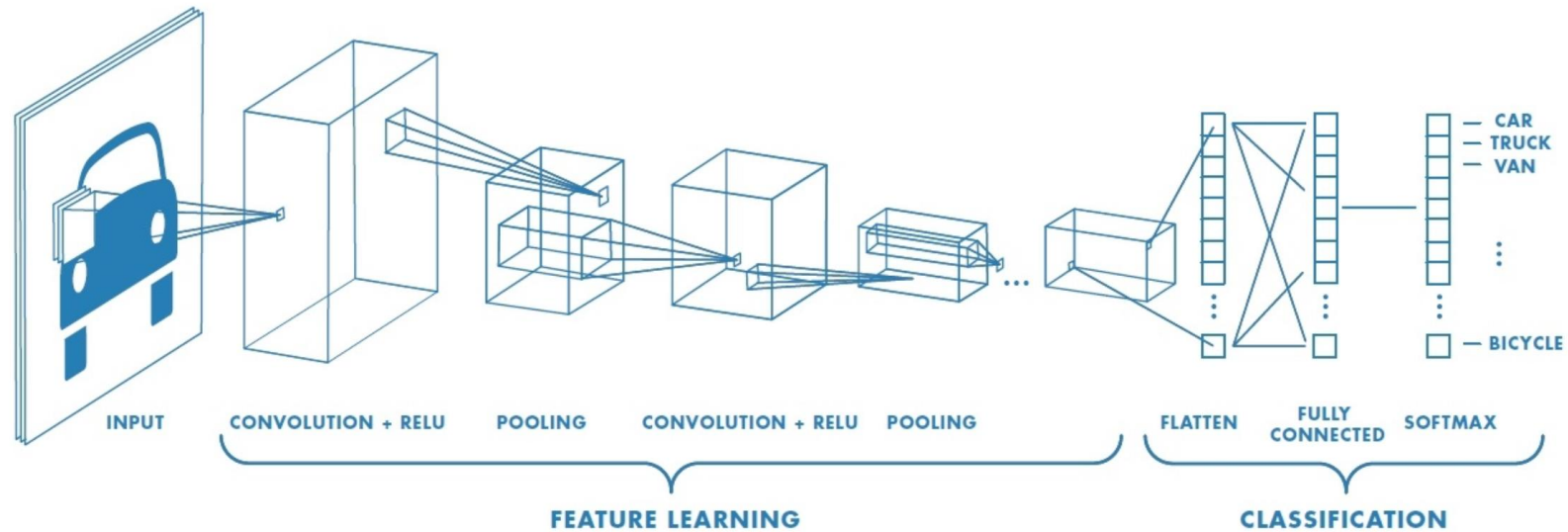# Beyond Convolutional Neural Networks (CNN)
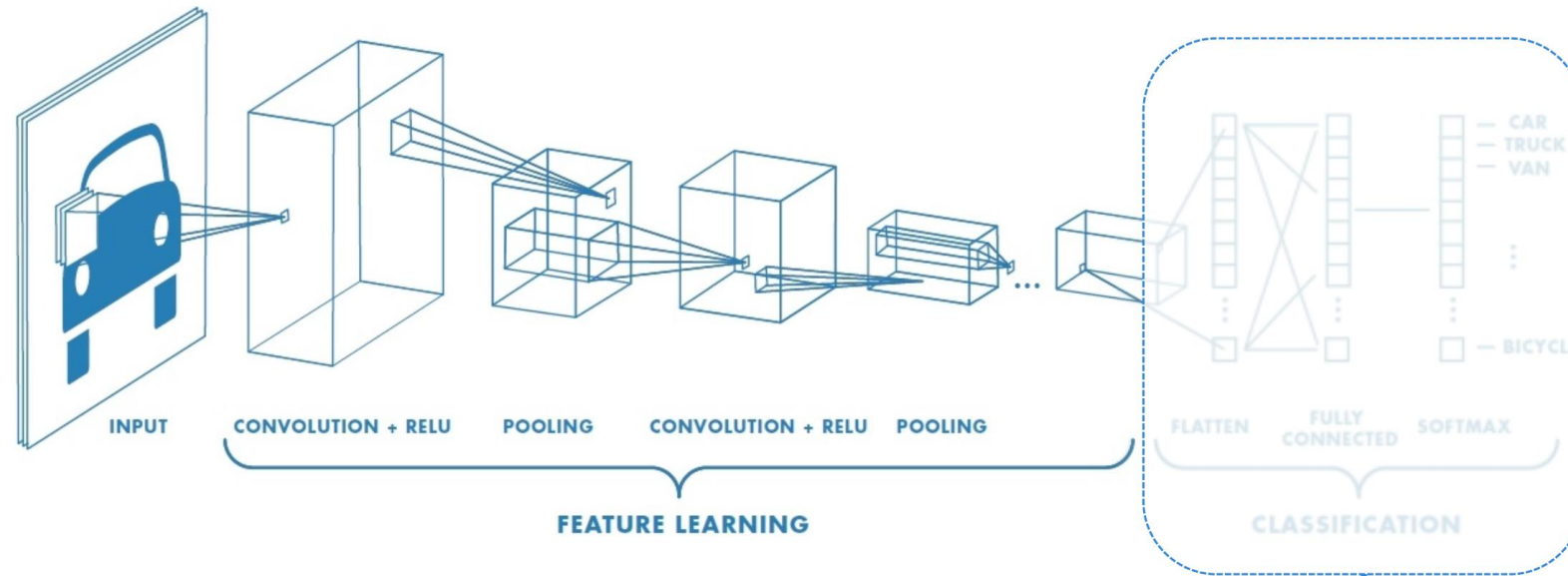
**Industrial AI Lab.**

