COMP 540 Recitation

Week 12: Finessing the Term Project

TAs: Gunny, Issac, Raymond, Tian

Tasks

- Recommended Timeline
  - Project, Homework 6, Final Exam
- Baseline Recommendations Walk-Thru
  - Pipeline
  - Pre-processing: Augmentation
  - Modelling
  - Ensembling
  - Report & Presentation
- Bonus Recommendations Walk-Thru
  - Bonus Models
  - Bonus Post-processing
- Workflow Recommendation

Timeline

- **ConvNet** -> less than 1 hour’s Training Time
- **U-Net** -> less than 1 hour’s Training Time
- **Masked Regional ConvNet** -> 24 hours Training Time (Purported)
- Pac-Man Reinforcement Learning (Homework 6) -> 1 Afternoon’s Training Time
- Apr 18 -> Project Presentation
- Apr 20 -> Homework 6 Due Date
- Apr 30 -> Final Exam
- Recommendation:
- **Finish your preprocessing-modelling-ensembling work next week**

Example Pipeline

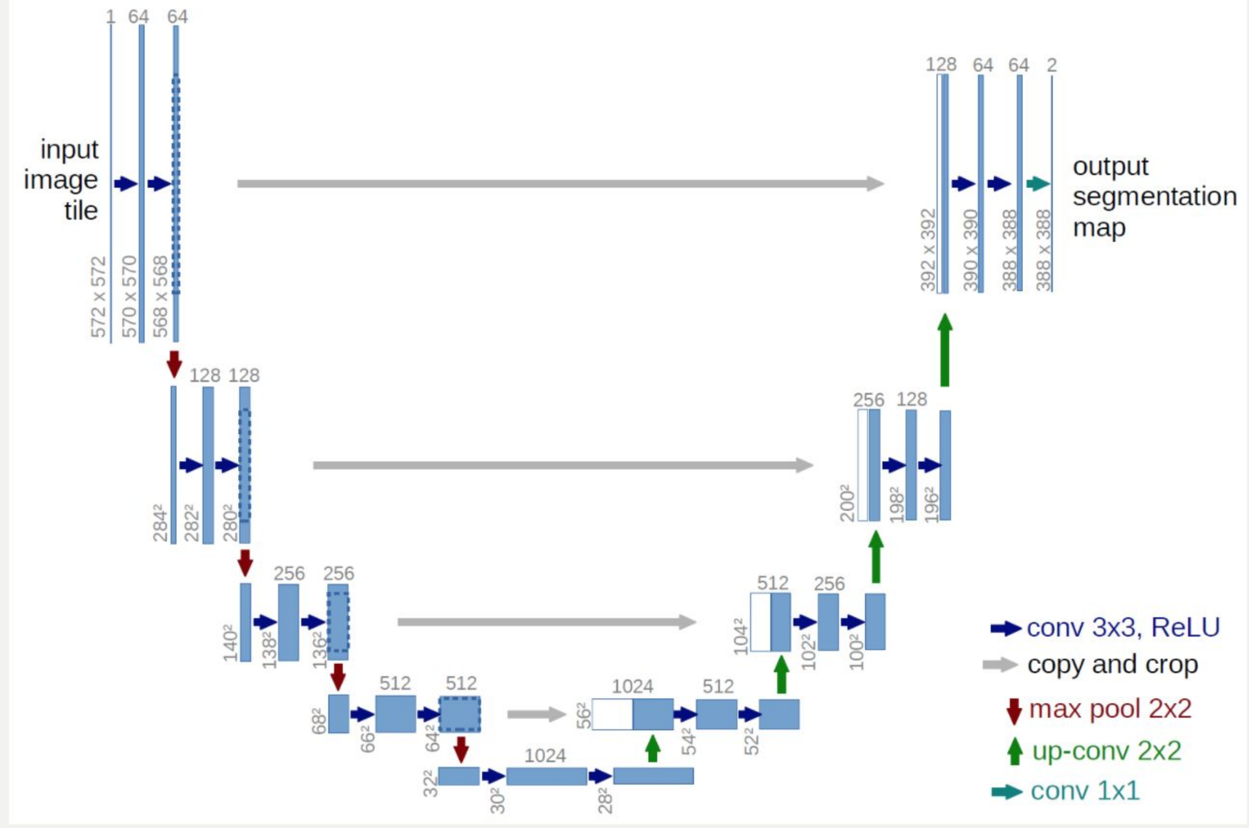
1. Set up k fold validation
2. Set up data augmentation
3. Train your model
4. Perform test time augmentation
5. Feed results into meta learner
6. Submit!

