DATA SCIENCE INTERVIEW PREPARATION (30 Days of Interview Preparation)

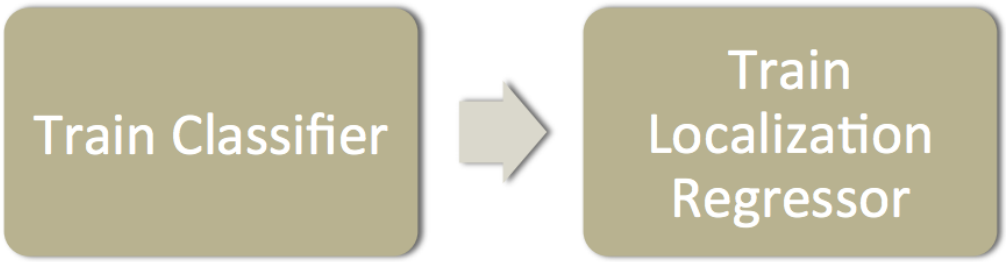
DAY 23

Q1.Explain Overfeat in Object detection.

Answer:

Overfeat: It is a typical model of integrating object detection, localization, and classification tasks whole into one convolutional neural network(CNN). The main idea is to do image classification at different locations on regions of multiple scales of the image in a sliding window fashion, and second, predict bounding box locations with the regressor trained on top of the same convolution layers.

This model architecture is too similar to AlexNet. This model is trained as follows:



1. Train a CNN model (identical to AlexNet) on image classification tasks.
2. Then, we replace top classifier layers by the regression network and trained it to predict object bounding boxes at each spatial location and scale. Regressor is class-specific, each generated for one class image.

- Input: Images with classification and bounding box.
- Output: (xleft,xright,ytop,ybottom)(xleft,xright,ytop,ybottom), 4 values in total, representing the coordinates of the bounding box edges.
- Loss: The regressor is trained to minimize l2 norm between the generated bounding box and truth for each training example.

At the detection time,

1. It Performs classification at each location using the pretrained CNN model.
2. It Predicts object bounding boxes on all classified regions generated by the classifier.
3. Merge bounding boxes with sufficient overlap from localization and sufficient confidence of being the same object from the classifier.

Q2. What is Multipath: Multiple Probabilistic Anchor Trajectory Hypotheses for Behavior Prediction?

Answer:

In this paper, we focus on problem of predicting future agent states, which is the crucial task for robot planning in real-world environments. We are specifically interested in addressing this problem for self-driving vehicles, application with a potentially enormous societal impact. Mainly, predicting the future of other agents in this domain is vital for safe, comfortable, and efficient operation. E.g., it is important to know whether to yield to the vehicle if they are going to cut in front of our robot or when would be the best time to add into traffic. Such future prediction requires an understanding of a static and dynamic world context: road semantics (*like* lane connectivity, stop lines), traffic light informations, and past observations of other agents, as in below Fig.

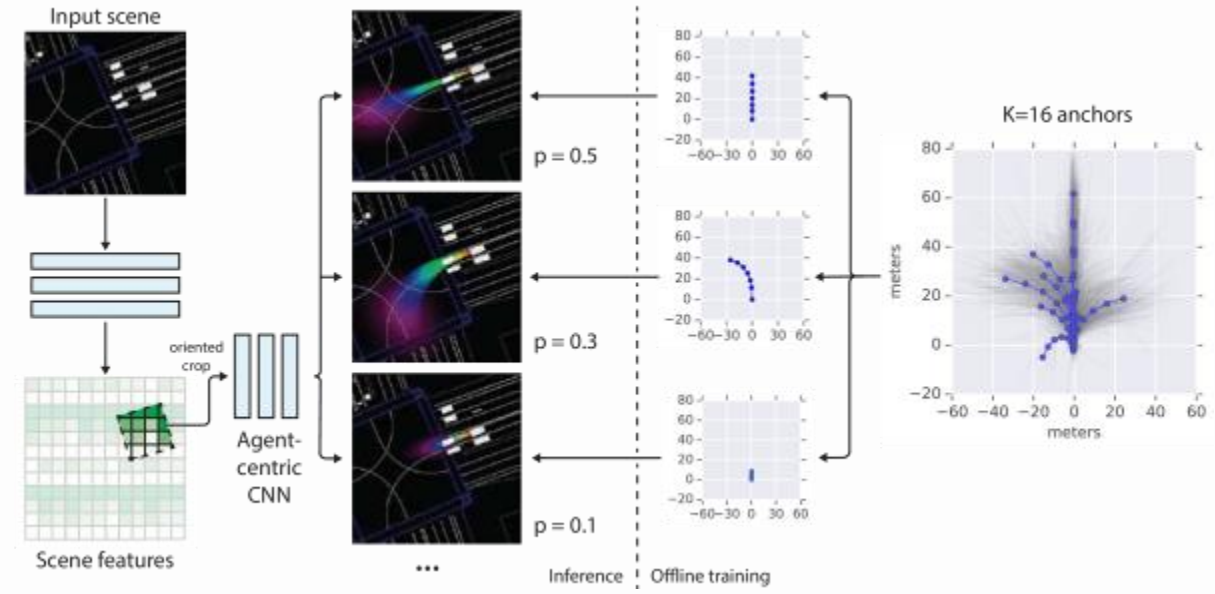
A fundamental aspect of the future state prediction is that it is inherently *stochastic*, as agents can't know each other's motivations. When we are driving, we can never really be sure what other drivers will do next, and it is essential to consider multiple outcomes and their likelihood.

