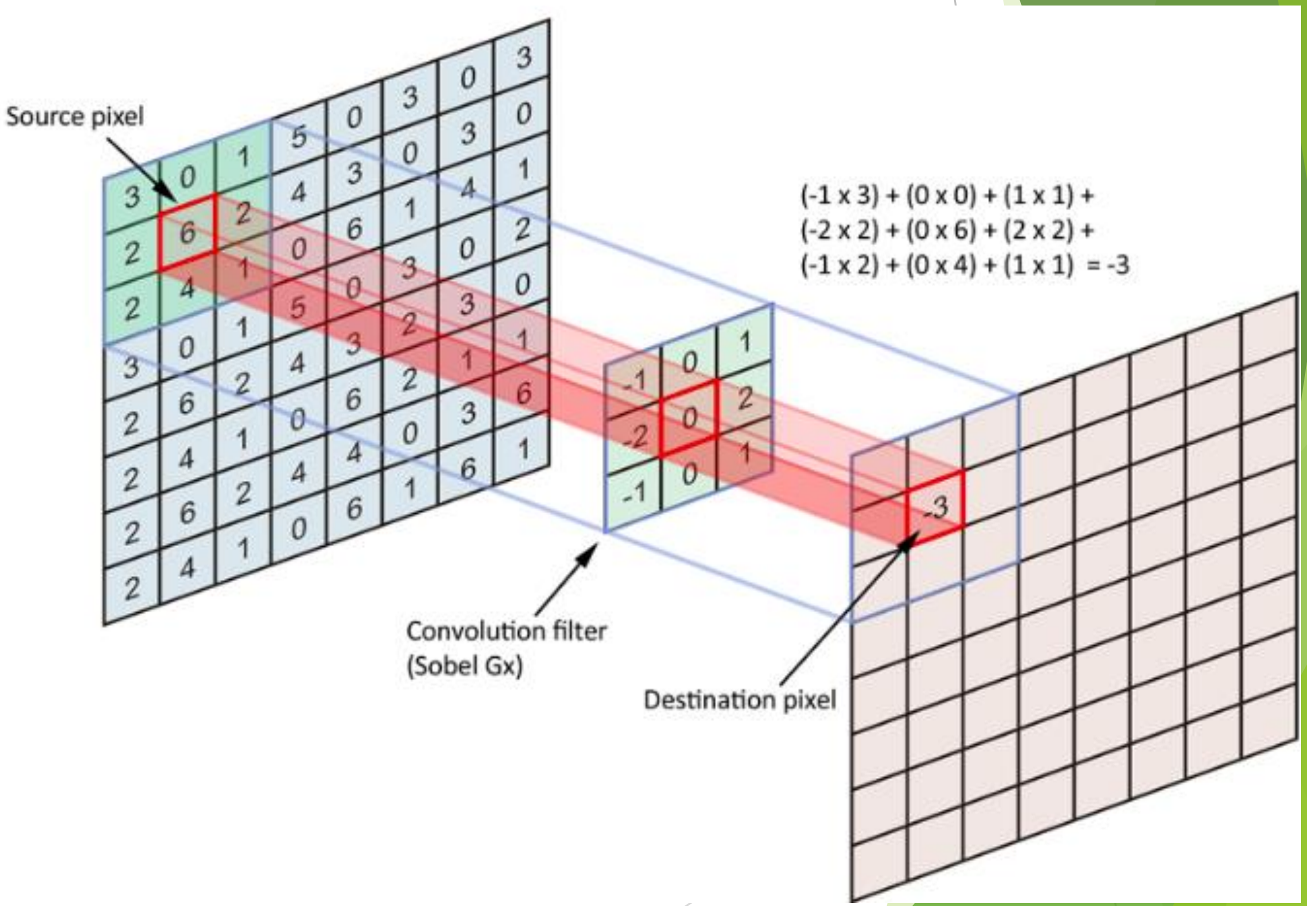
Convolutional Neural Networks

BITS F312: Neural Networks and Fuzzy Logic

Lab 06

Convolution Layer

- Element-wise multiply the pixel values in the matrix with the values in the filter and add all of them.
- Then move filter by "stride" number of steps each time to the right until you reach the end, then move down.
- Repeat for all channels and apply a non-linearity function (e.g. ReLU).

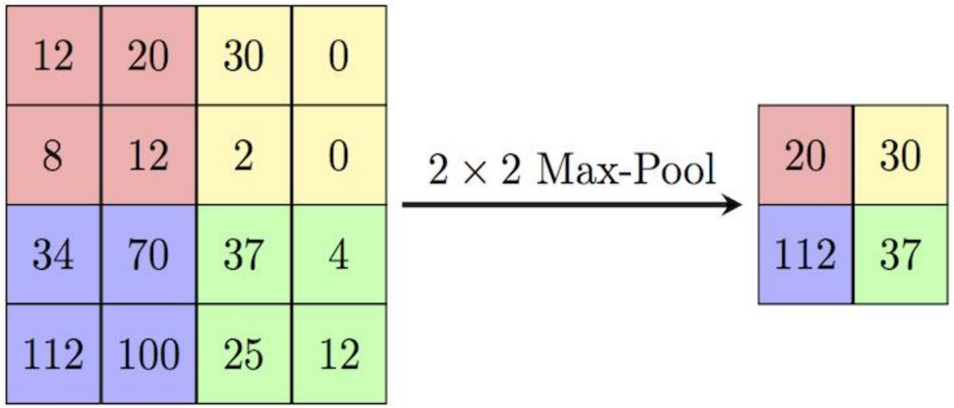


Convolution Layer: Properties

- Locality: Each neuron is related to only a few other neurons.
- Translational invariance: If a pattern (e.g. a cat) moves in the image, the ConvNet will still detect that pattern.
- Local Stationarity: Similar patches are shared across data domains, that is, always check for a repeating pattern and never for an object.
- Multi-scale: Simple structures combine to compose slightly more abstract structures and so on.