**Prof. Seungchul Lee**

# Beyond Classification

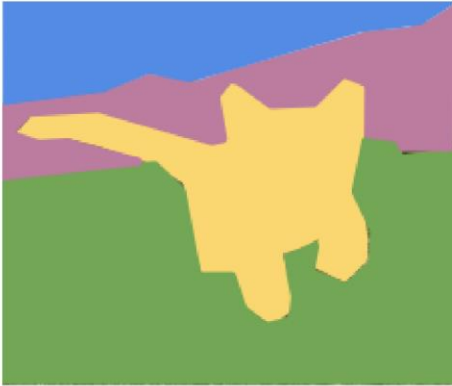# An Architecture for Many Applications

# An Architecture for Many Applications



Different image classification domains
Object detection with R-CNNs
Segmentation with fully convolutional networks
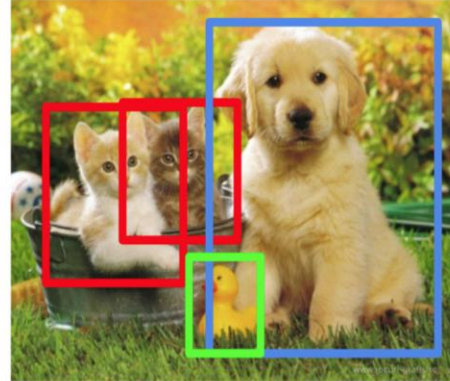Image captioning with RNNs

# Beyond Classification

### Semantic Segmentation



CAT

### Object Detection



CAT, DOG, DUCK

### Image Captioning



The cat is in the grass.