We seek the model of the future that can provide both (i) a weighted, parsimonious set of discrete trajectories that covers space of likely outcomes and (ii) a closed-form evaluation of the likelihood of any trajectory. These two attributes enable efficient reasoning in relevant planning use-cases, e.g., human-like reactions to discrete trajectory hypotheses (*e.g.*, yielding, following), and probabilistic queries such as the expected risk of collision in a space-time region.

This model addresses these issues with critical insight: it employs a fixed set of *trajectory anchors* as the basis of our modeling. This lets us factor stochastic uncertainty hierarchically: First, *intent uncertainty* captures the uncertainty of *what* an agent intends to do and is encoded as a distribution over the set of anchor trajectories. Second, given an intent, *control uncertainty* represents our uncertainty over *how* they might achieve it. We assume control uncertainty is normally distributed at each future time step [Thrun05], parameterized such that the mean corresponds to a context-specific offset from the anchor state, with the associated covariance capturing the unimodal aleatoric uncertainty [Kendall17]. In Fig. Illustrates a typical scenario where there are three likely intents given the scene context, with control mean offset refinements respecting road geometry, and control uncertainty intuitively growing over time.

Our trajectory anchors are modes found in our training data in state-sequence space via unsupervised learning. These anchors provide templates for coarse-granularity futures for an agent and might correspond to semantic concepts like "change lanes," or "slow down" (although to be clear, we don't use any semantic concepts in our modeling).

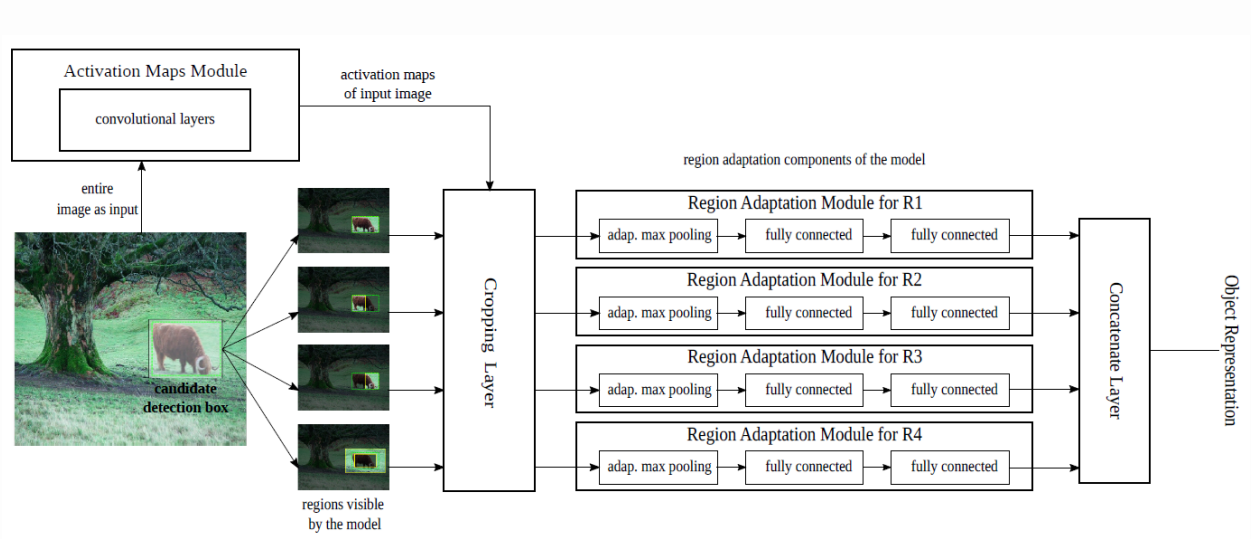
Our complete model predicts a Gaussian mixture model (GMM) at each time step, with the mixture weights (intent distribution) fixed over time. Given such a parametric distribution model, we can directly evaluate the likelihood of any future trajectory and have a simple way to obtain a compact, diverse weighted set of trajectory samples: the MAP sample from each anchor-intent.



Q3. An Object detection approach using MR-CNN

Answer:

Multi-Region CNN (MR-CNN): Object representation using multiple regions to capture several different aspects of one object.