Max Pooling Layer

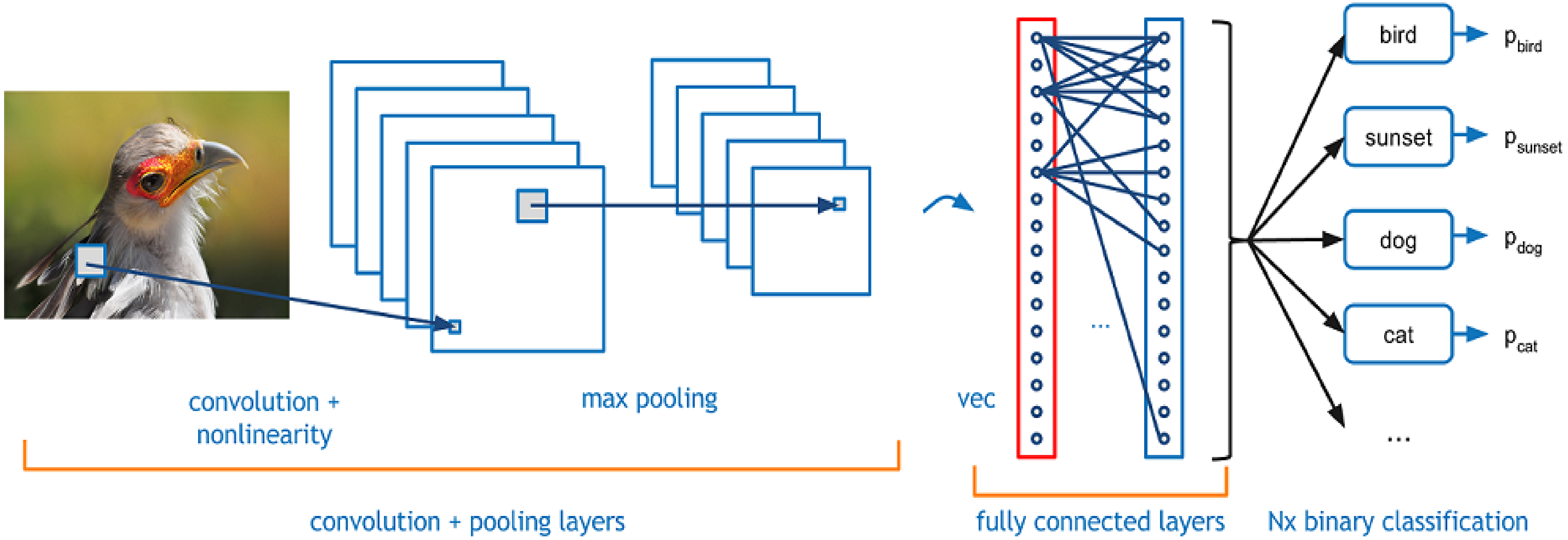
- Take a window of some size and move it like a filter in a Convolutional layer, and at each instance choose the pixel in the window with the maximum value.
- Typical values:
  - Filter size = (2, 2)
  - Stride = 2



General Architecture

- Repeating blocks, where each block consists of one or several Convolution Layers followed by a Max Pooling layer.
- This is followed by a series of Fully Connected (Dense) layers.
- Regularization is achieved by Dropout, Data Augmentation, L1 and L2 loss.