Baseline: Data Augmentation

- Not all training samples look the same
- You have to normalize them
- Numerically
- Stylistically
- Elastic Transform (affine transformation that involves “local distortion” of pixels e.g. making MNIST look like shaky handwriting)
- Averaging predictions (done during training and test time)

Baseline: Models

U-Net



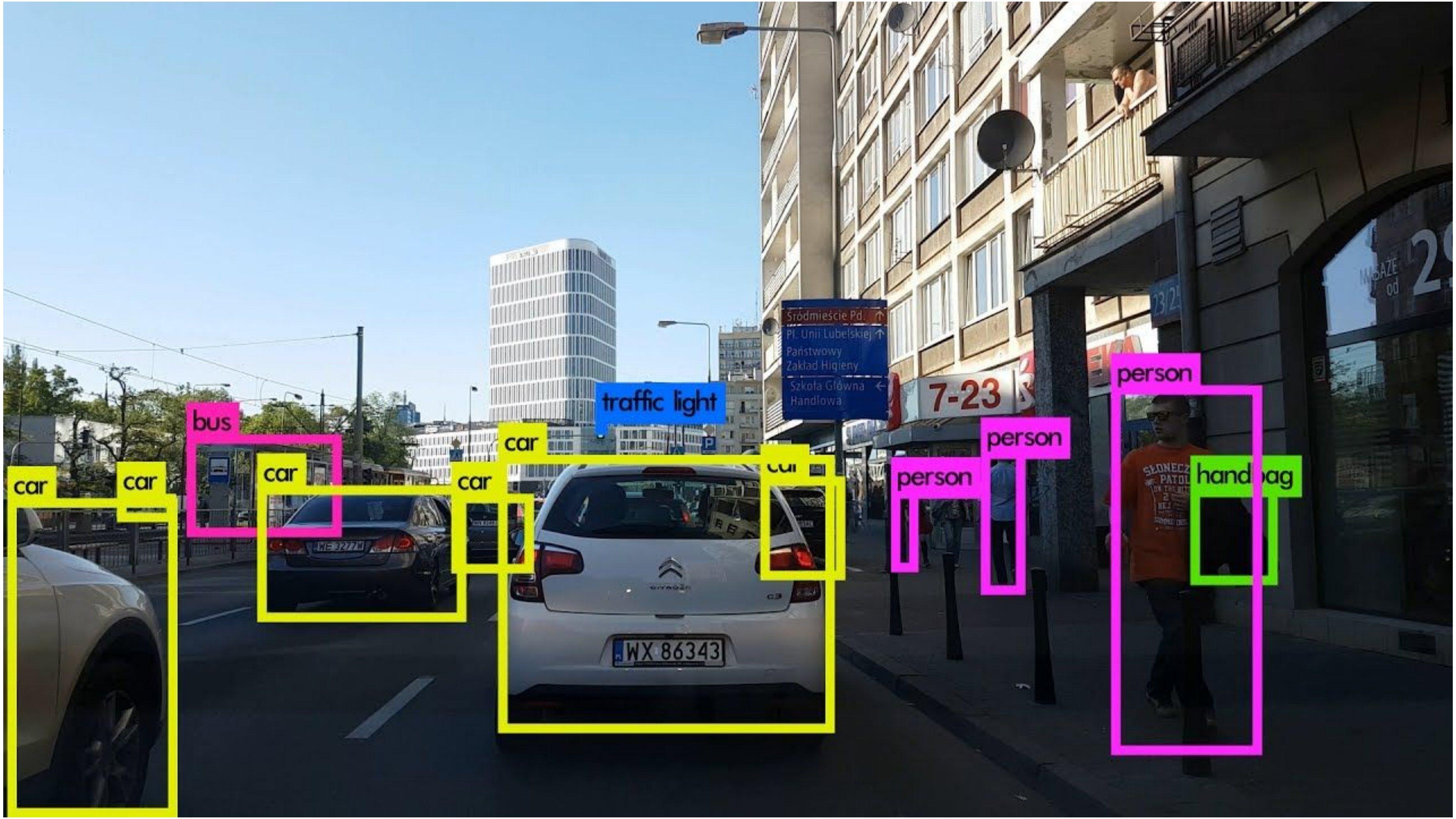
Baseline: Ensembling

- Majority Vote
  - Extremely basic: examining majority vote
- Bagging
  - essentially taking the mean of individual model submissions
- Ensembling highly uncorrelated, but weaker performing models is better!
- https://mlwave.com/kaggle-ensembling-guide/

