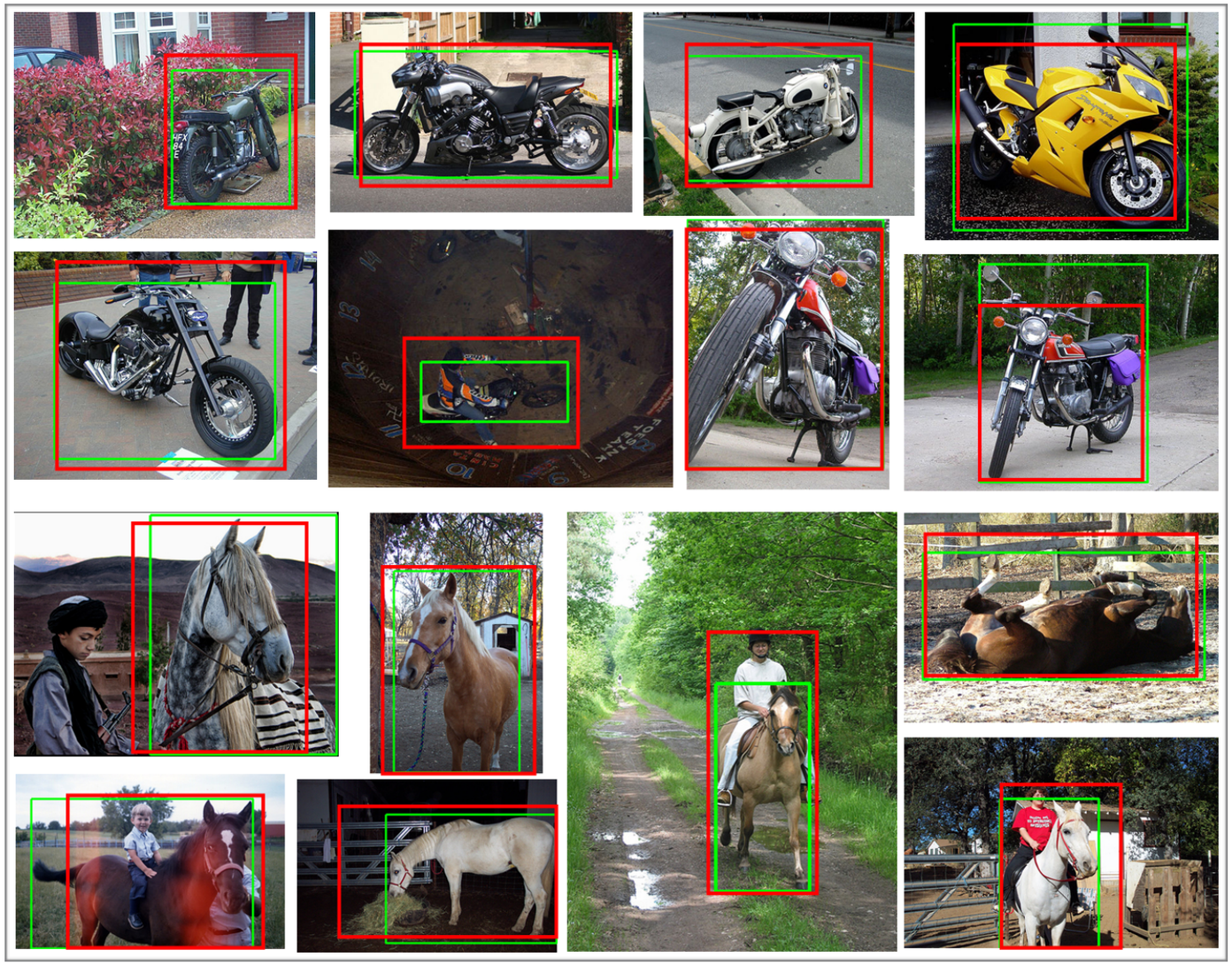
Deep Neural Networks for Object Detection

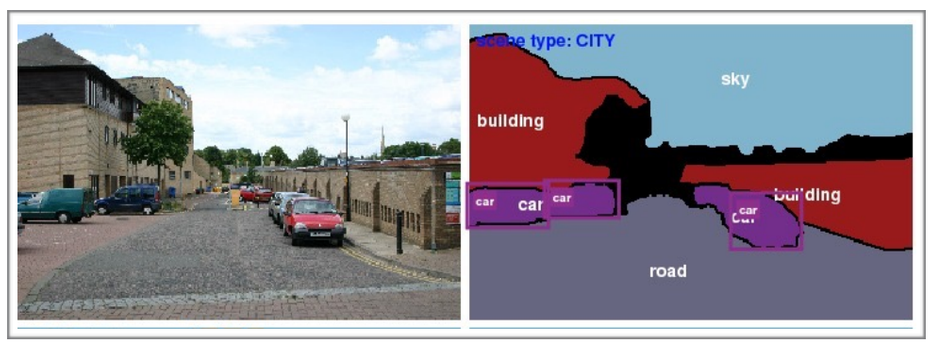
Paper by C. Szegedy, A. Toshev, D. Erhan [2013]
Presentation by Joaquín Ruales