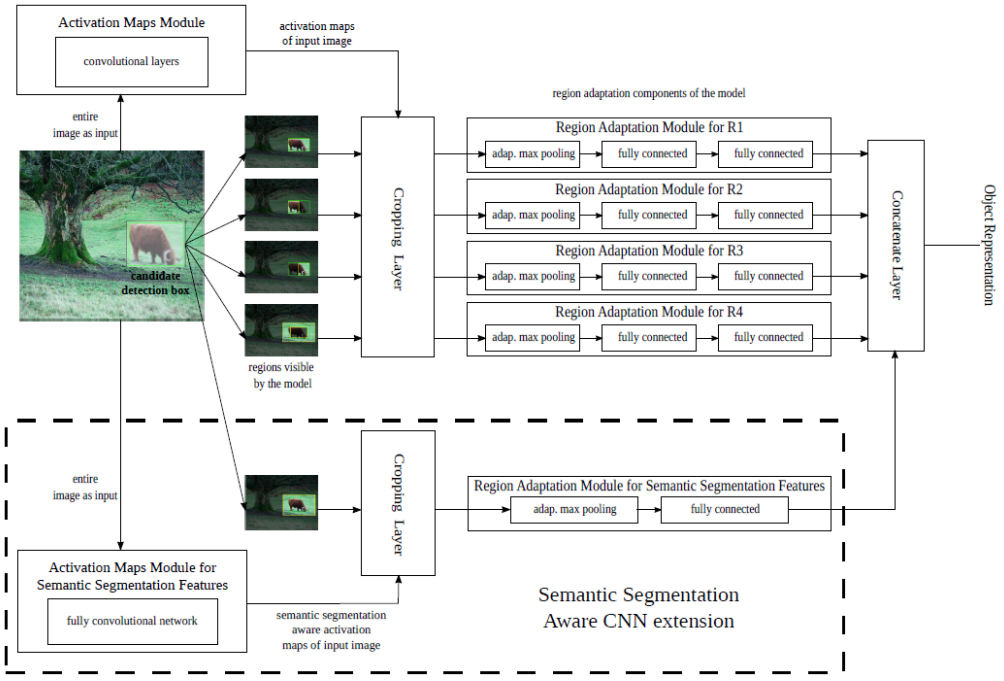


Network Architecture of MR-CNN

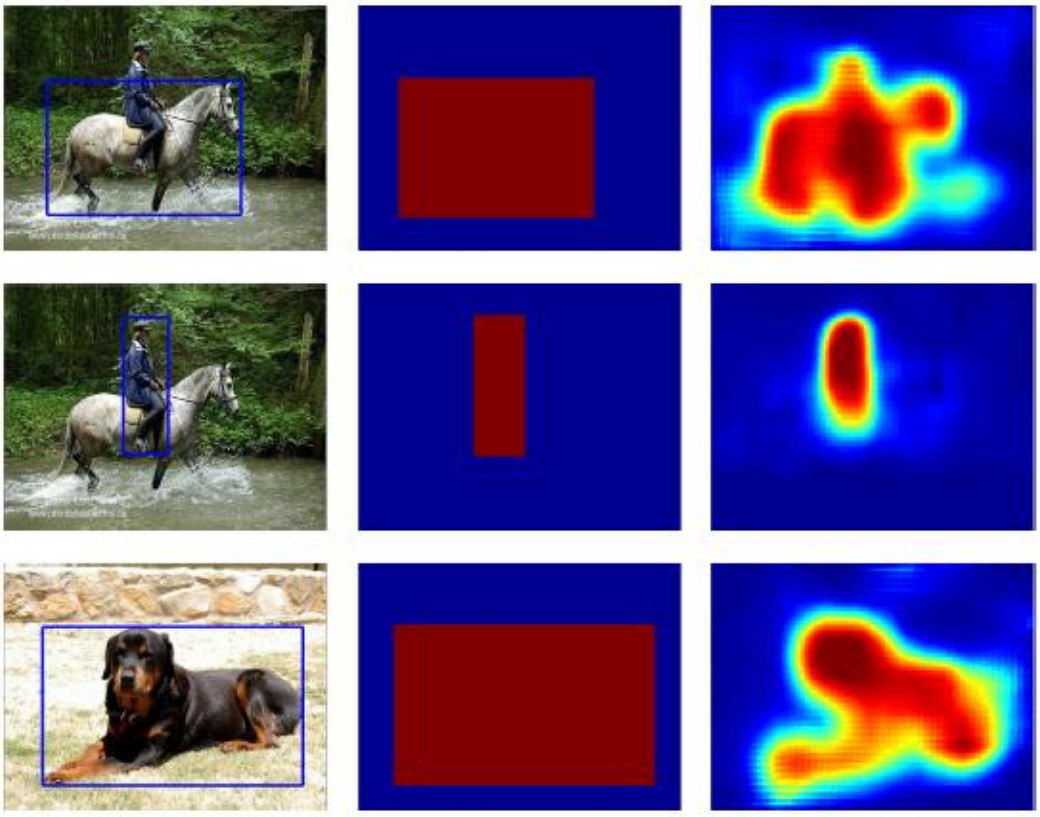
- First, the input image goes through Activation Maps Module, as shown above, and outputs the activation map.
- **Bounding box** or Region proposals candidates are generated using Selective Search.
- For each bounding box candidate *B*, a set of regions {*Ri*}, with *i*=1 to *k,* are generated, that is why it is known as multi-region. More details about the choices of multiple areas are described in next sub-section.
- ROI pooling is performed for each region *Ri*,cropped or pooled area goes through fully connected (FC) layers, at each Region Adaptation Module.
- Finally, the output from all FC layers are added together to form a 1D feature vector, which is an object representation of the bounding box *B*.
- Here, VGG-16 ImageNet pre-trained model is used. The max-pooling layer after the last conv layer is removed.

Q4. Object detection using Segmentation-aware CNN

Answer:



- There are close connections between segmentation and detection. And segmentation related ques are empirically known to help object detection often.
- Two modules are added: **Activation maps module for semantic segmentation-aware features**, and **regions adaptation module for grammarly segmentation-aware feature.**
- There is no additional annotation used for training here.
- FCN is used for an activation map module.
- The last FC7 layer channels number is changed from 4096 to 512.



- **The weakly supervised training** strategy is used. **Artificial foreground class-specific segmentation mask is created using bounding box annotations**.