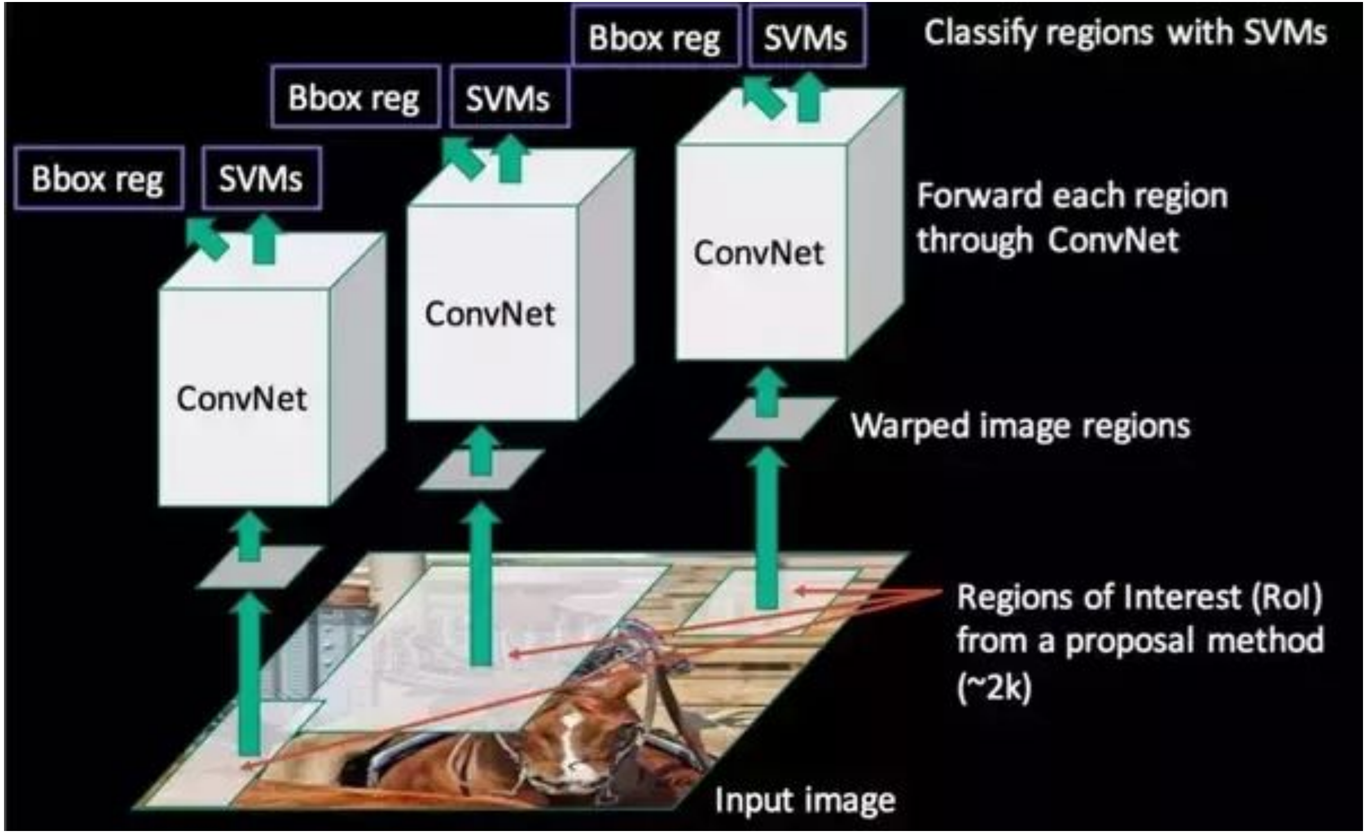
Baseline: Report & Presentation

- Baseline Expectations:
- Explanation of Justification of Pre-processing Techniques
- Explanation and justification of Algorithms
- Explanation and Justification of Ensembling Techniques
- Explanation and Justification of Basic Machine Learning Techniques
  - Hyperparameter Selection
  - Bias & Variance Tradeoff
  - Cross-Validation
  - Overfitting & Underfitting
  - Solution to Problem of Generalization