### Instance Segmentation
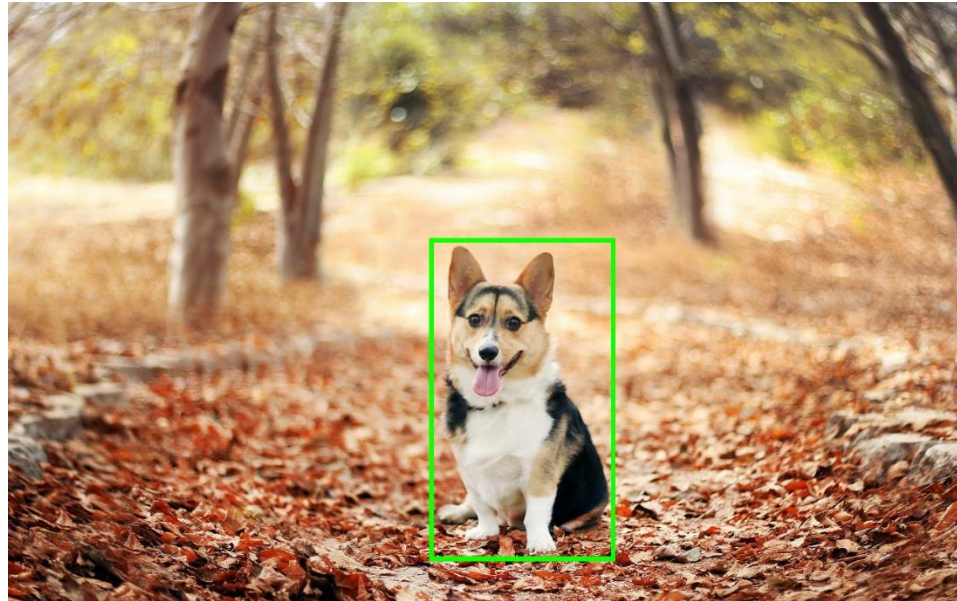


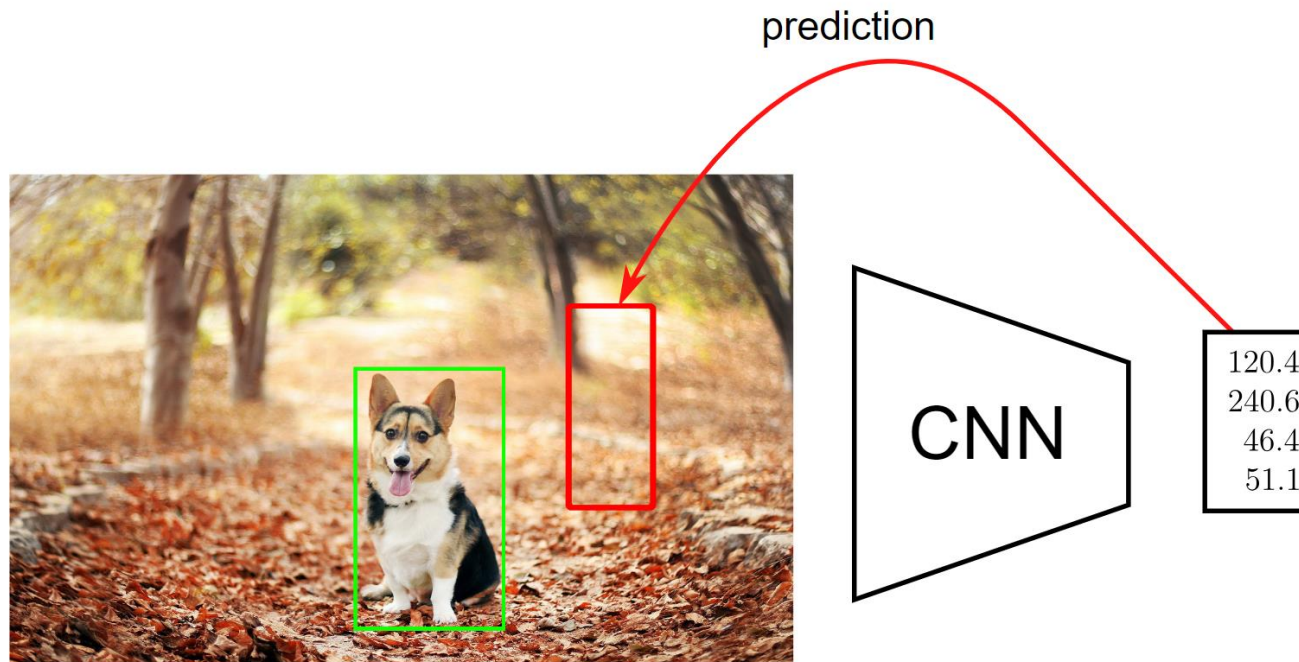CAT, DOG, DUCK

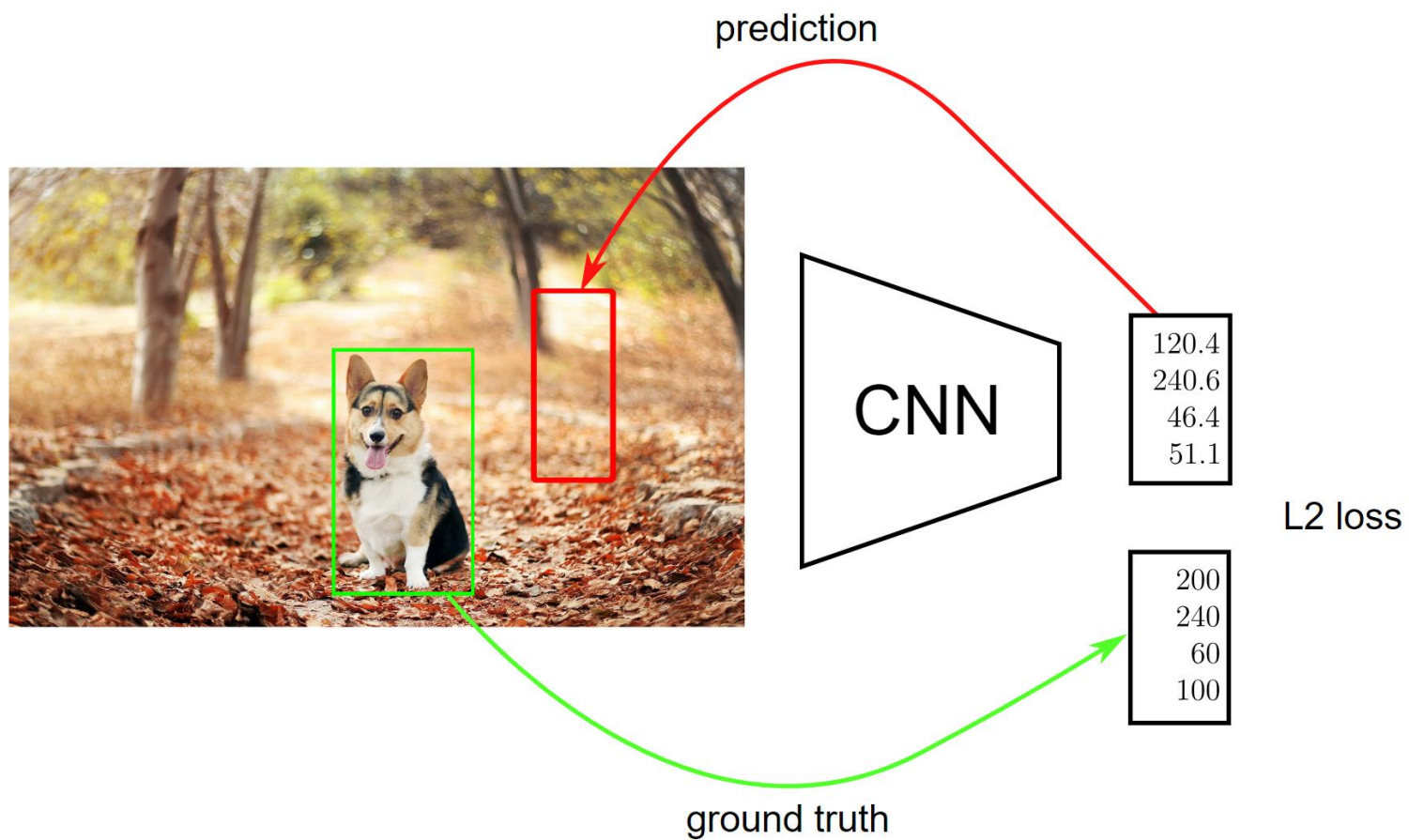### Classification + Localization



CAT

# Localization

- Single object per image
- Predict coordinates of a bounding box (x, y, w, h)
- Evaluate via Intersection over Union (IoU)