The Problem: Object Detection

- Identifying and locating objects in an image

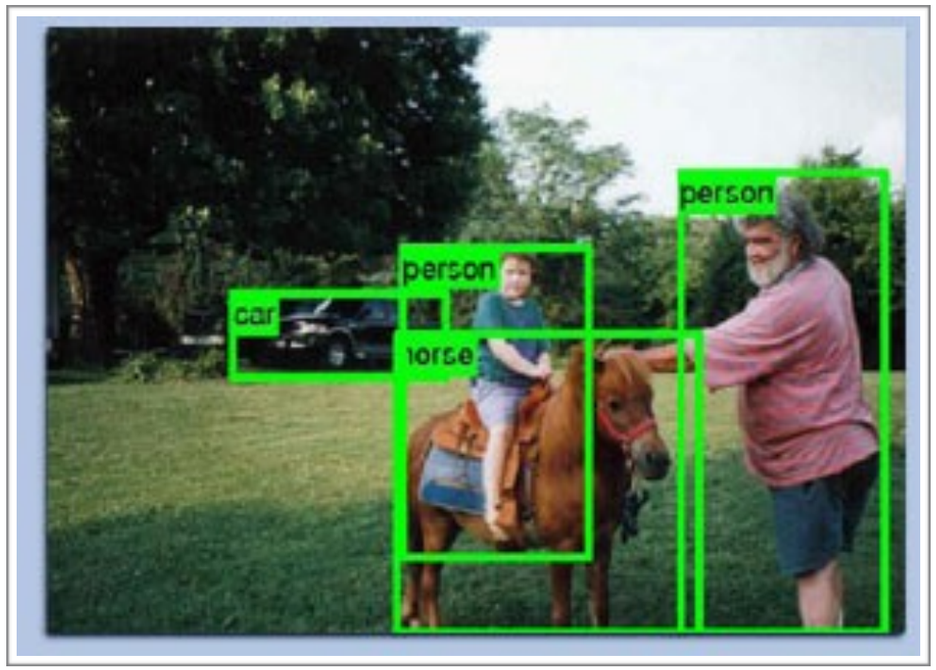






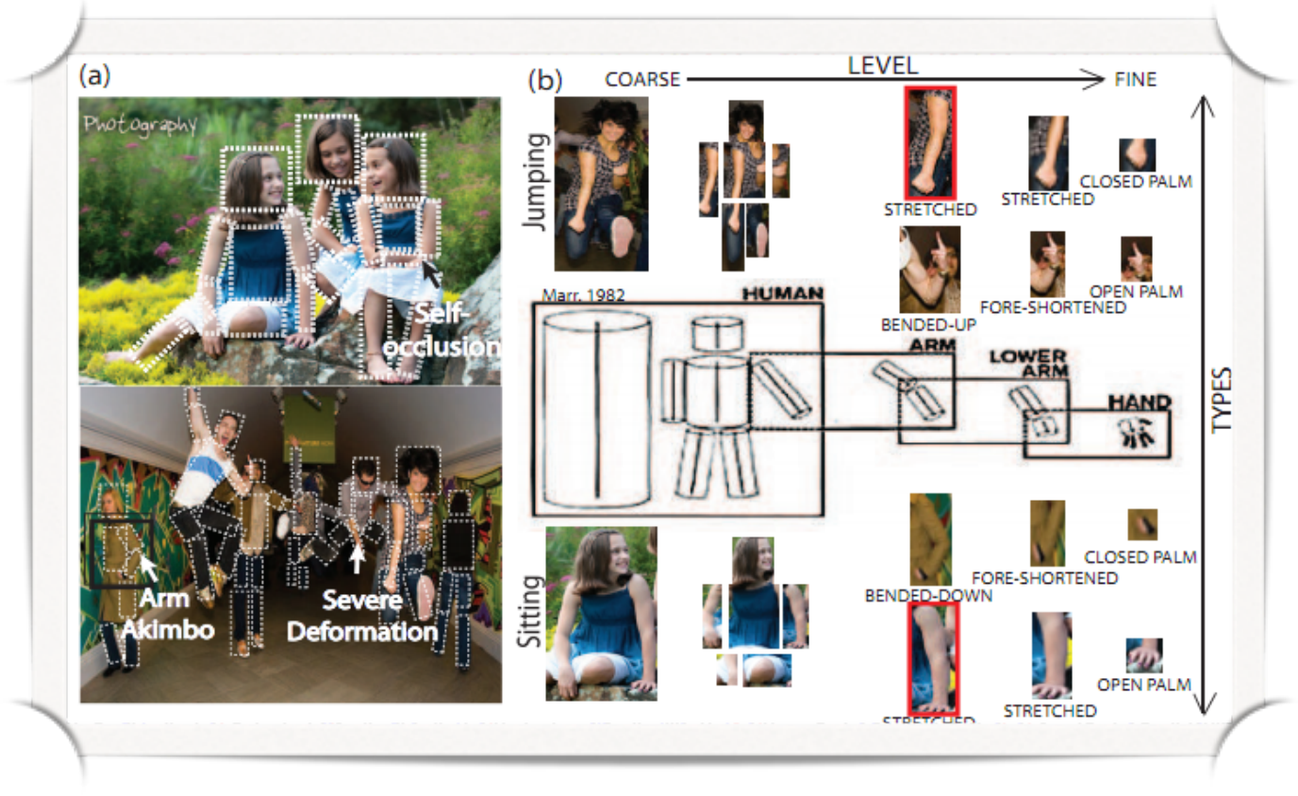
The Problem: Object Detection

- Identifying and locating objects in an image



Previous Work in Object Detection

- Discriminative Part-based models:
  - Identifying parts of an object and their relation to identify the whole
  - Exploits domain knowledge. Uses HOG descriptors
- Some NN approaches, but used as local classifiers, or incapable of distinguishing many instances of same class of object



Why DNN for Object Detection?

- Success of DNNs for related problem: image classification
  - A. Krizhevsky, I. Sutskever, G. Hinton. (2012). ImageNet Classification with Deep Convolutional Neural Networks