General Architecture



Problems that CNNs can solve

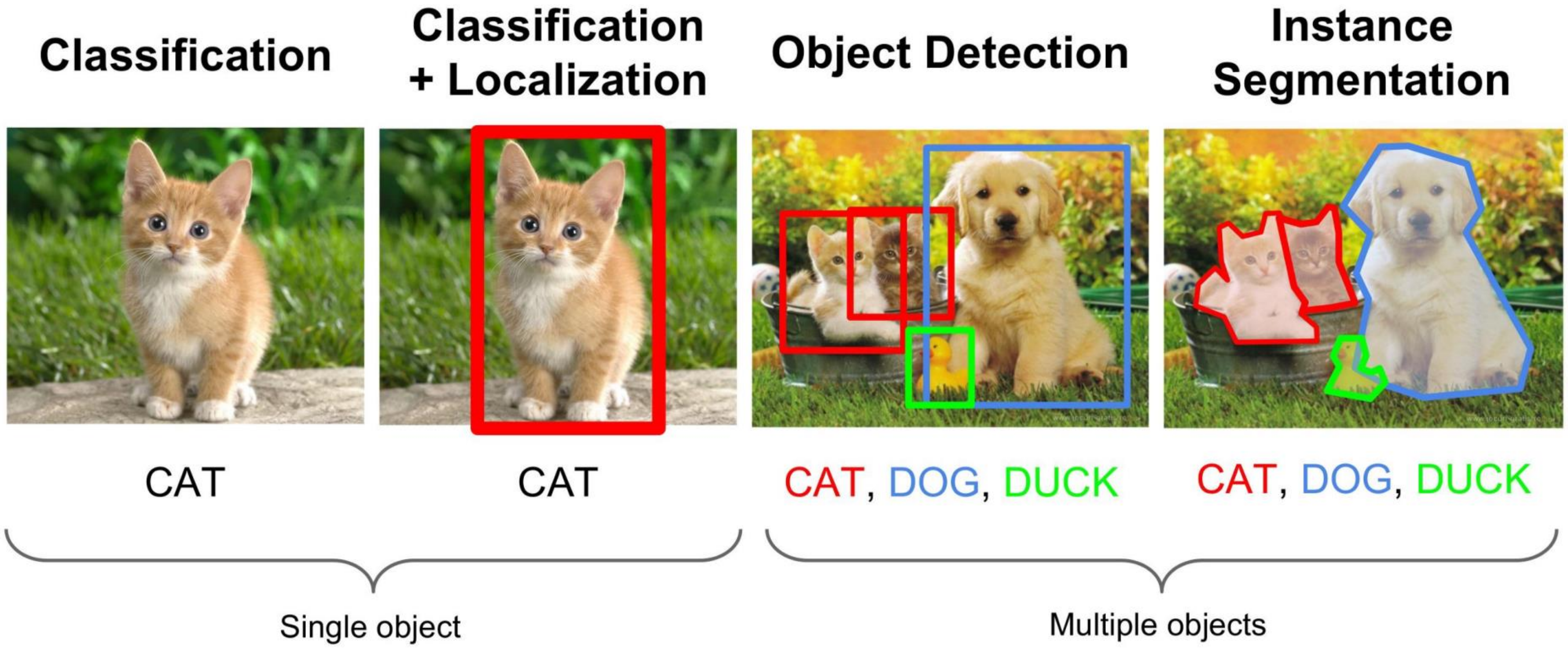
- Computer Vision
  - Face recognition
  - Scene labelling
  - Image classification
  - Action recognition
  - Pose estimation
  - Document analysis (OCR)
  - Neural style transfer
  - Object detection
- Natural Language Processing
  - Speech recognition
  - Text classification

And more...

Object Detection, Localization and Segmentation

- Object detection: A set of objects is given. Predict if any of these objects are present in the image and if yes, then which one?
- Object Localization: Draw a bounding box around the object if it is present along with object detection.
- Image Segmentation: Assign a label to each pixel in the image, that is, draw exact boundaries around all objects in image.

Object Detection, Localization and Segmentation



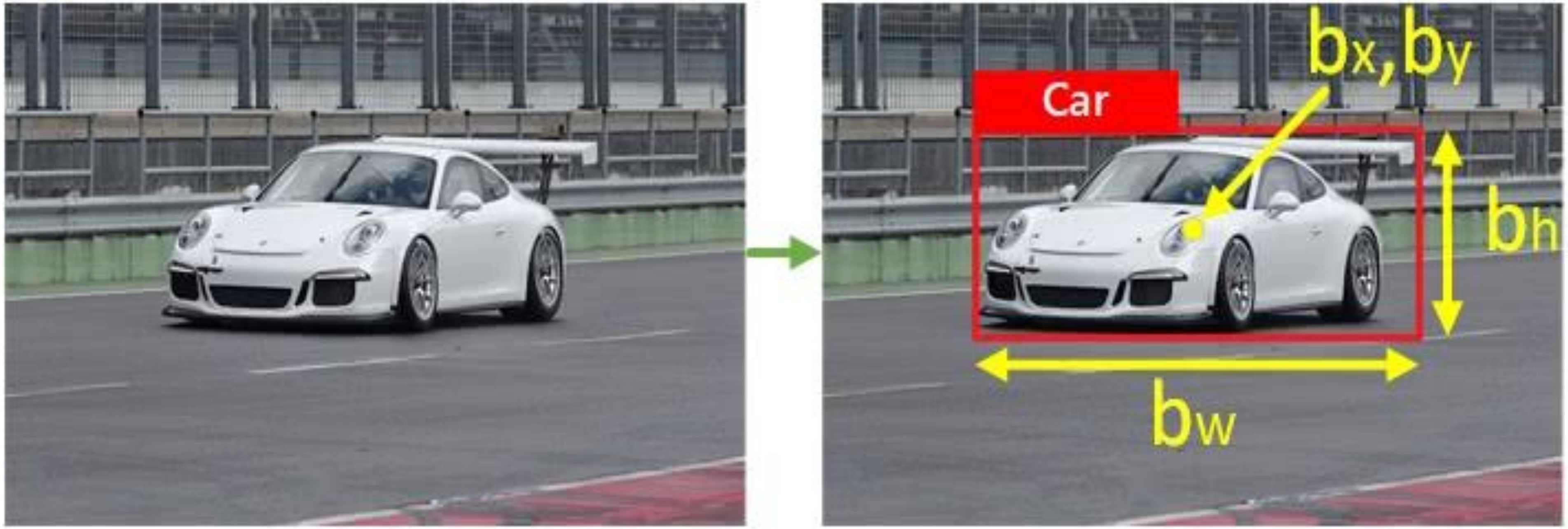
Object Localization

- Solution: Regression
  - P(object being present)
  - X coordinate of center
  - Y coordinate of center
  - Height of bounding box
  - Width of bounding box
  - Label for class 1
  - Label for class 2
  - Label for class 3

$$Y = \begin{bmatrix} p_c \\ b_x \\ b_y \\ b_h \\ b_w \\ c_1 \\ c_2 \\ c_3 \end{bmatrix}$$