# Localization as Regression
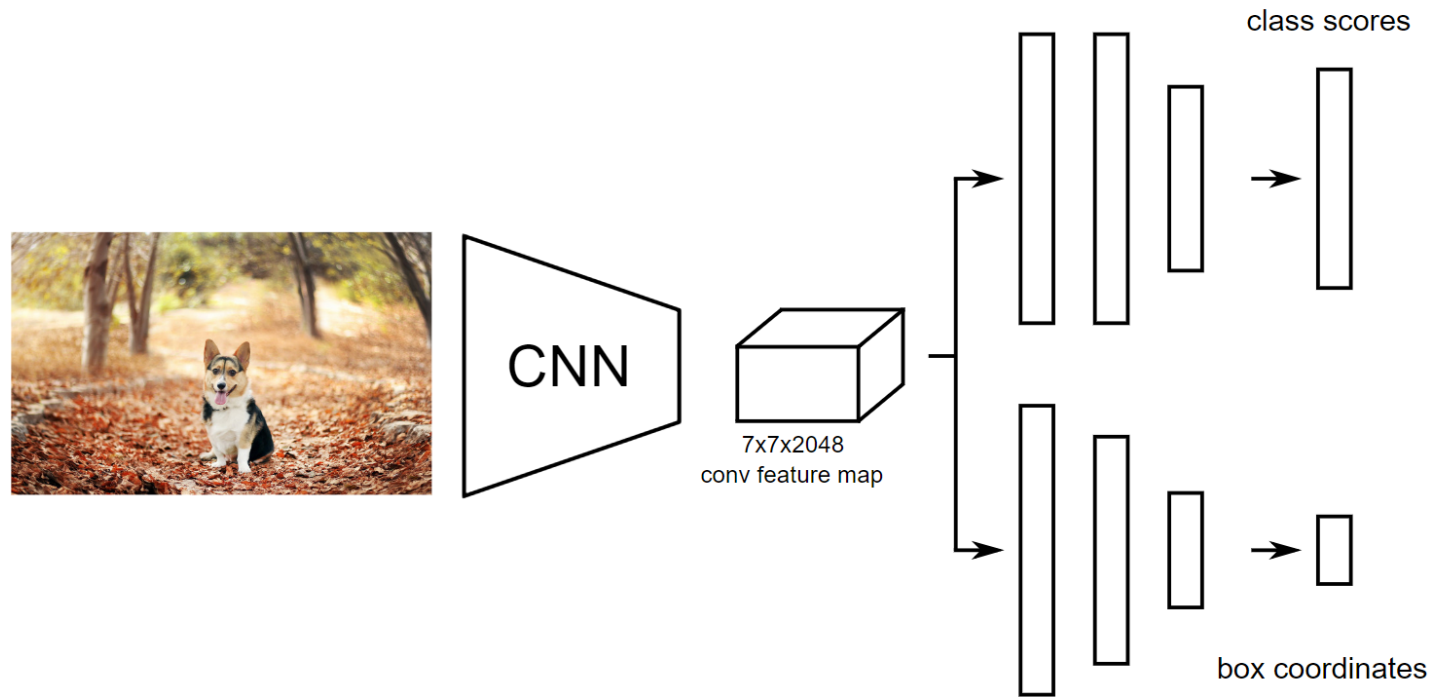


prediction

CNN

120.4
240.6
46.4
51.1

# Localization as Regression

# Classification + Localization

- Use a pre-trained CNN on ImageNet (ex. ResNet)
- The "localization head" is trained separately with regression
- Possible end-to-end fine-tuning of both tasks

# Object Detection

- We don't know in advance the number of objects in the image.