- Can take advantage of the small shift-invariance in DNN image classification

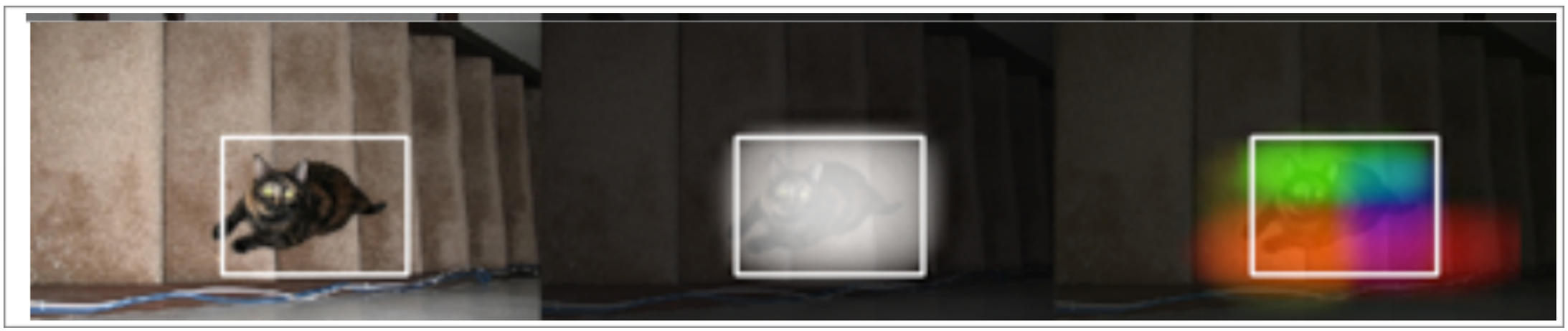
- Simpler models, easily extensible to new classes of objects

Deep Neural Networks for Object Detection

- This paper uses DNNs to classify and precisely locate objects of 20 classes (plane, bicycle, bird, boat, etc.)

- Requires several applications of the DNNs

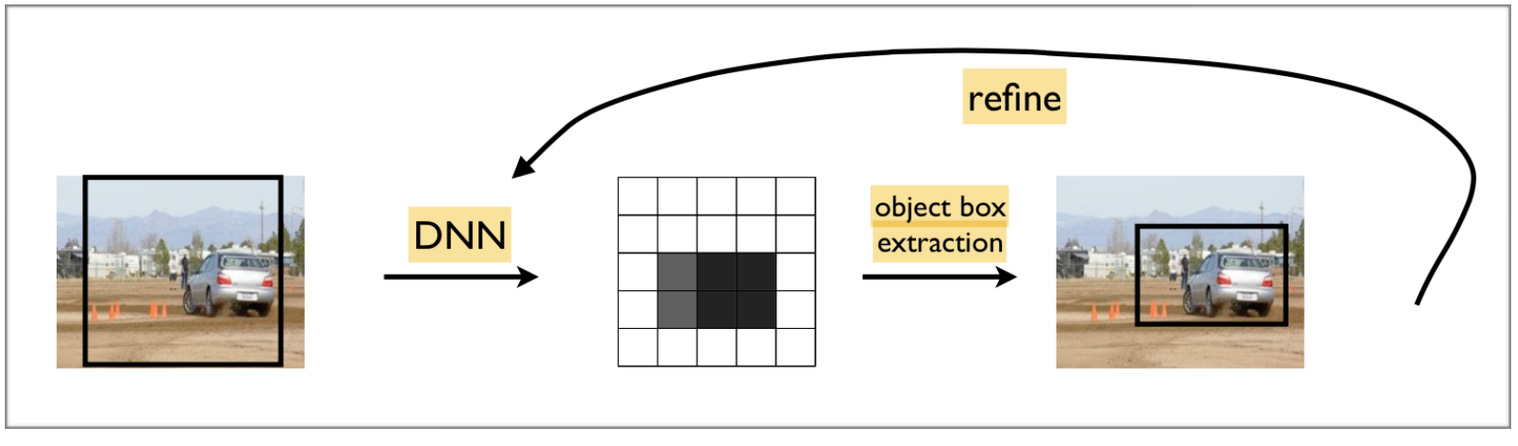
- Obtains state-of-the-art performance on the Pascal VOC dataset



Detection

Detection

- For each object category X∈{plane, bicycle, bird, boat, etc.}
  - Input: Image.
  - Step 1: Generate binary masks using DNN specific to X
  - Step 2: Get bounding boxes from masks
  - Step 3: Refine bounding boxes
  - Output: Bounding boxes and confidence scores for all objects of type X in the image

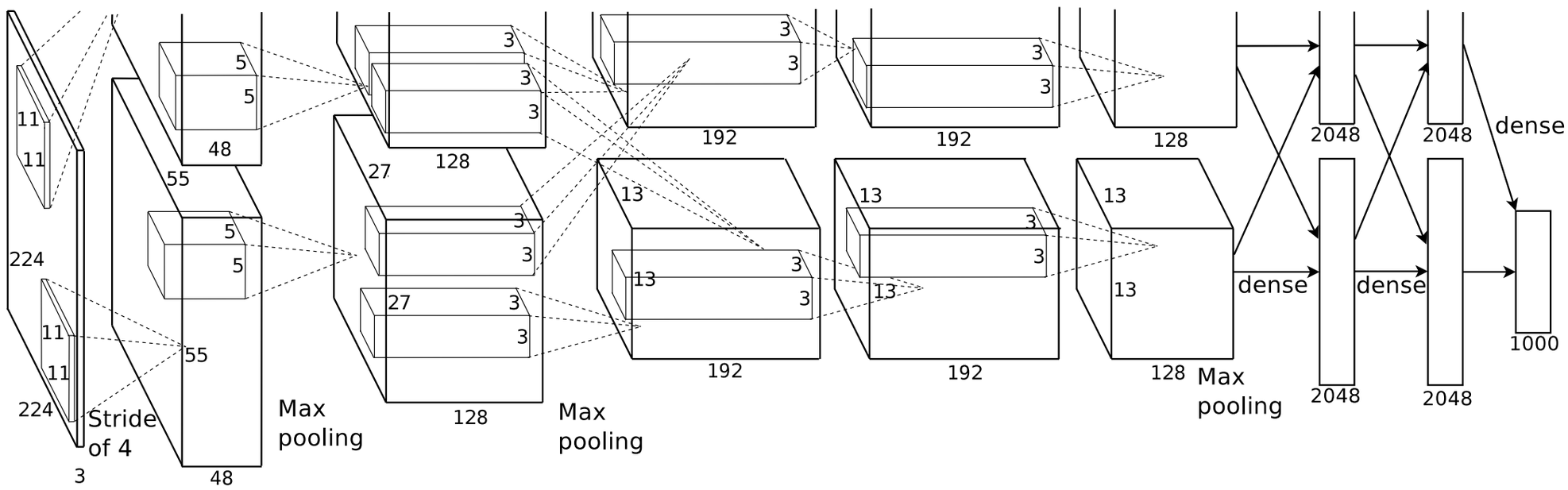


Detection Step #1: Generate Binary Masks using DNN

- Same DNN structure as [A. Krizhevsky, I. Sutskever, G. Hinton. (2012). ImageNet Classification with Deep Convolutional Neural Networks]