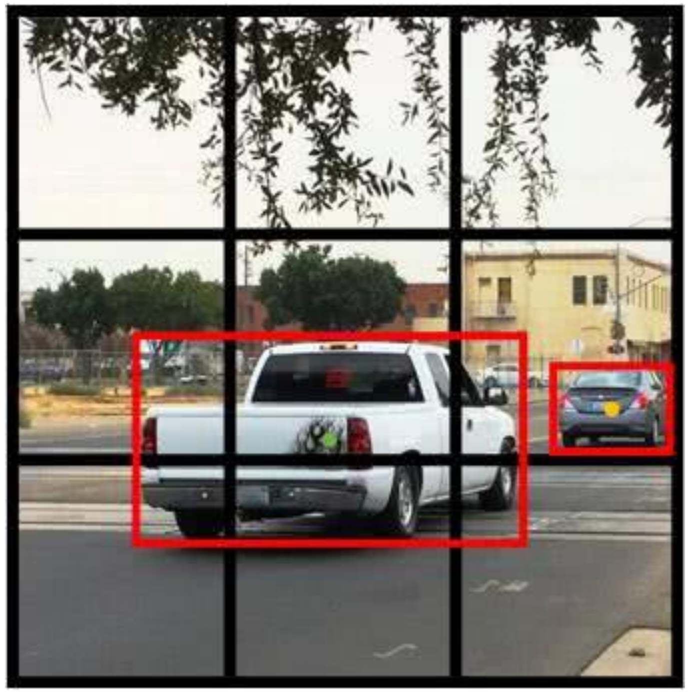
- Class labels are one-hot encoded, and loss used can be MSE for the first 5 labels and cross-entropy for the last 3.