- More particularly, the ground truth bounding boxes of an image are projected on the spatial domain of the last hidden layer of the FCN, and the ”pixels” that lay inside the projected boxes are labelled as foreground while the rest are labelled as background.

- After training the FCN using the mask, the last classification layer is dropped. Only the rest of FCN is used.

- Though it is weakly supervised training, the foreground probabilities shown as above still carry some information, as shown above.

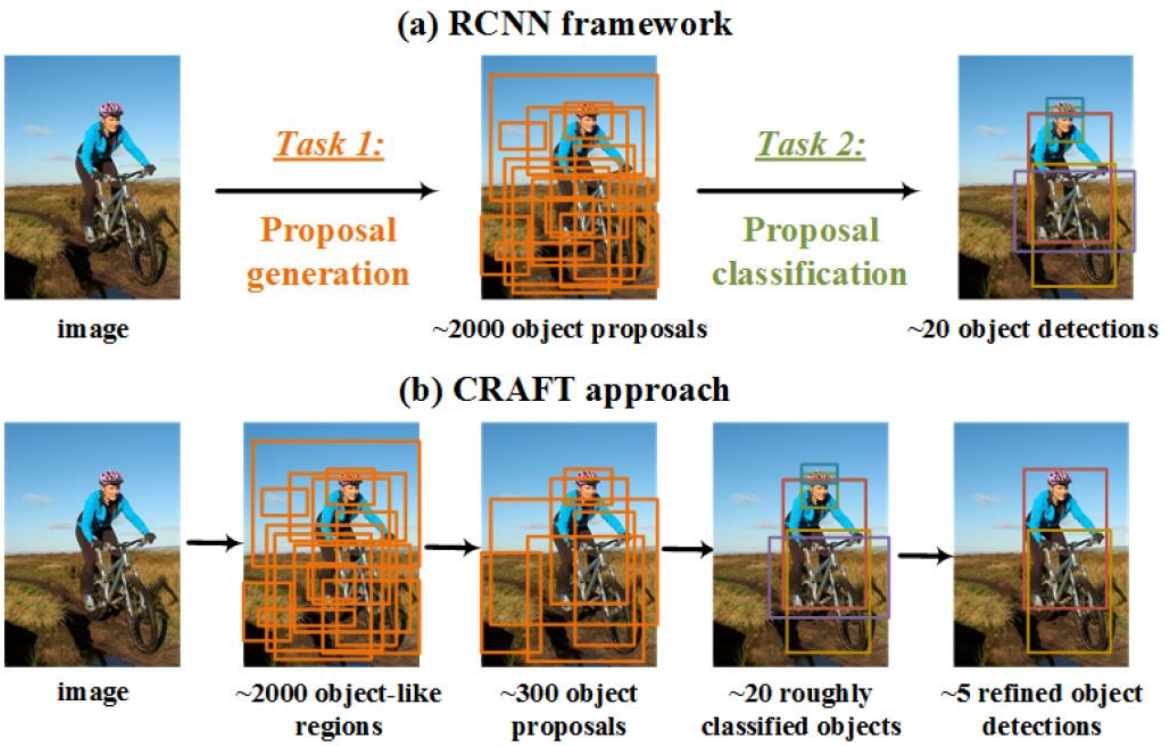
- The bounding box used is 1.5× larger than the original bounding box.

Q5. What is CRAFT (Object detection)?

Answer:

CRAFT stands for Cascade **R**egion-proposal-network **A**nd **F**as**T** R-CNN. It is reviewed by the Chinese Academy of Sciences **and** Tsinghua University. In Faster R-CNN, region proposal network is used to generate proposals. These proposals, after ROI pooling, are going through network for classification. However, CRAFT is found that there is a core problem in Faster R-CNN:

- In proposal generation, there is still a large proportion of background regions. The existence of many background sample causes many false positives.



In CRAFT(**C**ascade **R**egion-proposal-network**),** as shown above, another CNN(Convolutional neural network) is added after RPN to generate fewer proposals (i.e., 300 here)**.** Then, classification is performed on 300 proposals and outputs about 20 first detection results. For each primitive result**,** refined object detection is performed using one-vs-rest classification.

Cascade Proposal Generation

Baseline RPN

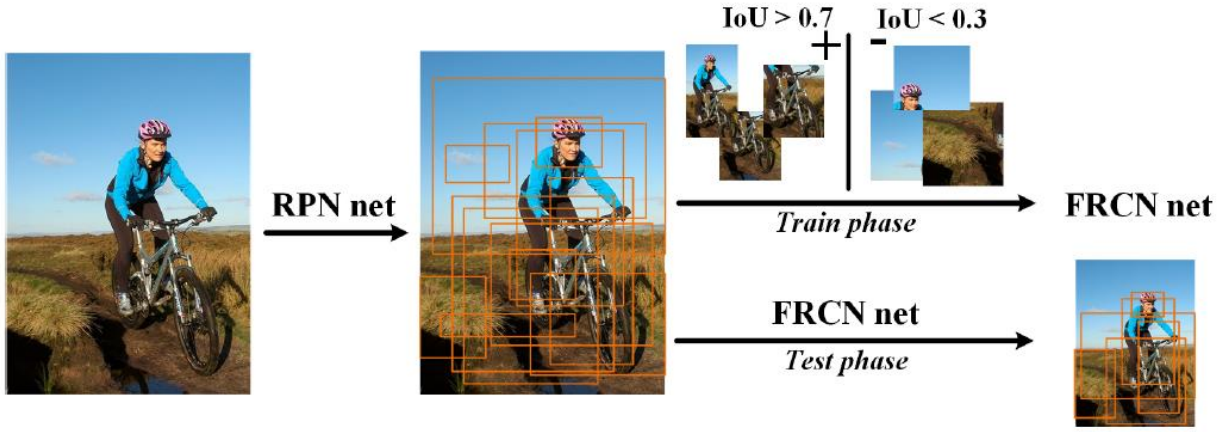
- An ideal proposal generator should generate as few proposal as possible while covering almost all object instances. Due to resolution loss caused by CNN pooling operation and the fixed aspect ratio of the sliding window, RPN is weak at covering objects with extreme shapes or scales.

