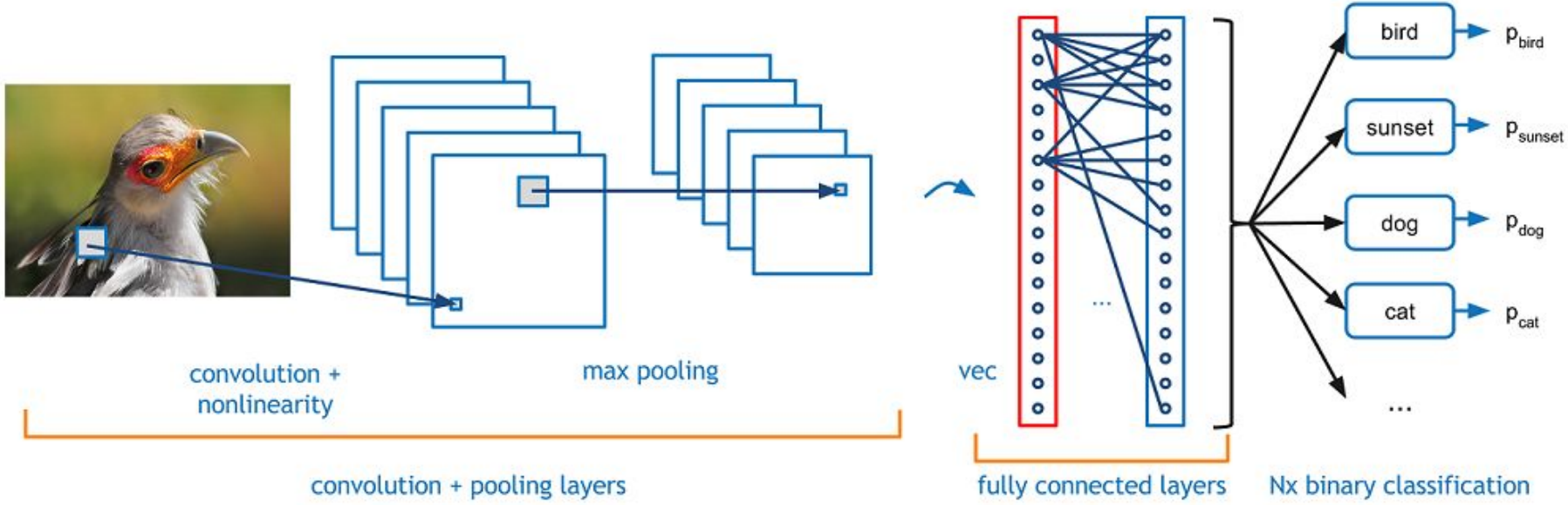
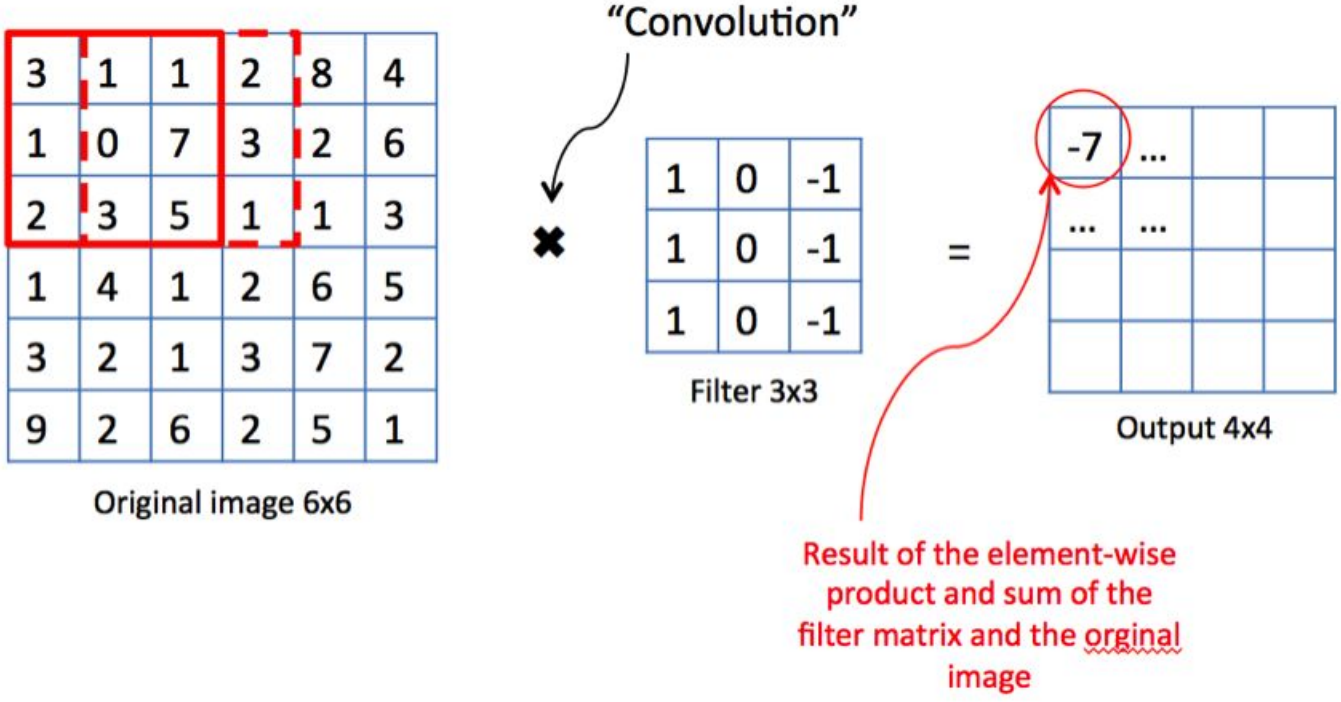
Improve Object Detection Efficiency by Using KCF Tracker