- Object detection relies on object proposal and object classification
  - Object proposal: find regions of interest (ROIs) in the image
  - Object classification: classify the object in these regions

- Two main families:
  - A grid in the image where each cell is a proposal (SSD, YOLO)
  - Region proposal (SPP, MultiBox, Faster RCNN, ⋯)

# Object Detection with R-CNNs

- R-CNN:
  - Find regions that we think have objects.
  - Use CNN to classify



1. Input image  2. Extract region proposals (~2k)  3. Compute CNN features  4. Classify regions
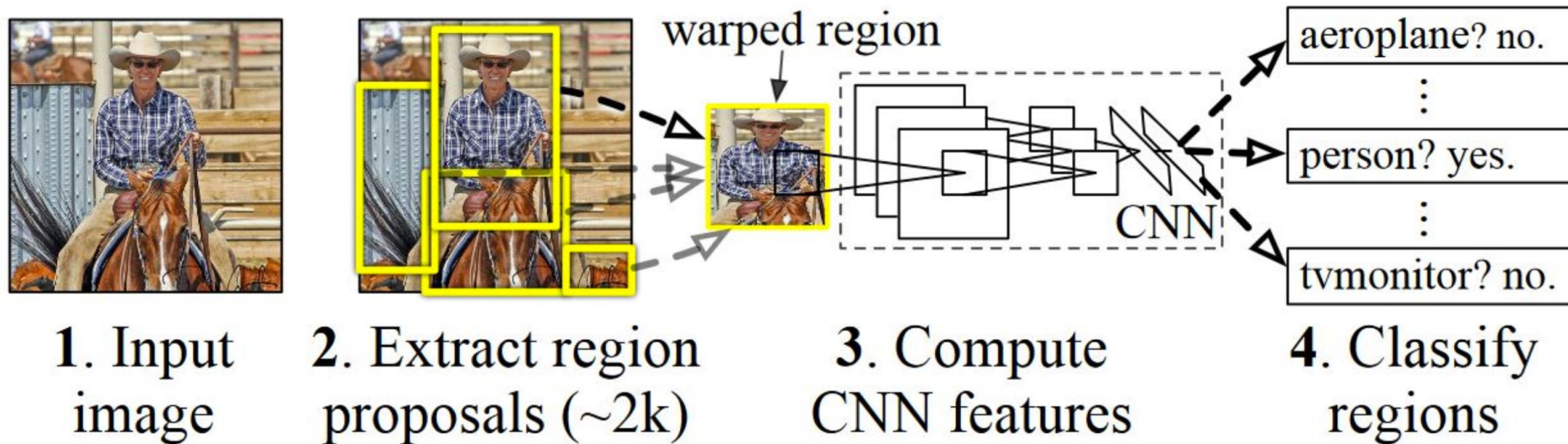
# Segmentation

- Output a class map for each pixel (here: dog vs background)



- Instance segmentation: specify each object instance as well (two dogs have different instances)
- This can be done through object detection + segmentation