aero	bike	**bird**	**boat**	**bottle**
95.44	98.81	**93.90**	**92.78**	**80.38**
bus	car	cat	**chair**	cow
98.12	96.00	99.16	**91.80**	99.18
table	dog	horse	mbike	persn
95.15	99.59	97.70	96.31	95.49
plant	sheep	sofa	train	**tv**
86.87	98.76	98.74	97.52	**90.58**

Recall Rates (is in %), Overall is 94.87%, lower than 94.87% is bold in the text.

- The above results are baseline RPN based on VGG_M trained using PASCAL VOC 2007 train+val, and tested on the test set.
- The recall rate on each object category varies a lot. Object with extreme aspect ratio and scale are hard to be detected, such as boat and bottle.

Proposed Cascade Structure



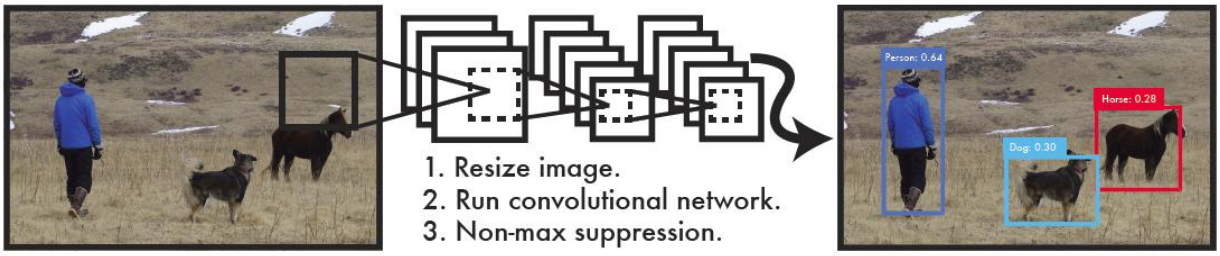
The concatenation classification network after RPN is denoted as FRCN Net here

- An additional classification network that comes after RPN.
- The additional network is the 2- class detection network denoted as FRCN net in above figure. It uses output of RPN as training data.
- After RPN net is trained, the 2000 first proposals of each training image are used as training data for the FRCN net.
- During training, +ve and -ve sampling are based on 0.7 IoU for negatives and below 0.3 IoU for negatives, respectively.
- **There are 2 advantages:**
- 1) First, additional FRCN net further **improves quality of the object proposals** and **shrinks more background regions**, making proposals fit better with task requirement.
- 2) Second, **proposals from multiple sources can be merged** as the input of the FRCN net so that complementary information can be used.

Q6. Explain YOLOv1 for Object Detection.

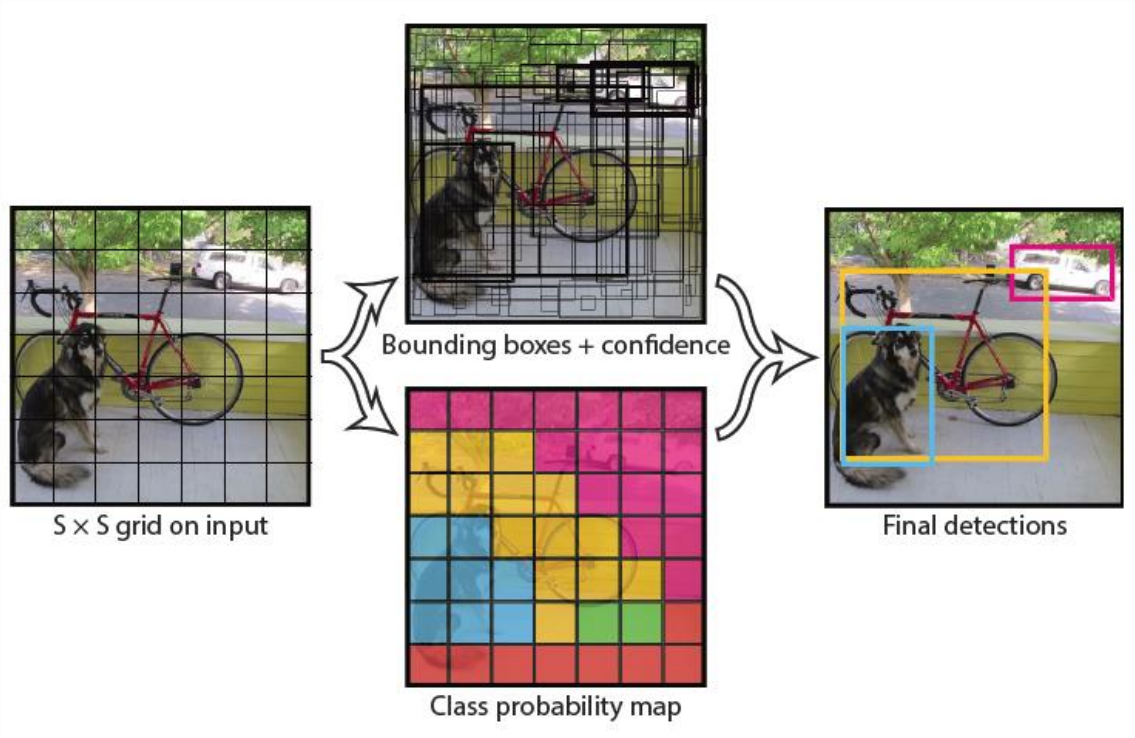
Answer:

YOLOv1 stands for You Look Only Once, it is reviewed by FAIR (Facebook AI Research). The network only looks at the image once to detect multiple objects.



By just looking image once, the detection speed is in real-time (45 fps). Fast YOLOv1 achieves 155 fps.

YOLO suggests having a unified network to perform all at once. Also, an end-to-end training network can be achieved.



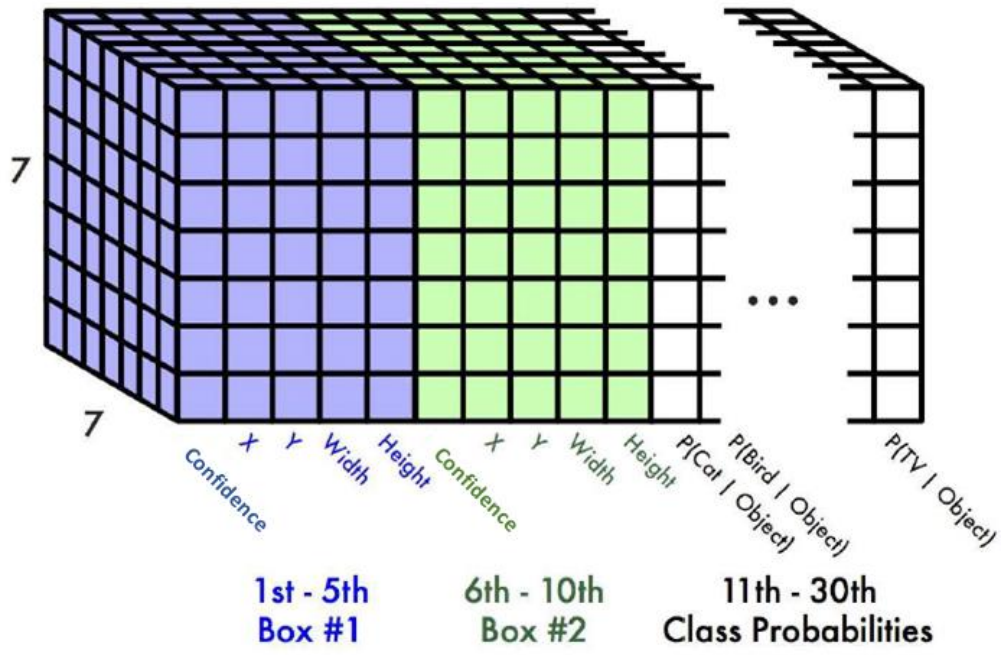
The input image is divided into the S×S grid (S=7). If the center of the object falls into the grid cell, that grid cell is responsible for detecting that object.

Each grid cell predict B bounding boxes (B=2) and confidence scores for those boxes. These confidence score reflect how confident model is that the box contains an object, i.e., any objects in the box, P(Objects).

Each bounding box consists of five predictions: x, y, w, h, and confidence.

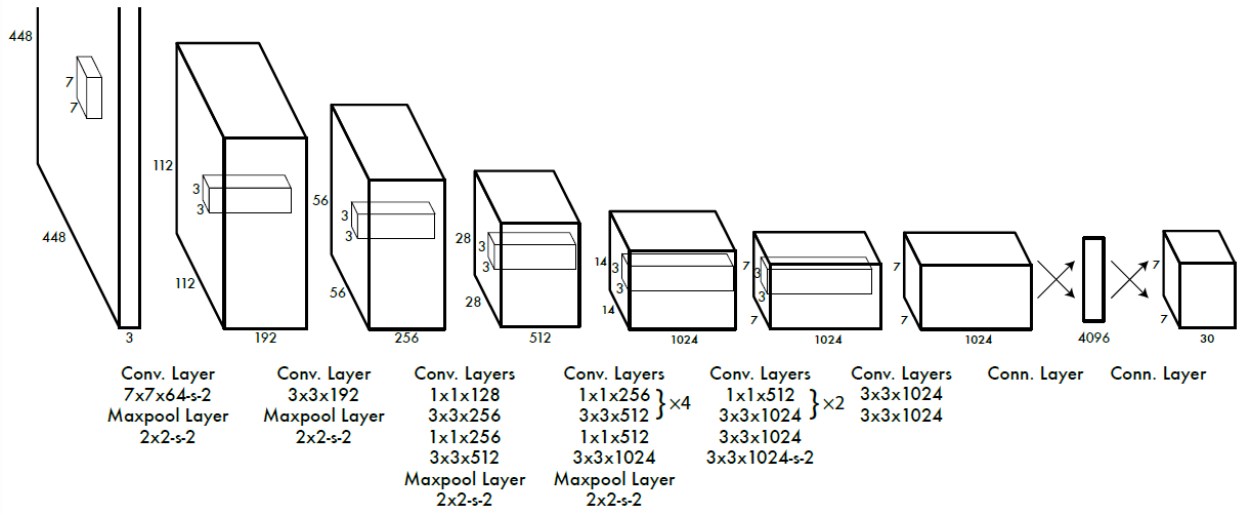
- The (x, y) coordinates represent center of the box relative to the bound of the grid cell.
- The height h and width w are predicted relative to whole image.
- The confidence represents the IOU (Intersection Over Union) between the predicted box and any ground truth box.