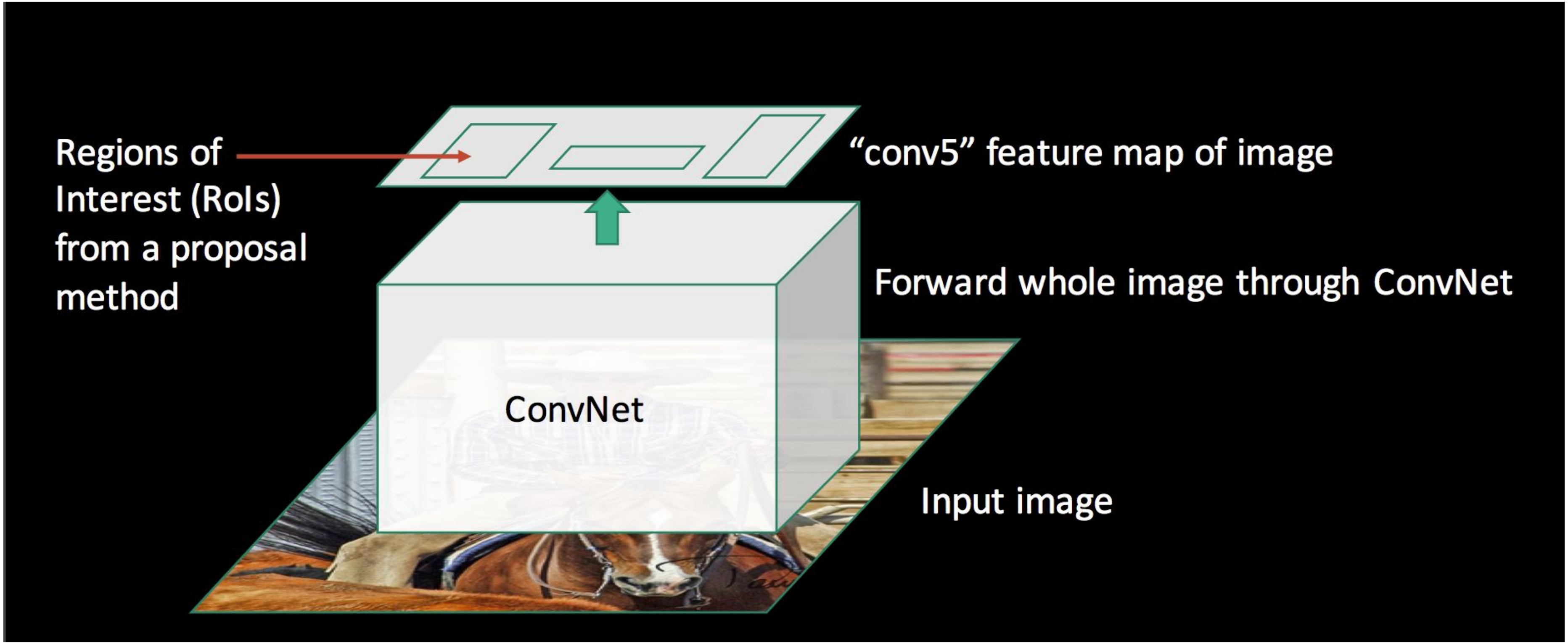
Object Detection


car
car
bus
car
car
car
traffic light
car
person
person
person
handbag
Śródmieście Pd.
Pl. Unii Lubelskiej
Państwowy
Zakład Higieny
Szkoła Główna
Handlowa
7-23
23/25
WE 3277W
WX 86343