Yiming (Billy) Li

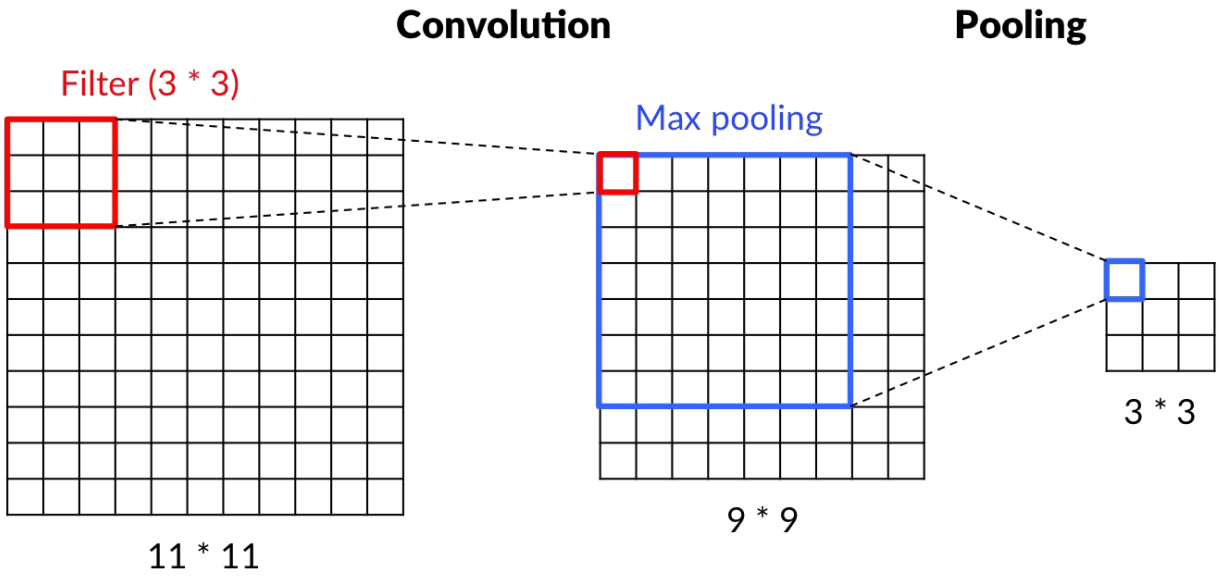
Convolutional Neural Network (CNN)