Object Localization

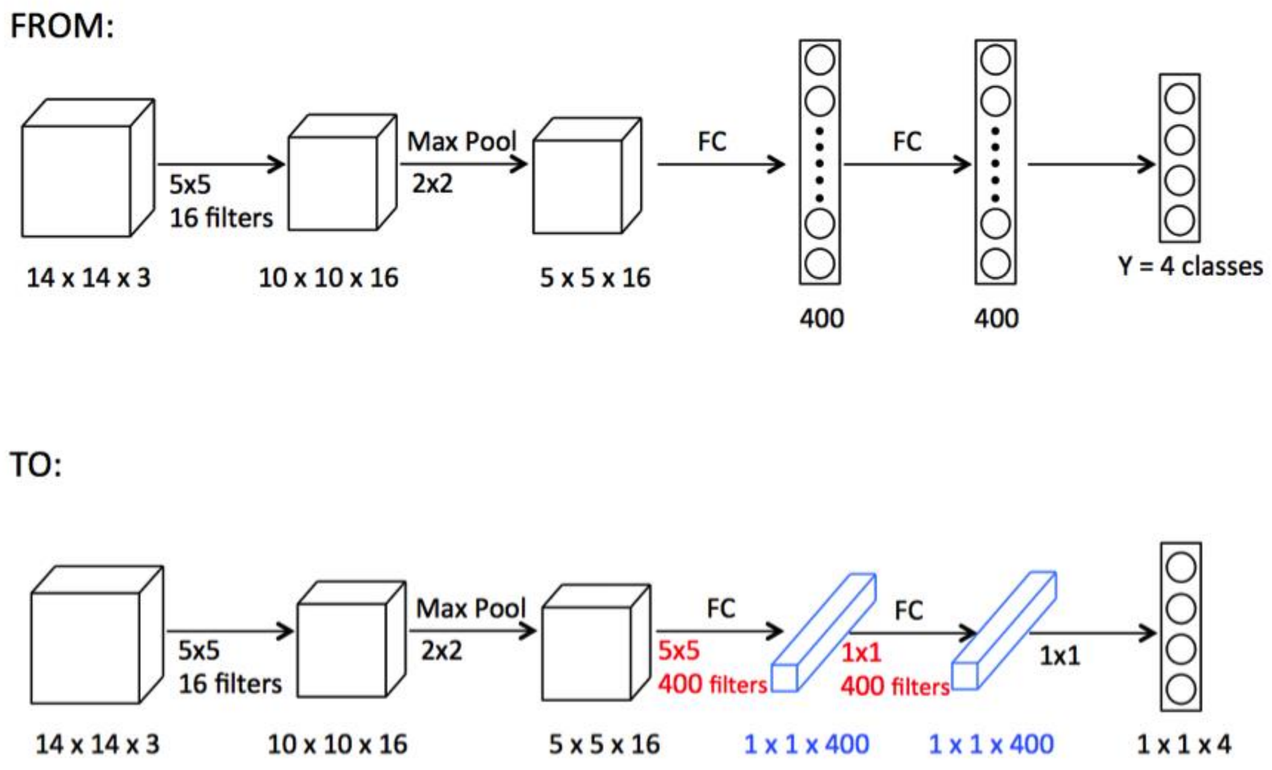


Object Localization

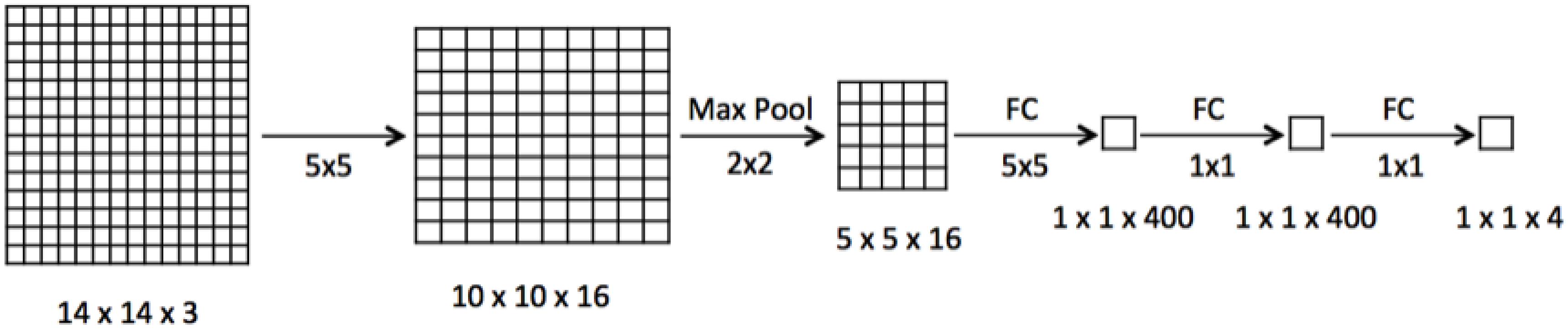
- Multiple objects?
- Sliding windows
- Problems:
  - Time complexity
  - Need to run the algorithm many times