Each grid cell also predicts conditional class probabilities, P(Class|Object). (Total number of classes=20)



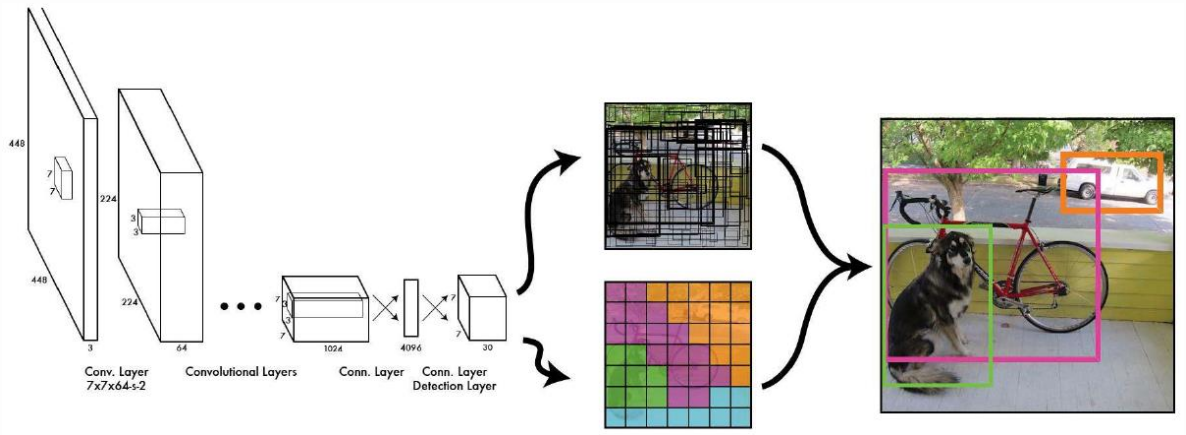
The output size becomes: 7×7×(2×5+20)=1470

Network Architecture of YOLOv1



The model consists of 24 convolutional layers, followed by two fully connected layers. Alternating 1×1 convolutional layers reduce features space from preceding layers. (1×1)Conv has been used in GoogLeNet for reducing the number of parameters.)

Fast YOLO fewer convolutional layers (9 instead of 24) and fewer filters in those layers. The network pipeline is summarized like below:



Therefore, we can see that the input image goes through network once and then objects can be detected. And we can have end-to-end learning.

Q7. Adversarial Examples Improve Image Recognition

Answer:

Adversarial examples crafted by adding imperceptible perturbations to images can lead to (ConvNets)Convolutional Neural Networks to make wrong predictions. The existence of adversarial examples not only reveal limited generalization ability of ConvNets, but also poses security threats on the real-world deployment of these models. Since the first discovery of the vulnerability of ConvNets to adversarial attacks, many efforts have been made to improve network robustness.

Above Fig.: AdvProp improves image recognition. By training model on ImageNet, AdvProp helps EfficientNet-B7 to achieve 85.2% accuracy on ImageNet, 52.9% mCE (mean corruption error, lower is better) on ImageNet-C, 44.7% accuracy on ImageNet-A and 26.6% accuracy on Stylized-ImageNet, beating its vanilla counterpart by 0.7%, 6.5%, 7.0% and 4.8%, respectively. Theses sample images are randomly selected from category "goldfinch."

In this paper, rather than focusing on defending against adversarial examples, we shift our attention to leveraging adversarial examples to improve accuracy. Previous works show that training with adversarial examples can enhance model generalization but are restricted to certain situations—the improvement is only observed either on small datasets (*e.g.*, MNIST) in the fully-supervised setting [5], or on larger datasets but in the semi-supervised setting [21, 22]. Meanwhile, recent works [15, 13, 31] also suggest that training with adversarial examples on large datasets, *e.g.*, ImageNet [23], with supervised learning results in performance degradation on clean images. To summarize, it remains an open question of how adversarial examples can be used effectively to help vision models.

We observe all previous methods jointly train over clean images and adversarial examples without distinction, even though they should be drawn from different underlying distributions. We hypothesize this distribution mismatch between fresh examples and adversarial examples is a key factor that causes performance degradation in previous works.