  - 5 convolutional layers (3 with max pooling), 2 connected layers, ReLu nonlinearities

- Except: replace softmax classification layer (last layer) with a regression layer that produces a binary mask

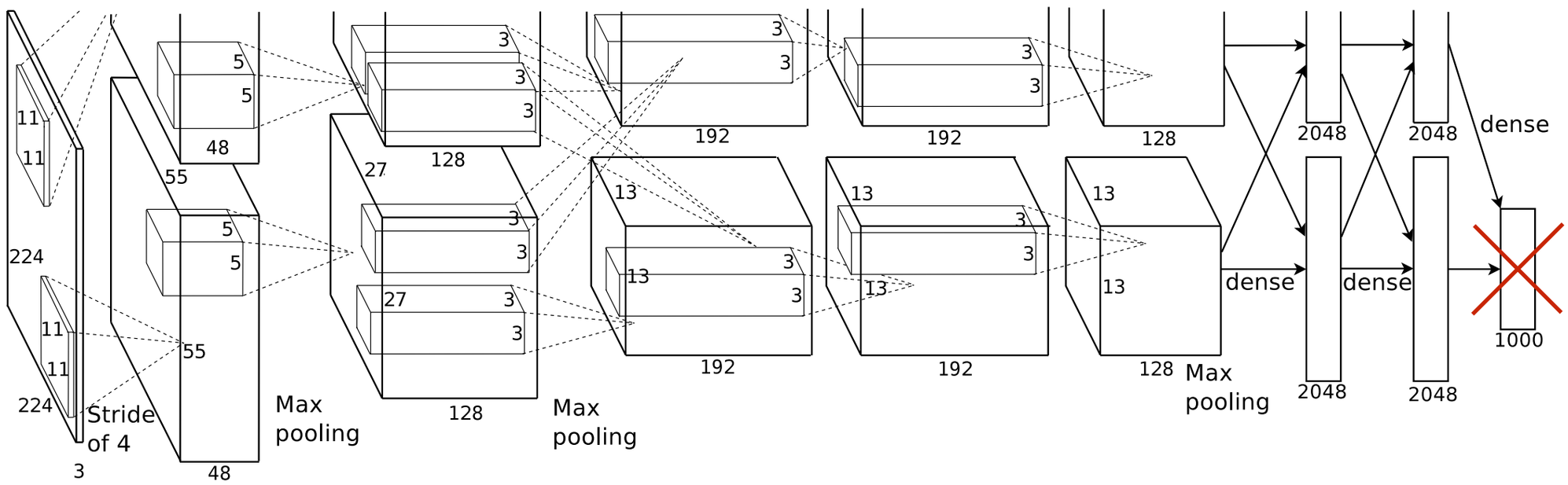


Detection Step #1: Generate Binary Masks using DNN

- Same DNN structure as [A. Krizhevsky, I. Sutskever, G. Hinton. (2012). ImageNet Classification with Deep Convolutional Neural Networks]

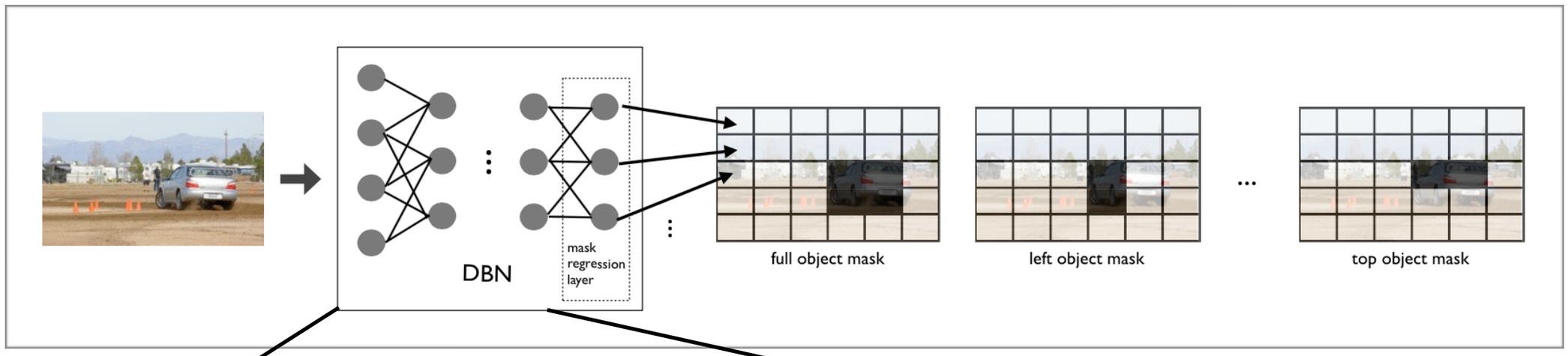
  - 5 convolutional layers (3 with max pooling), 2 connected layers, ReLu nonlinearities