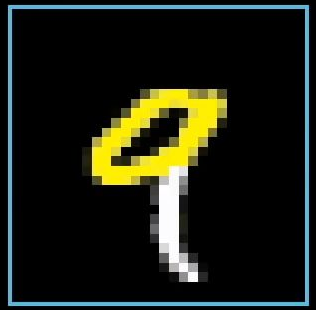


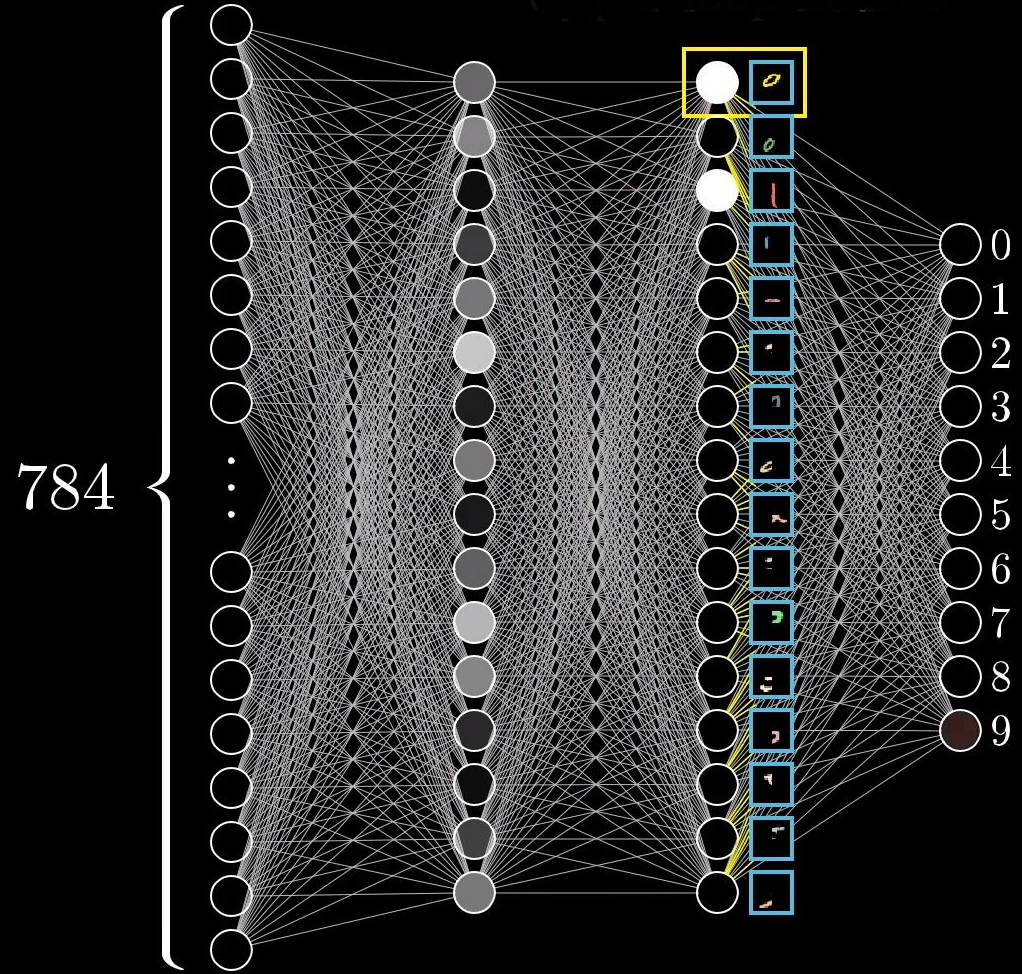
Convolution Layer



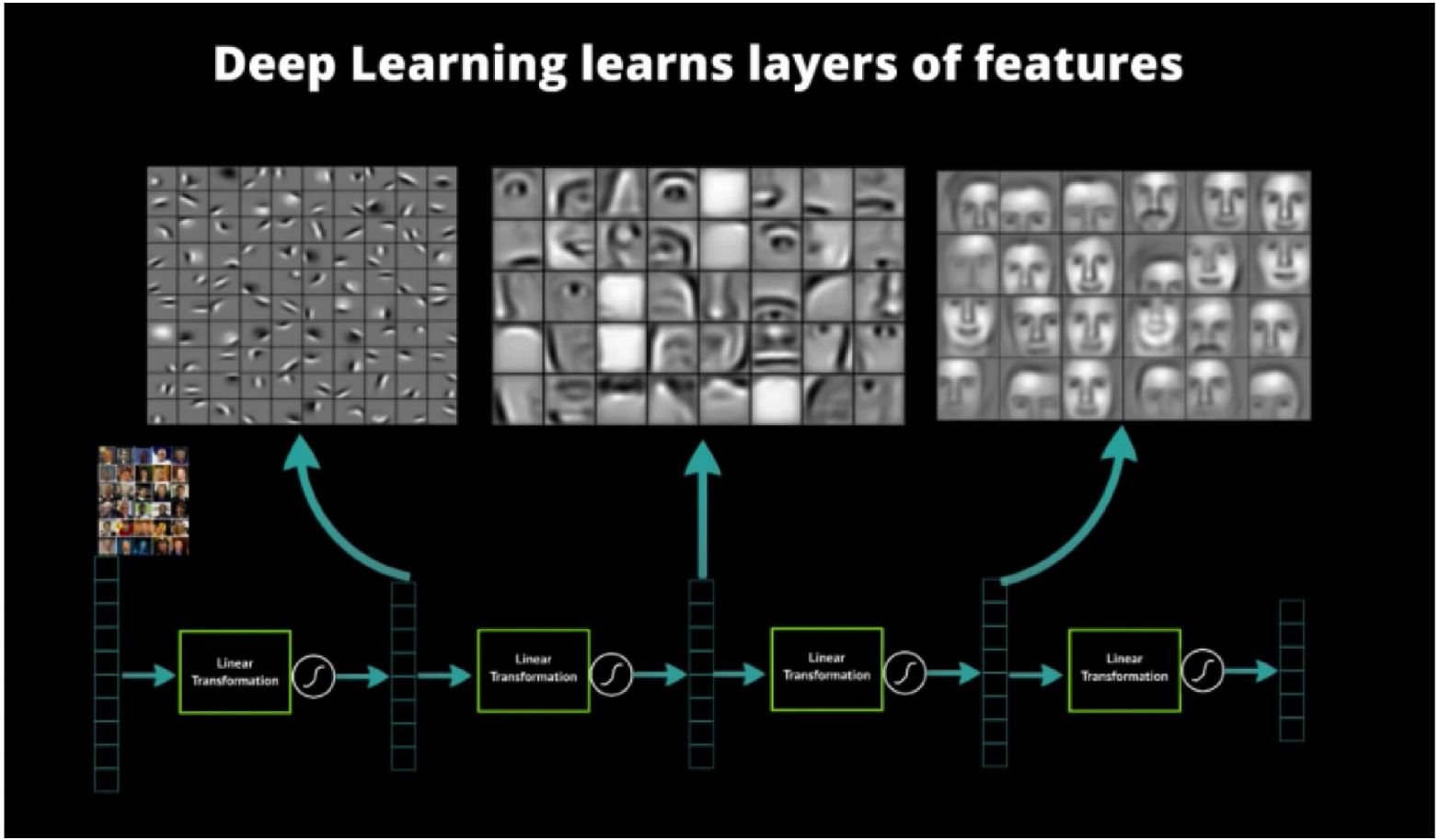
Pooling Layer



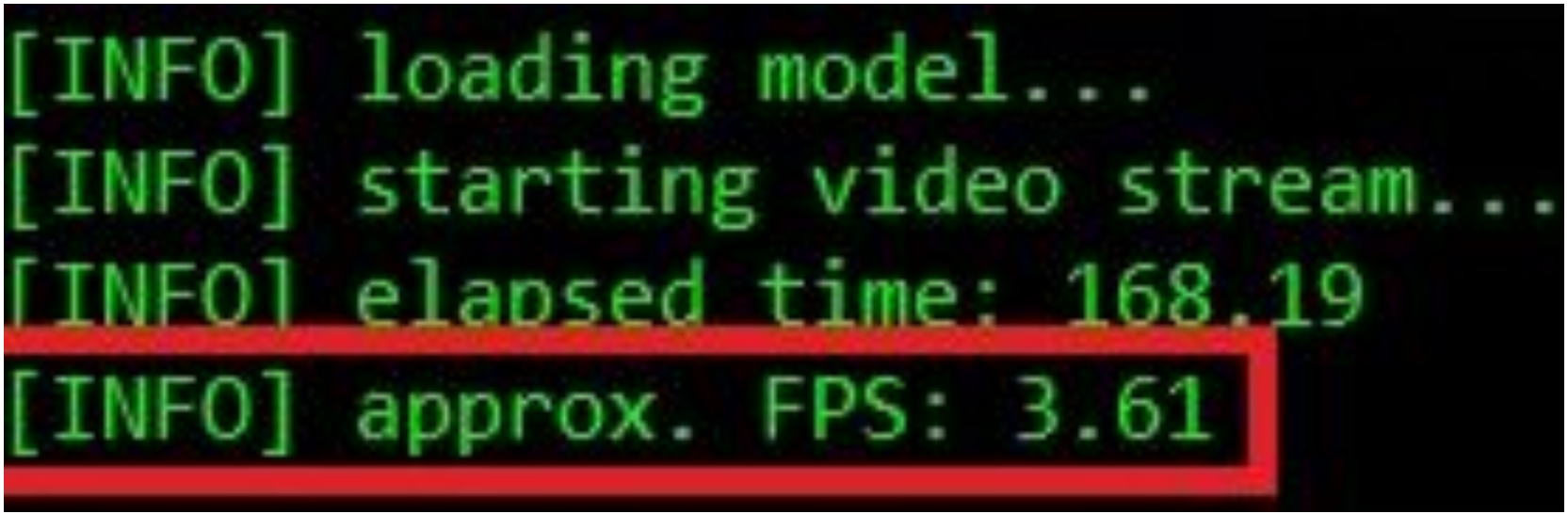



784
0
1
2
3
4
5
6
7
8
9

Deep Learning learns layers of features

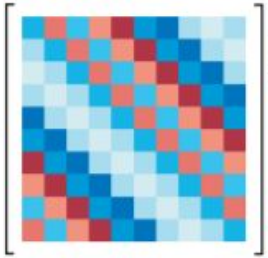


Low Speed


[INFO] loading model...
[INFO] starting video stream...
[INFO] elapsed time: 168.19
[INFO] approx. FPS: 3.61

Model name	Speed (ms)	COCO mAP[^1]