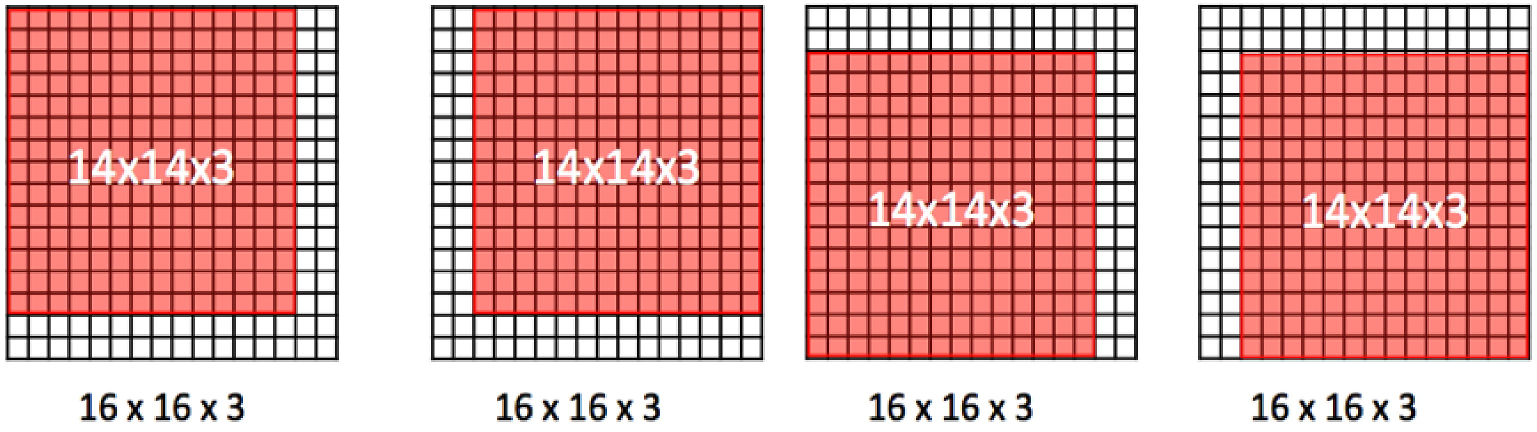


Convolutional Implementation of Sliding Windows

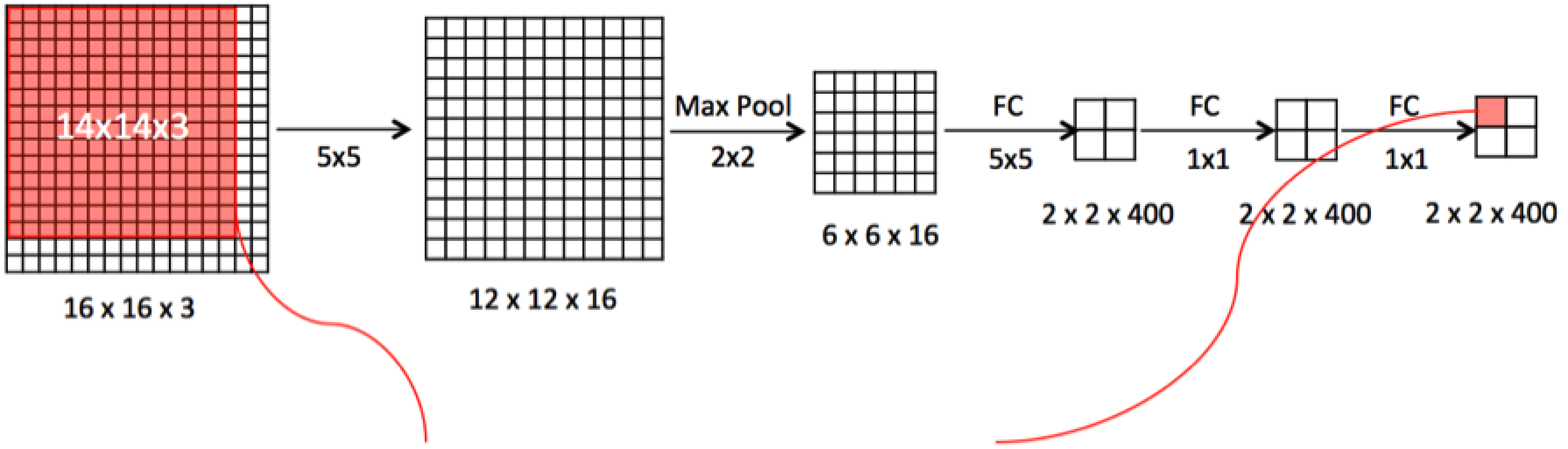


Convolutional Implementation of Sliding Windows





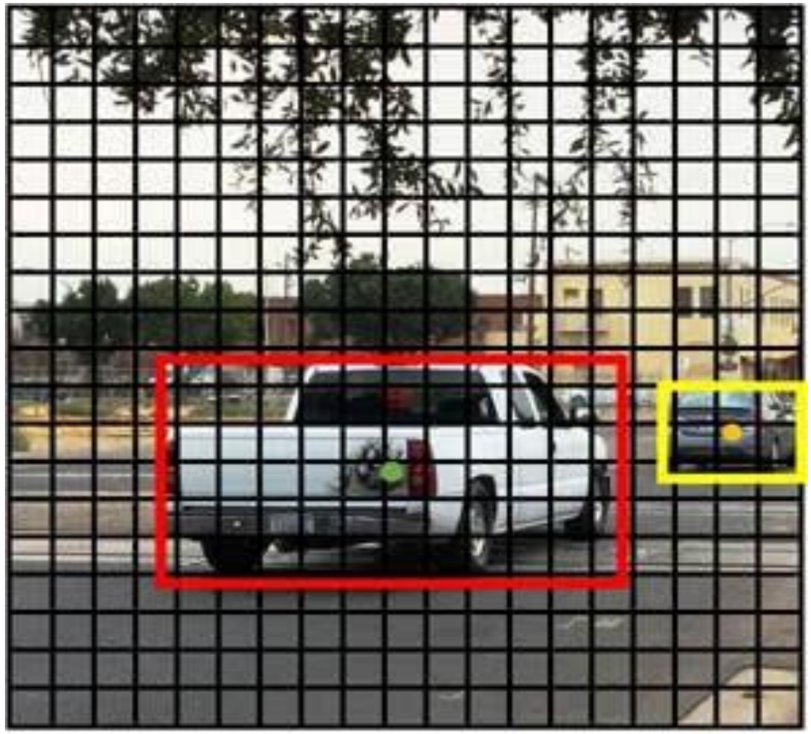
Convolutional Implementation of Sliding Windows



Result of running ConvNet in the upper left corner with
a 14x14x3 region in the original image

You Only Look Once (YOLO)

- Break image into a grid (e.g. (19 x 19) grid).



19 x 19

YOLO: Label Generation

- For each cell where the center of an object lies, the label will be:

$[1, b_x, b_y, b_h, b_w, c_1, c_2, c_3]$.

- For each cell where the centre of an object does not lie, the label will be:

[0, ?, ?, ?, ?, ?, ?, ?]

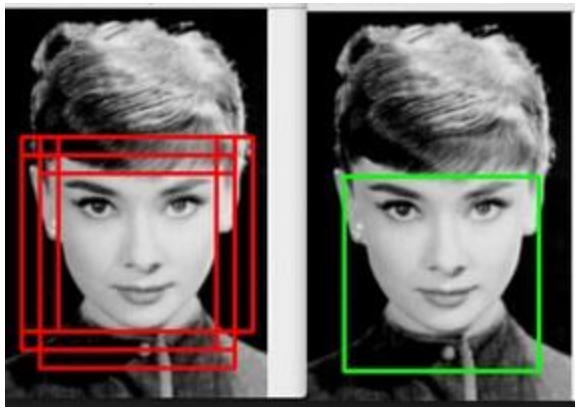
- Labels are combined into a tensor of shape (19, 19, 8).
- Note that b_x, b_y, b_h and b_w are specified relative to the cell boundaries.

YOLO: Intersection over Union (IoU)

- IoU = area of intersection/area of union
- "Correct" if IoU >= 0.5.
- Accuracy = # of correct samples / # of total samples

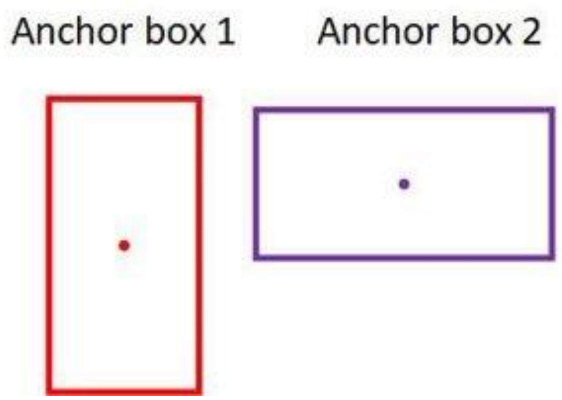
YOLO: Non-Max Suppression

- To make sure that each object is detected only once.
- Algorithm:
  - Discard all boxes with p_c <= 0.6.
  - Check which box has highest value of p_c. Let this be B.
  - Check which boxes have high IoU with B (>= 0.5) and remove them.
  - Add B to solution and remove it from consideration and repeat from Step 2.



YOLO: Anchor Boxes

- If you have multiple objects in same cell, anchor boxes are the solution.
- Also helps in better convergence.
- Different shapes of anchor boxes – change labels accordingly.