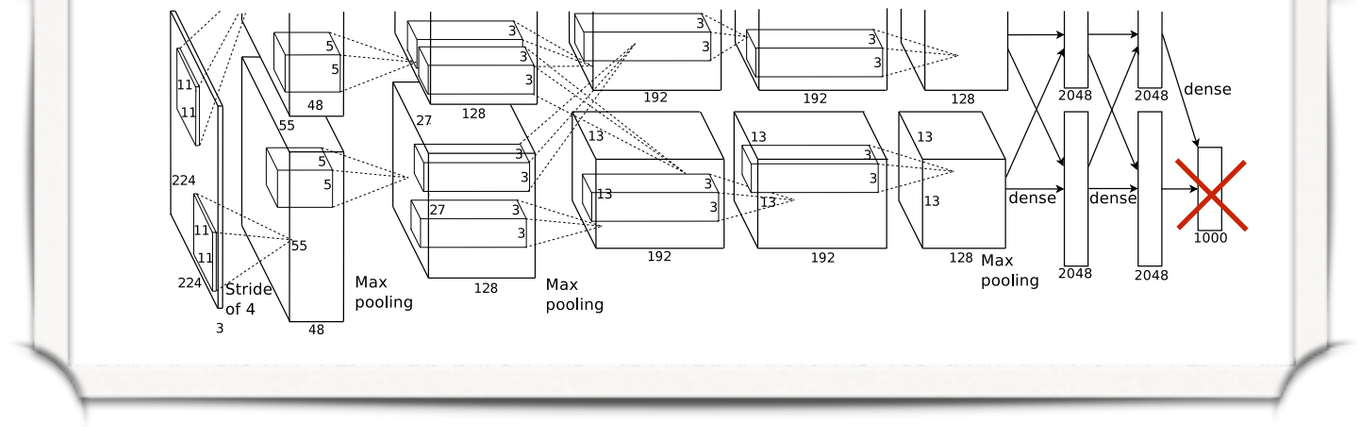
- Except: replace softmax classification layer (last layer) with a regression layer that produces a binary mask



Detection Step #1: Generate Binary Masks using DNN

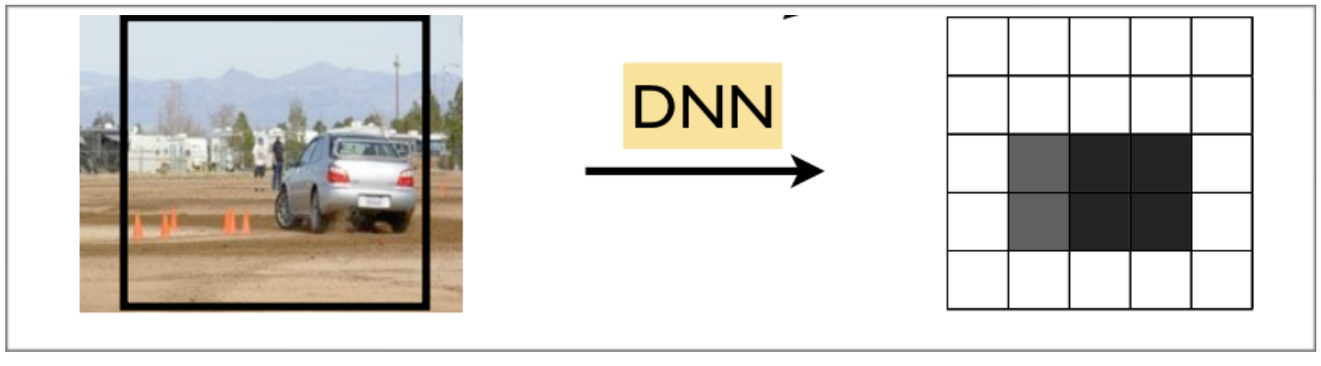
- Actually, 5 DNNs trained per category
  - Full object mask, left half, bottom half, right half, top half
  - 5 masks are then merged to get the final mask
- DNN inputs are 225x225 pixels. Output masks are 24x24 pixels





Detection Step #1: Generate Binary Masks using DNN

- Compute these masks for many sub windows of the original image, at several scales
- (Different than sliding window approach since usually need <40 windows per image)