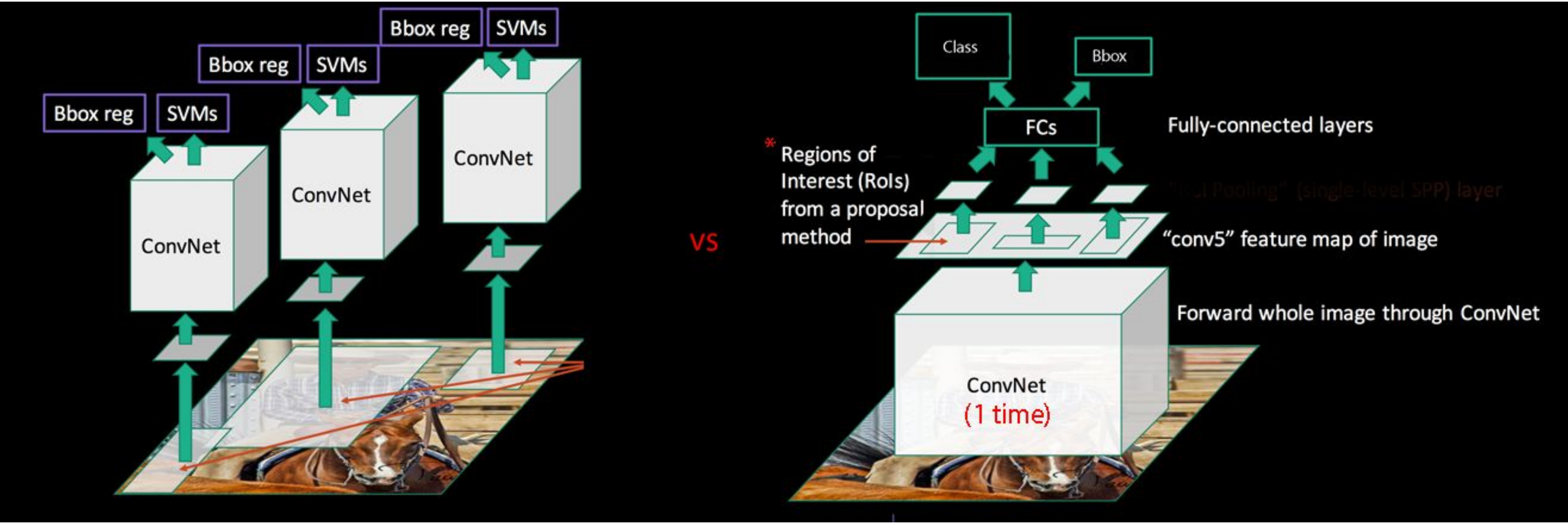
RCNN

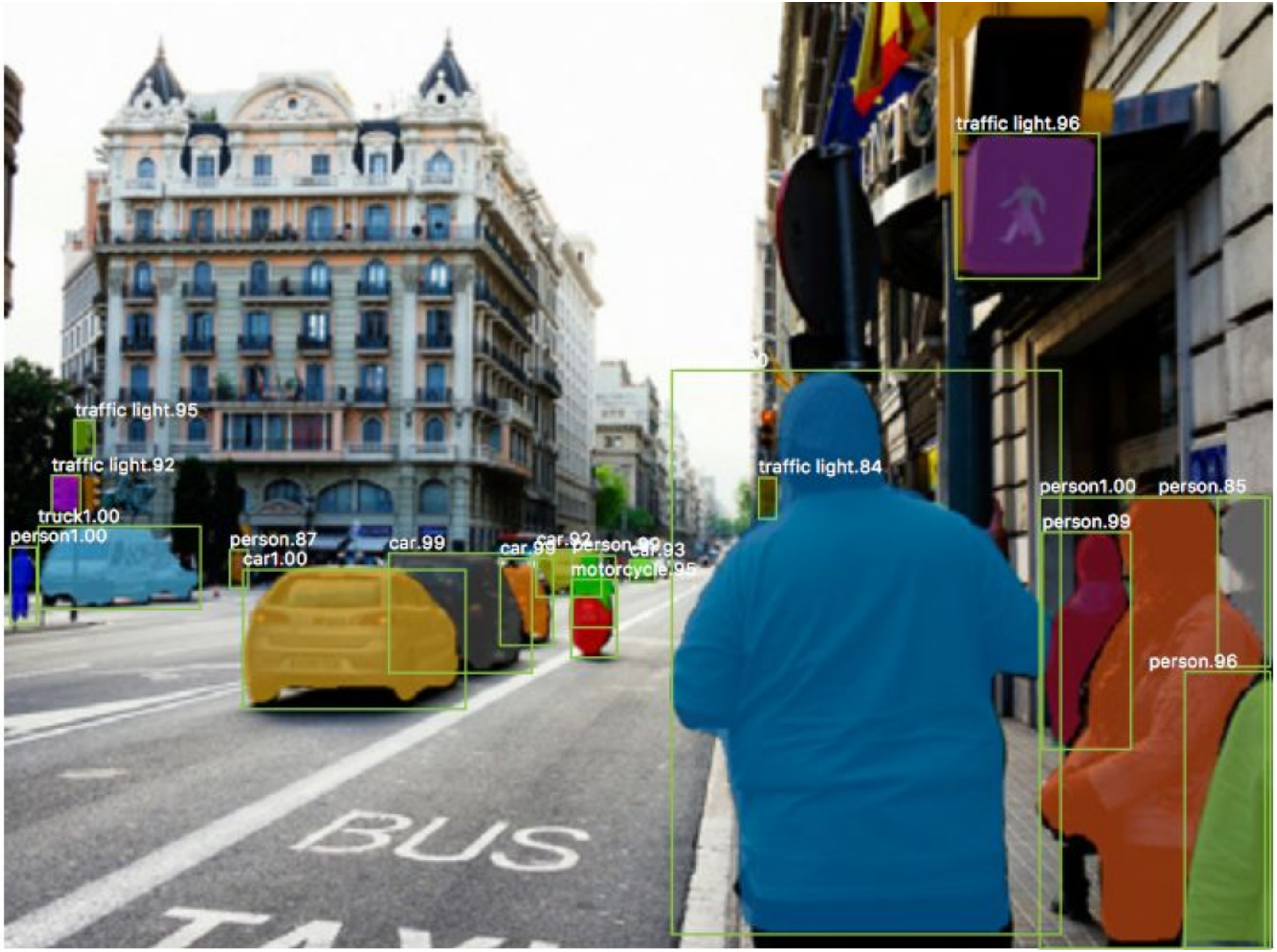

Bbox reg
SVMs
Classify regions with SVMs
Bbox reg
SVMs
Bbox reg
SVMs
ConvNet
ConvNet
ConvNet
Forward each region through ConvNet
Warped image regions
Regions of Interest (RoI) from a proposal method (~2k)
Input image