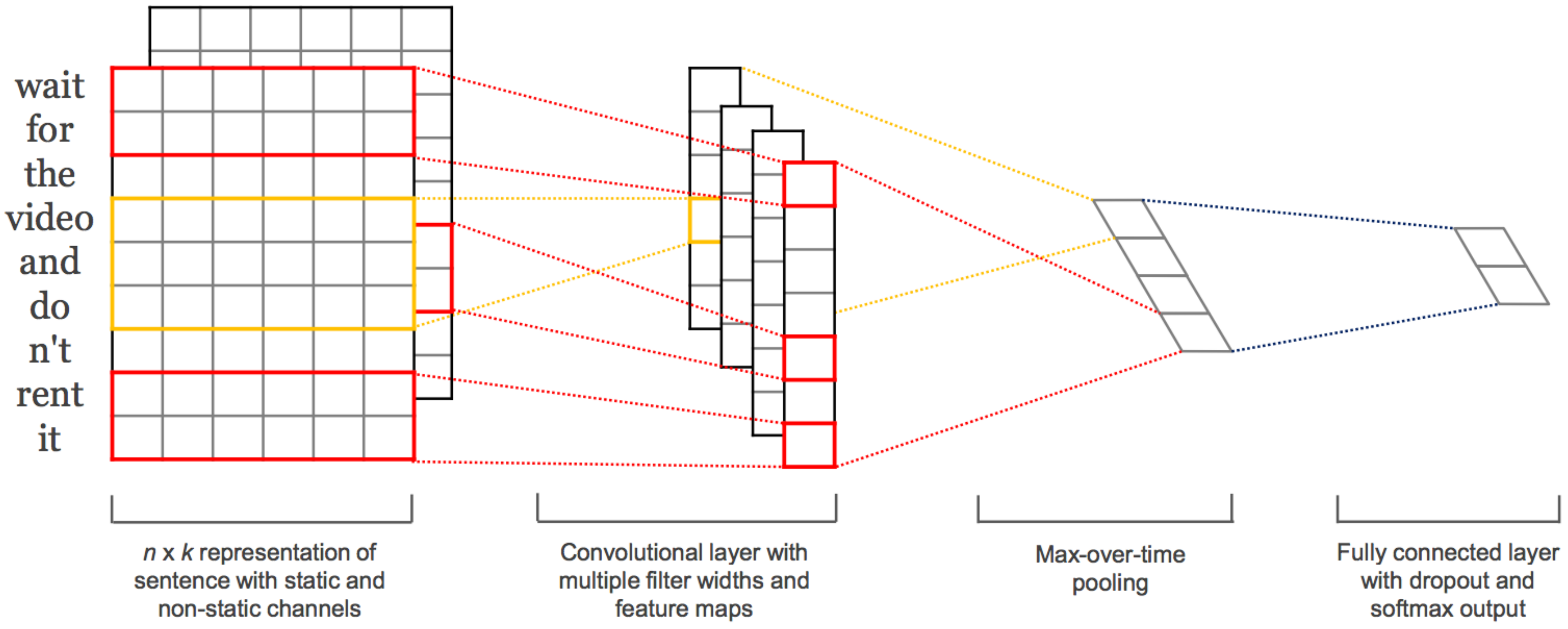
YOLO: Putting it together

- Step 1: Break image into cells.
- Step 2: Decide on anchor boxes.
- Step 3: Generate labels according to cells and anchor boxes.
- Step 4: Run Convnet to get output tensor.
- Step 5: Remove redundant boxes using Non-max suppression.
- Step 6: Predict "correct" or not by checking IoU with label.

Text Classification

- Problem: Classify a document according to its text.
- Solution: CNNs (effective because they capture the salient features only).
- Embeddings: Numerical representations of words (will be explained in later labs).
- 1D CNN: Contains 1D Convolution Layer

Text Classification: CNN Architecture



Face Verification and Recognition

Face Verification

- Input an image of a person and name / ID of that person.
- Predict if the two are the same people.

Face Recognition

- We have a database of K people.
- Input an image and output the ID of the person if that person is in the database or ‘None’ if he/she is not.