Detection Step #2: Get Bounding Boxes

- Find the bounding boxes with best scores for the set of 24x24px output masks

$$S(bb) = \sum_{h \in halves} (S(bb(h), m^h) - S(bb(\bar{h}), m^h))$$

Percentage of bounding box that overlaps with region h

The complement of region h

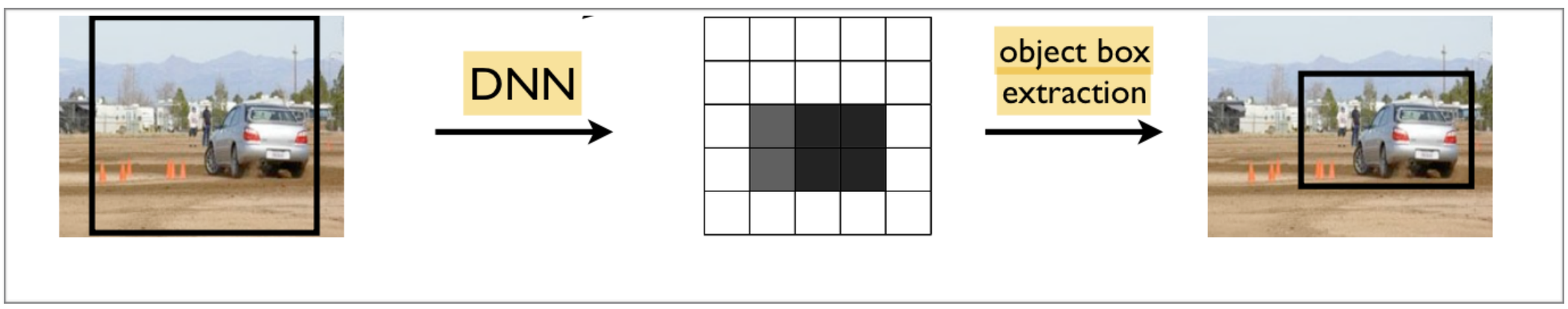
score

bounding box

mask

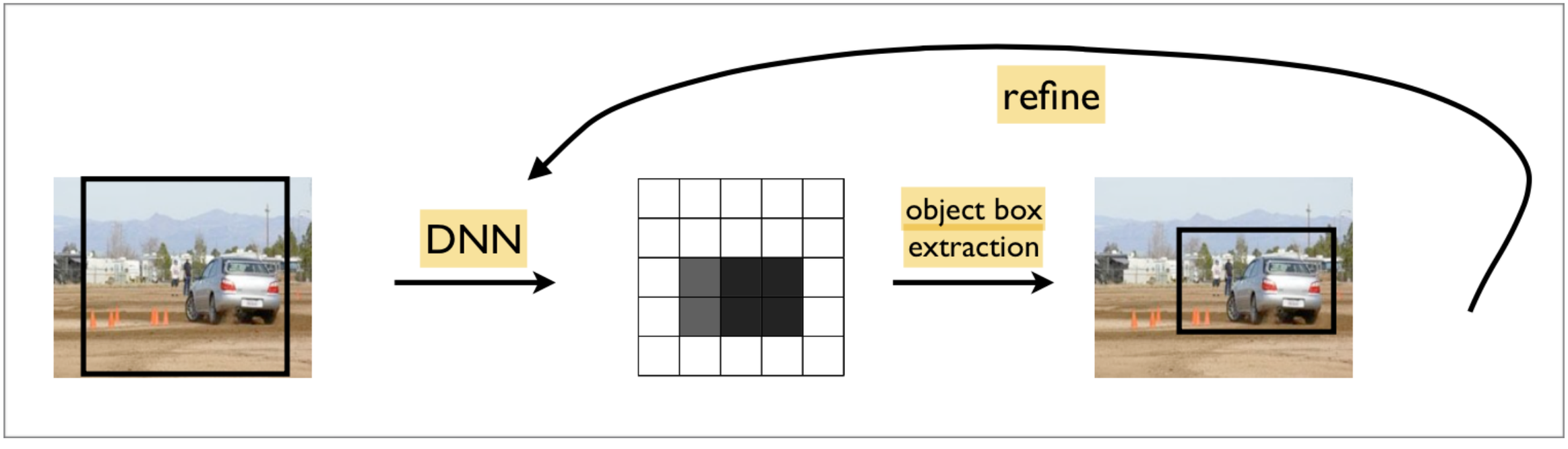
  - (exhaustive search. Sped up using integral images)

- Map bounding box back to image space (note resolution loss)



Detection Step #3: Refine bounding boxes

- Crop original image to each bounding box
- Repeat step #1 (Generate binary masks with DNN) on the cropped image
- Repeat step #2 (Get bounding boxes) for the generated binary masks
- Discard the bounding boxes that received a low score
- Run the detected object through a classifier DNN and discard the corresponding bounding box if misclassified
- Result: Final, fine-grained bounding boxes around the object with scores



Precision and Recall
Before and After Refinement

- Based on results on VOC2007 test data

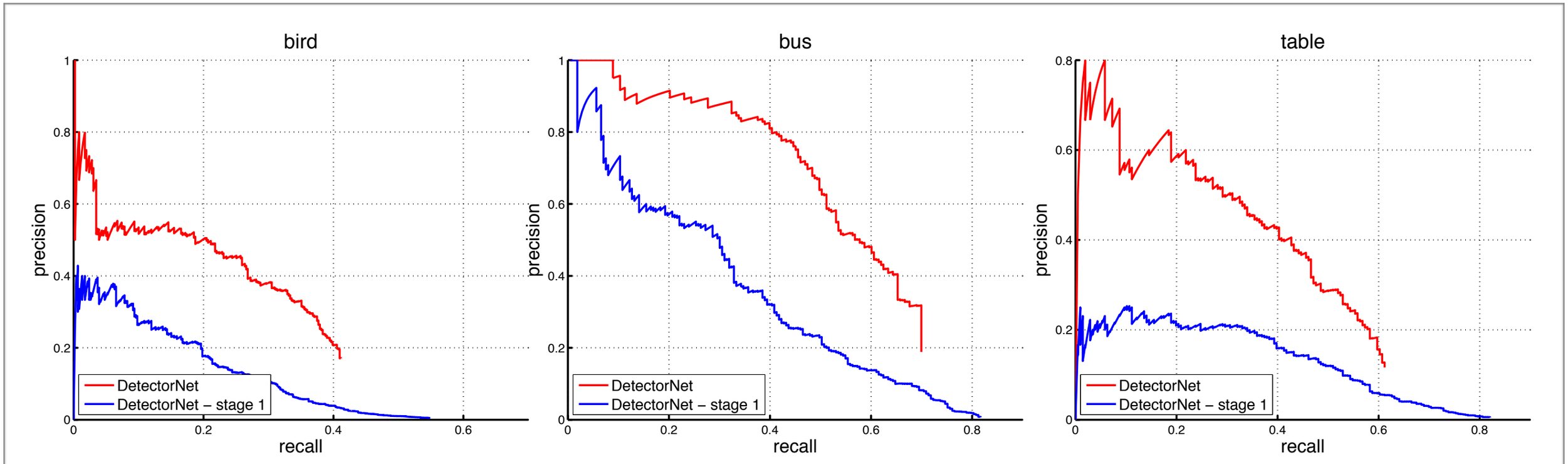


Figure 4: Precision recall curves of DetectorNet after the first stage and after the refinement.

Training

Training

- Needs a lot of training data: Objects of different sizes at almost every location
- Use VOC2012 training and validation set (~11K images) for training
- Remember: we need to train 2 types of DNNs:
  - 1) Mask generator DNN (maps images to binary masks)
  - 2) Classifier DNN (used for final pruning of detections)