Faster-RCNN

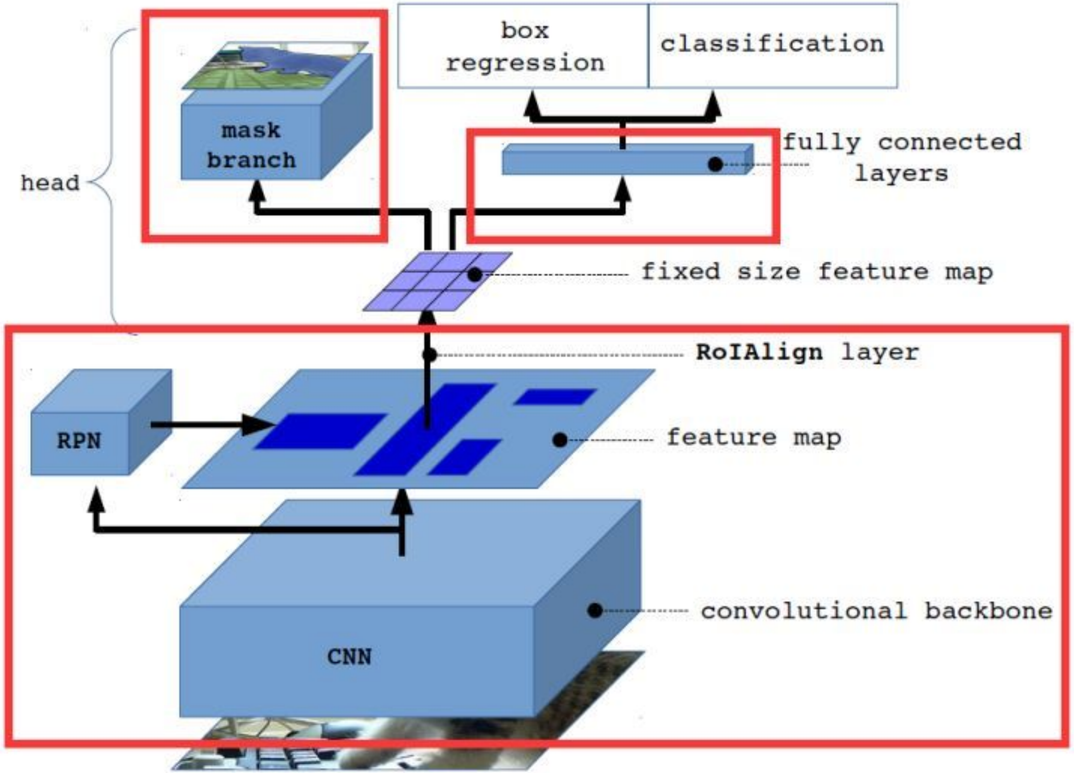

Regions of Interest (RoIs) from a proposal method
"conv5" feature map of image
Forward whole image through ConvNet
ConvNet
Input image

Faster-RCNN


Bbox reg
SVMs
Bbox reg
SVMs
Bbox reg
SVMs
ConvNet
ConvNet
ConvNet
VS
Class
Bbox
FCs
Fully-connected layers
* Regions of Interest (RoIs) from a proposal method
"RoI Pooling" (single-level SPP) layer
"conv5" feature map of image
Forward whole image through ConvNet
ConvNet
(1 time)


traffic light.96
traffic light.95
traffic light.92
truck1.00
person1.00
person.87
car1.00
car.99
car.99
car.92
person.99
car.93
motorcycle.95
traffic light.84
person1.00
person.85
person.99
person.96

Mask-RCNN


box regression
classification
mask branch
fully connected layers
head
fixed size feature map
RoIAlign layer
RPN
feature map
convolutional backbone
CNN

Other State-of-the-Art Models

- **YOLO**
- **Single-shot, Multi-object Detection**

Bonus: Super Learner

Set up the ensemble

- Specify a list of L base algorithms (with a specific set of model parameters).
- Specify a metalearning algorithm.

Train the ensemble

- Train each of the L base algorithms on the training set.
- Perform k-fold cross-validation on each of these learners and collect the cross-validated predicted values from each of the L algorithms.
- The N cross-validated predicted values from each of the L algorithms can be combined to form a new N x L matrix. This matrix, along wtih the original response vector, is called the "level-one" data. (N = number of rows in the training set)
- Train the metalearning algorithm on the level-one data.
- The "ensemble model" consists of the L base learning models and the metalearning model, which can then be used to generate predictions on a test set.

Bonus: Super Learner

Predict on new data

- To generate ensemble predictions, first generate predictions from the base learners.
- Feed those predictions into the metalearner to generate the ensemble prediction.

Bonus: Post-Processing

- Note that your test score is evaluated as **masks around individual cells**, not an entire mask on a input scan sampe
- This could lead to a double loss in your score
- Use classic computer vision techniques (for e.g. edge detection) to segment and identify masks blobs for individual cells

Recommendations: Workflow

- Amazon Web Service EC2 | p2.xlarge
- https://aws.amazon.com/ec2/instance-types/p2/
- Jet's CUDA-CuDNN-Keras-Torch Bootstrap Installer
- https://github.com/abhmul/InstallCUDA-Kerasv2
- Dev Kit: Keras, numpy, jupyter, skimage
- https://chrisalbon.com/software_engineering/cloud_computing/run_project_jupyter_on_amazon_ec2/
- Advanced Dev Kit: pytorch
- Masked R-CNN, U-Net: Use Working Implementations with Citations
- Technical Blogs for Bonus Players

Recommendations: Workflow

- **Stop instance**, don’t terminate instance
- This saves disk state on your AWS virtual machine
- Avoid developing library code for U-Net and Masked R-CNN
- It’s great if you want application development practice, but a vanilla Masked R-CNN implementation would require a week to write and test
- You are graded on:
- Application & Explanation of Models
- Application & Explanation of Pre/post-processing, Ensembling
- Make sure you explain them in text (report) and verbally (presentation)
- Avoid reading technical papers. **Blogs and kaggle discussion boards** are more time-efficient and human-friendly