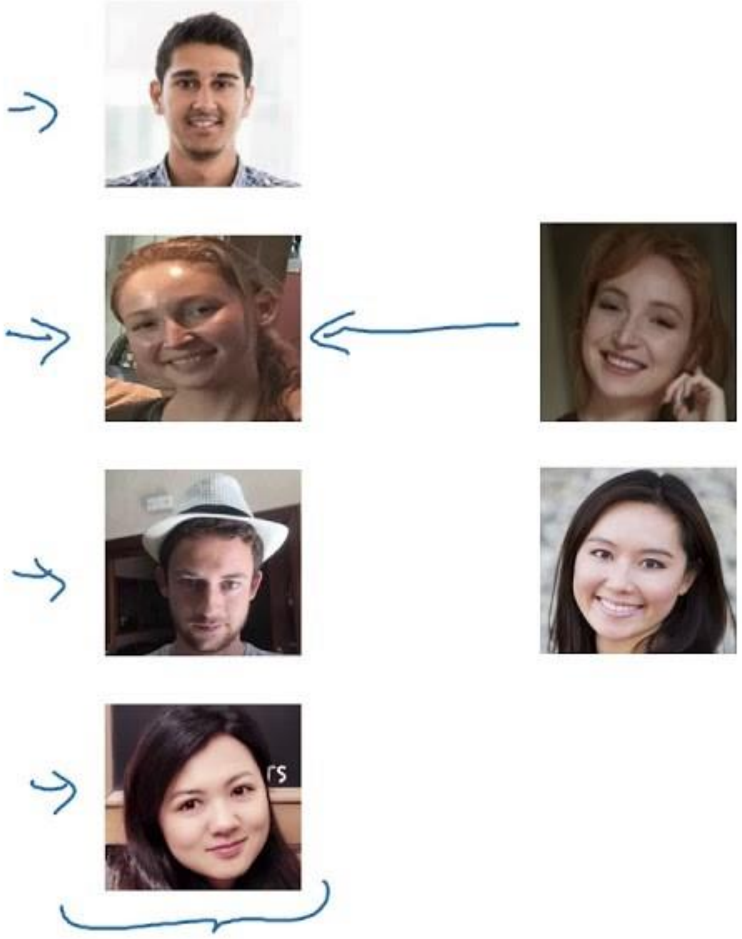
One-Shot Learning

- You only have one example of the face of that person as an example to recognize again.
- Traditional ConvNet will have problems because the output layer will be equal to number of people and the model will become too complex to train.
- Achieved by training a Siamese Network which learns a similarity function instead of a direct mapping.

One-Shot Learning

$dist(x_i, x_j) <= t$: ‘same person’

$dist(x_i, x_j) > t$: ‘different person’



dist(., .) is usually the Euclidean norm

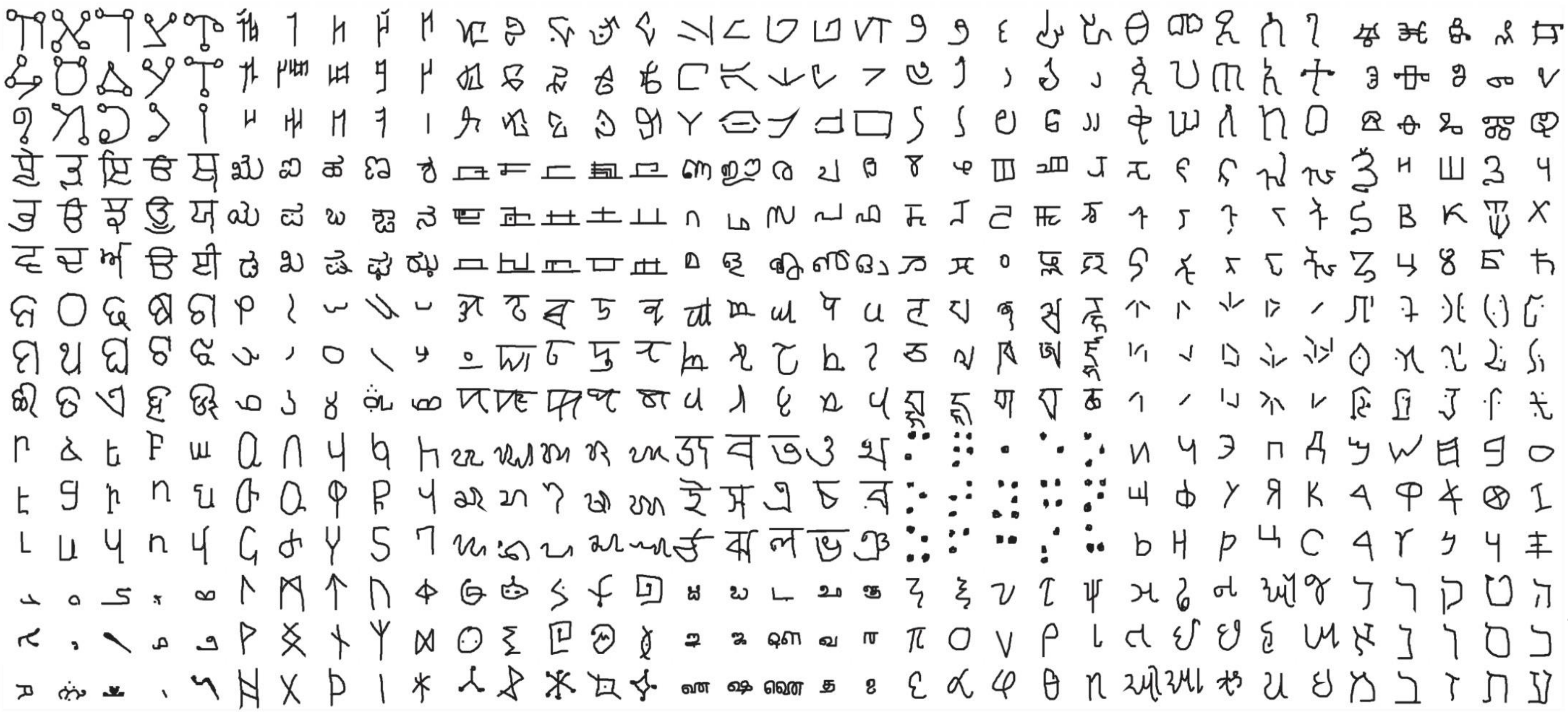
and x_i and x_j are the images.

Siamese Network

- Instead of using the images directly, learn encodings and use them.
- Hence, $dist(x_i, x_j) = (|f(x_i) - f(x_j)|_2)^2$.
- Learn parameters so that this quantity is small if x_i and x_j are the same person and large if they are different people.
- Two ways to train: Triplet loss and binary classification.

Lab Question

The Omniglot Dataset



Lab Question

The Omniglot Dataset

- Developed for ‘human-like’ learning, i.e., how humans learn new concepts from just a few examples.
- Contains 1623 characters, each having 20 samples, across 50 alphabets (30 in training, 20 in testing) for One-Shot learning.