1) Mask Generator Training

- Krizhevsky et al. ImageNet CNN with last layer replaced by regression layer

- Minimize L_2 error for predicting a ground truth mask *m* for an image *x*

Regularizer in $\mathbf{R}^+$.
When small, it penalizes all-zero masks

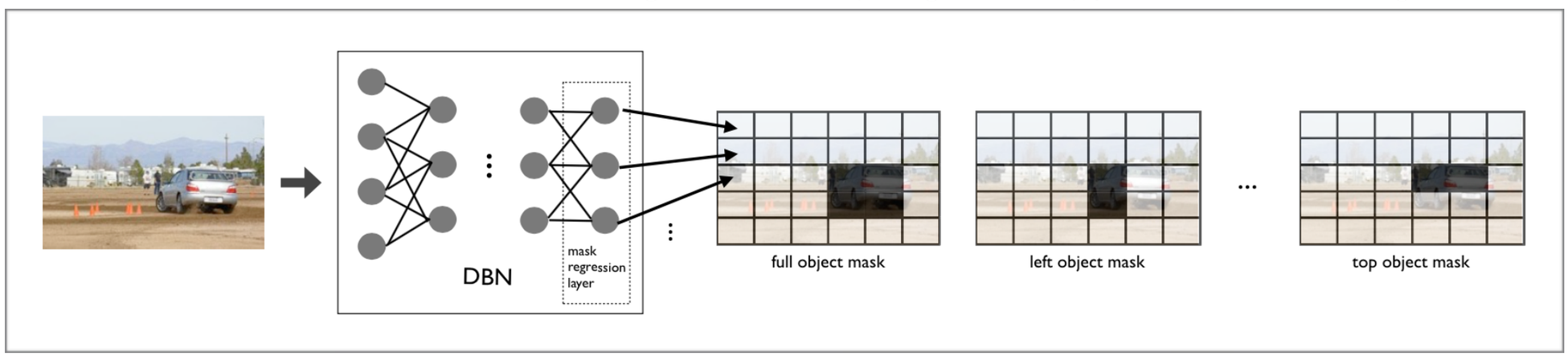
Ground truth mask

$$\min_{\Theta} \sum_{(x,m)\in D} ||(Diag(m) + \lambda I)^{1/2}(DNN(x;\Theta) - m)||_2^2,$$

Vector of mask generator DNN parameters

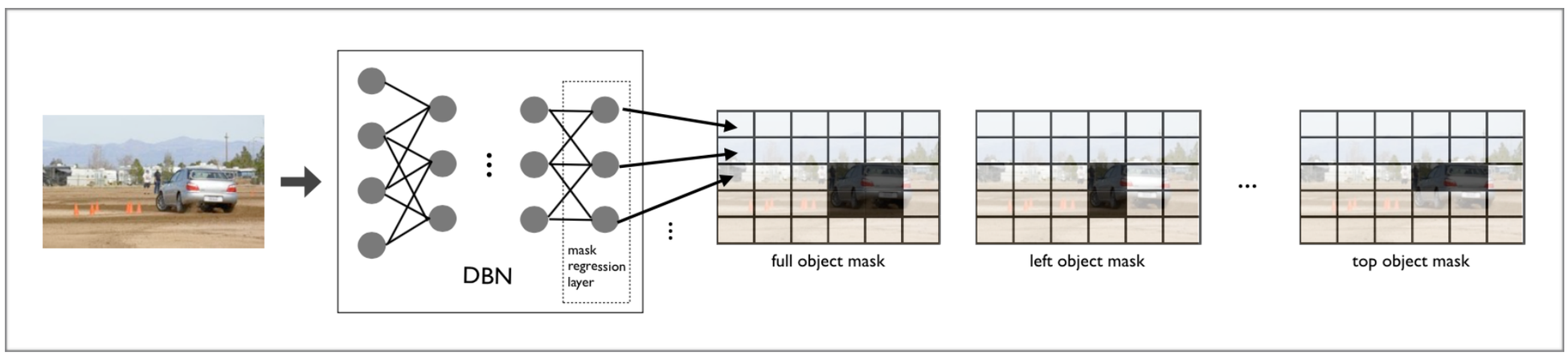
Set of ground truth (image, mask) pairs

Mask generator output



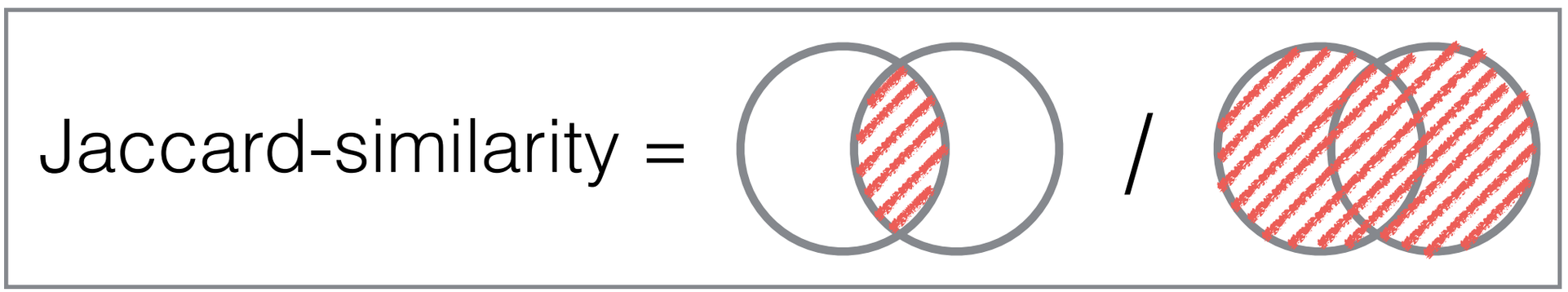
1) Mask Generator Training

- Several thousand samples from each image (10M total)
- 60% negative examples
  - outside of bounding box of any object of interest
- 40% positive examples
  - each covers >80% of area of some ground truth bounding box of interest
- Crops sampled so that cropWidth~Uniform(minScale, imageWidth)



2) Classifier Training

- Krizhevsky et al. ImageNet CNN
- Several thousand samples per image (10M total)
- 60% negative examples
    - each has <0.2 Jaccard-similarity with any ground truth box
    - acts as a 21st class in the classifier
- 40% positive examples
    - each has >0.6 Jaccard-similarity with any ground truth box
    - labeled according to category of most similar bounding box