# Semantic Segmentation: FCNs

- FCN: Fully Convolutional Network
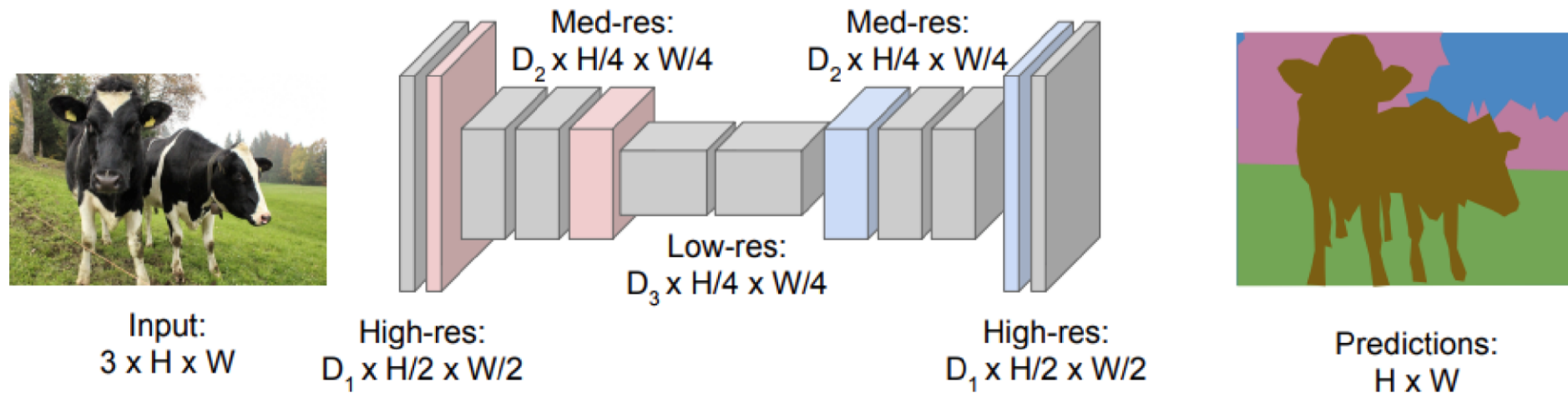  - Network designed with all convolutional layers, with down-sampling and up-sampling operations



Input: 3 x H x W

High-res: $D_1$ x H/2 x W/2

Med-res: $D_2$ x H/4 x W/4

Low-res: $D_3$ x H/4 x W/4
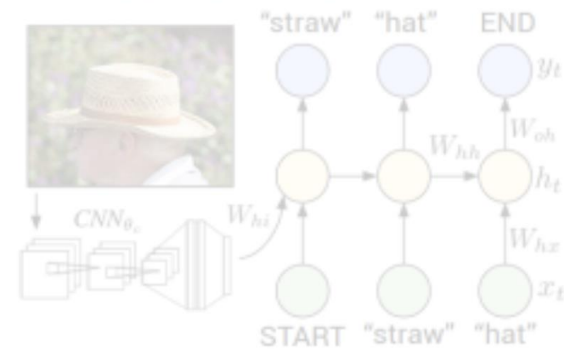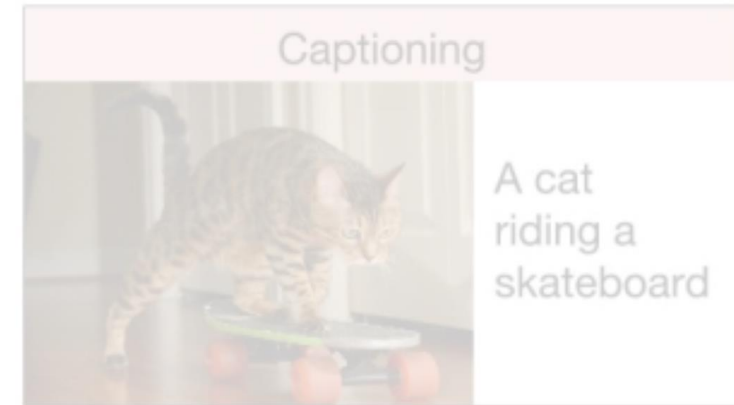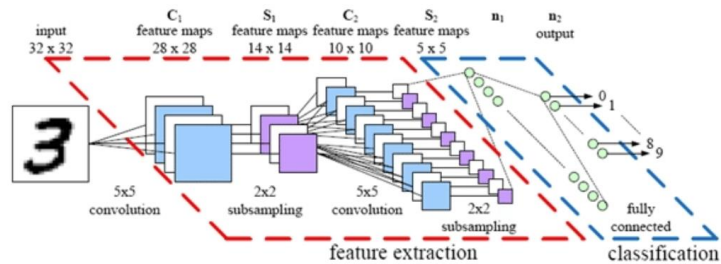
Med-res: $D_2$ x H/4 x W/4

High-res: $D_1$ x H/2 x W/2

Predictions: H x W

# Image Captioning using RNNs

# Image Captioning using RNNs