Q8. Advancing NLP with Cognitive Language Processing Signals

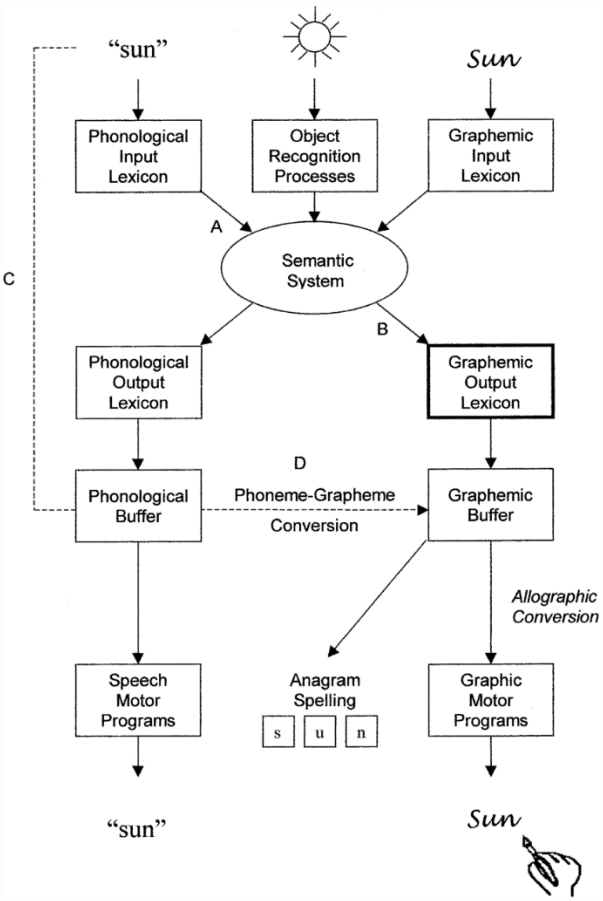
Answer:

When reading, humans process language "automatically" without reflecting on each step — Humans string words together into sentences, understand the meaning of spoken and written ideas, and process language without overthinking about how the underlying cognitive process happens. This process generates cognitive signals that could potentially facilitate natural language processing tasks.

In recent years, collecting these signals has become increasingly accessible and less expensive Papoutsaki et al. (2016); as a result, using cognitive features to improve NLP tasks has become more popular. For example, researchers have proposed a range of work that uses eye-tracking or gaze signals to improve part-of-speech tagging (Barrett et al., 2016), sentiment analysis (Mishra et al., 2017), named entity recognition Hollenstein and Zhang (2019), among other tasks. Moreover, these signals have been used successfully to regularize attention in neural networks for NLP Barrett et al. (2018).

However, most previous work leverages only eye-tracking data, presumably because it is the most accessible form of cognitive language processing signal. Also, most state-of-the-artwork(SOTA) focused on improving a single task with a single type of cognitive signal. But can cognitive processing signals bring consistent improvements across modality (e.g., eye-tracking and EEG) and across various NLP

tasks? And if so, does the combination of different sources of cognitive signals bring incremental improvements?



Q8. Do you have any idea how can we use NLP on News headlines to predict index trends?

Answer:

Traders generally look up information about the company they are looking to buy shares into, for long and short trading. A frequent source of information is news media, which provides updates about the company's activities, such as expansion, better or worse revenues than expected, new products and much more. Depending on the news, trader can determine a bearish or bullish trend and decide to invests in it.

We may be able to correlate overall public sentiments towards as the company and its stock price: Apple is generally well-liked by the public, receives daily news coverage of its new product and financial stability, and its stock has been growing steadily. These facts may be correlated but first may not cause the second; we will analyze if news coverage can be used to predict the market trend. To do so, we will examine the top 25 news headlines of each open-market day from 2008 to late 2015 and try to predict

the end-of-day value of DJIA index for the same day. The theory behind predicting same day value is that traders will respond to news quickly and thus, the market will adjust within an hour of release. Therefore in the single business day, if the news is spread during business hours, its effect may be measured before closing bell of the market.

The motivation behind this analysis is that humans take decision using most of the available information. This usually takes several minutes to discover new information and take the decision. An algorithm is capable of processing gigabytes of texts from multi-source streams in second. We could potentially exploit this difference in order to create a trading strategy.

NLP (Natural Language Processing) techniques can be used to extract different information from headlines such as sentiment, subjectivity, context and named entities. We obtain an indicator vector using each of these techniques, which allows us to train different algorithms to predict a trend. To predict these values, we can use several methods that should be well suited for this type of information: Linear regression, Support Vector Machine(SVM), Long Short-Term Memory(LSTM) recurrent neural network, and a dense feed-forward (MLP) neural network. We included techniques used by Bollen et al (2010), which resulted in state-of-the-art(SOTA) results. We will also analyze the method used in other studies with a similar context

Information in headlines

Latent Sentiment Analysis is done by building up a corpus of labeled words which usually connote a degree of +ve or -ve sentiment. We can extend the corpus to include emoticons (i.e. ":-)") and expression, which often correlates to strong emotions. Naive sentiment analysis consists of a lookup of each word in sentence to be analyzed and evaluation of a score for sentence overall. This approach is limited by its known vocabulary, which can be mitigated by context analysis and introduction of synonyms.Second limitation is sarcasm, which is prevalent in twitter feed analysis. The sentiment inferred by words is opposed to the sentiment assumed by the user. This is mitigated by technique detecting sarcasm which

lead to a polarity flip of such tweets.

Sentiment analysis gives insight on how favorable the media is and maybe the bias traders may have towards buying or selling.

Another NLP technique which gave promising results was context analysis. This is a recent deep learning approach where you rely on a large corpus of text in order to learn and predict the words around a target. You can then deduce in what context it usually appears. The result is a vector representing each word. Other vectors with little distance are usually synonyms. The representation also allows us to do algebra, such as the famous "king - man + woman = queen"

Learning this representation offers the possibility of associating a specific context with a bullish or bearish market.