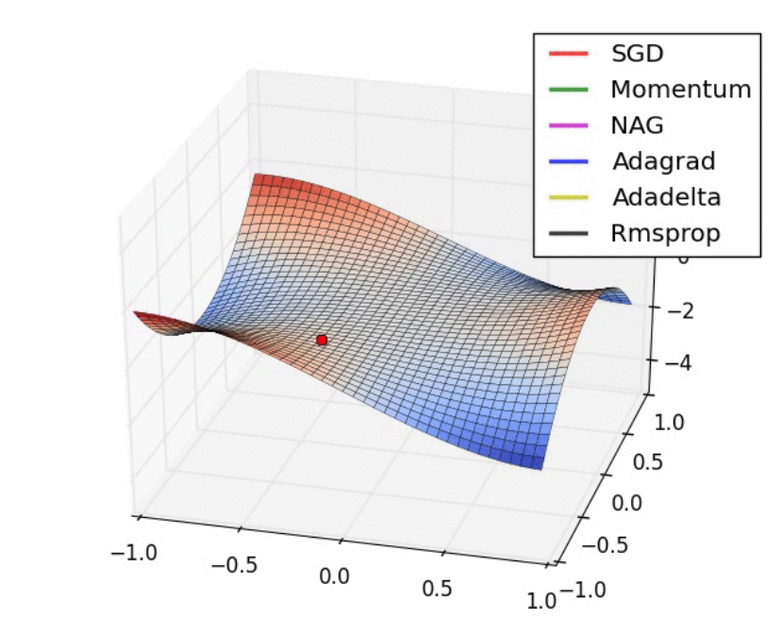
Final Notes on Training

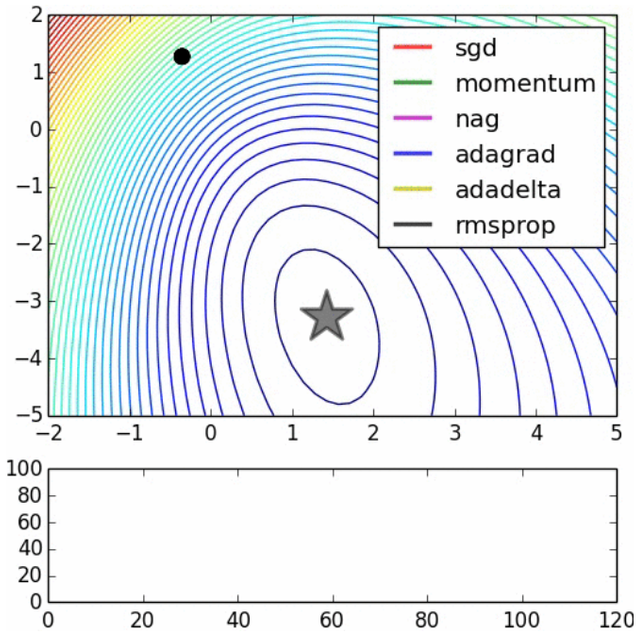
- CNNs, max pooling, dropout
- AdaGrad training

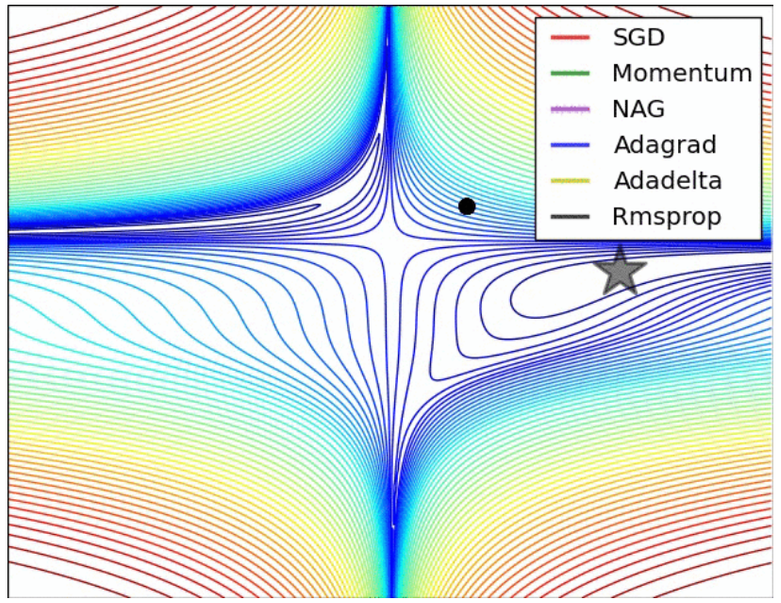
```
# Assume the gradient dx and parameter vector x
cache += dx**2
x += - learning_rate * dx / np.sqrt(cache + 1e-8)
```

  - A type of adaptive learning rate for SGD

- Training for localization harder than for classification, so they reuse the classification DNN weights for the localization DNN







Results

Results

- Algorithm obtained state-of-the-art for VOC2007 (Pascal Visual Object Challenge 2007) dataset
  - Best detection for 8 of the 20 categories
  - Best detection for 5 out of 7 animal categories (bird, cat, cow, dog, sheep)
  - 5-6sec per image per class on a 12-core machine
- More training data than others in this table. Unfair comparison?

class	aero	bicycle	bird	boat	bottle	bus	car	cat	chair	cow
DetectorNet[1]	.292	.352	**.194**	.167	.037	**.532**	.502	**.272**	.102	**.348**
Sliding windows[1]	.213	.190	.068	.120	.058	.294	.237	.101	.059	.131
3-layer model [19]	.294	.558	.094	.143	**.286**	.440	.513	.213	.200	.193
Felz. et al. [9]	**.328**	.568	.025	**.168**	.285	.397	.516	.213	.179	.185
Girshick et al. [11]	.324	**.577**	.107	.157	.253	.513	**.542**	.179	**.210**	.240
class	table	dog	horse	m-bike	person	plant	sheep	sofa	train	tv
DetectorNet[1]	**.302**	**.282**	.466	.417	.262	.103	**.328**	.268	.398	**.470**
Sliding windows[1]	.110	.134	.220	.243	.173	.070	.118	.166	.240	.119
3-layer model [19]	.252	.125	.504	.384	.366	**.151**	.197	.251	.368	.393
Felz. et al. [9]	.259	.088	.492	.412	.368	.146	.162	.244	.392	.391
Girshick et al. [11]	.257	.116	**.556**	**.475**	**.435**	.145	.226	**.342**	**.442**	.413

Table 1: Average precision on Pascal VOC2007 test set.

Thank You