ssd_mobilenet_v1_coco	30	21
ssd_mobilenet_v1_0.75_depth_coco ☆	26	18
ssd_mobilenet_v1_quantized_coco ☆	29	18
ssd_mobilenet_v1_0.75_depth_quantized_coco ☆	29	16
ssd_mobilenet_v1_ppn_coco ☆	26	20
ssd_mobilenet_v1_fpn_coco ☆	56	32
ssd_resnet_50_fpn_coco ☆	76	35
ssd_mobilenet_v2_coco	31	22
ssd_mobilenet_v2_quantized_coco	29	22
ssdlite_mobilenet_v2_coco	27	22
ssd_inception_v2_coco	42	24
faster_rcnn_inception_v2_coco	58	28
faster_rcnn_resnet50_coco	89	30
faster_rcnn_resnet50_lowproposals_coco	64	
rfcn_resnet101_coco	92	30
faster_rcnn_resnet101_coco	106	32

Cyclic shifts

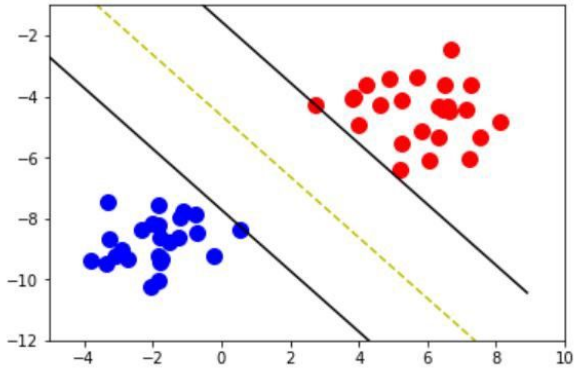
$C\left(\text{[color bar]}\right) =$ 

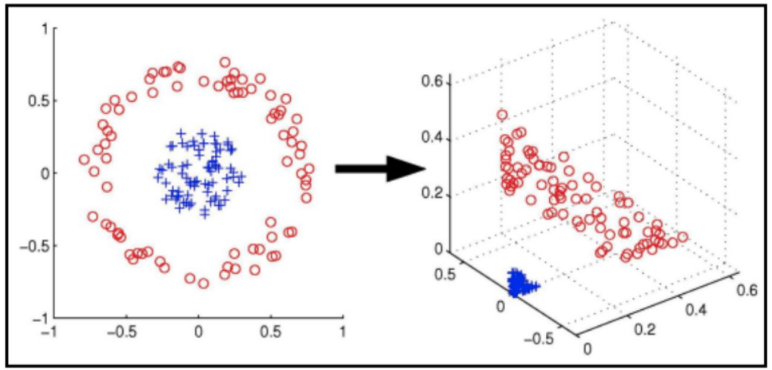
$$X = C(\mathbf{x}) = \begin{bmatrix} x_1 & x_2 & x_3 & \cdots & x_n \\ x_n & x_1 & x_2 & \cdots & x_{n-1} \\ x_{n-1} & x_n & x_1 & \cdots & x_{n-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_2 & x_3 & x_4 & \cdots & x_1 \end{bmatrix}.$$

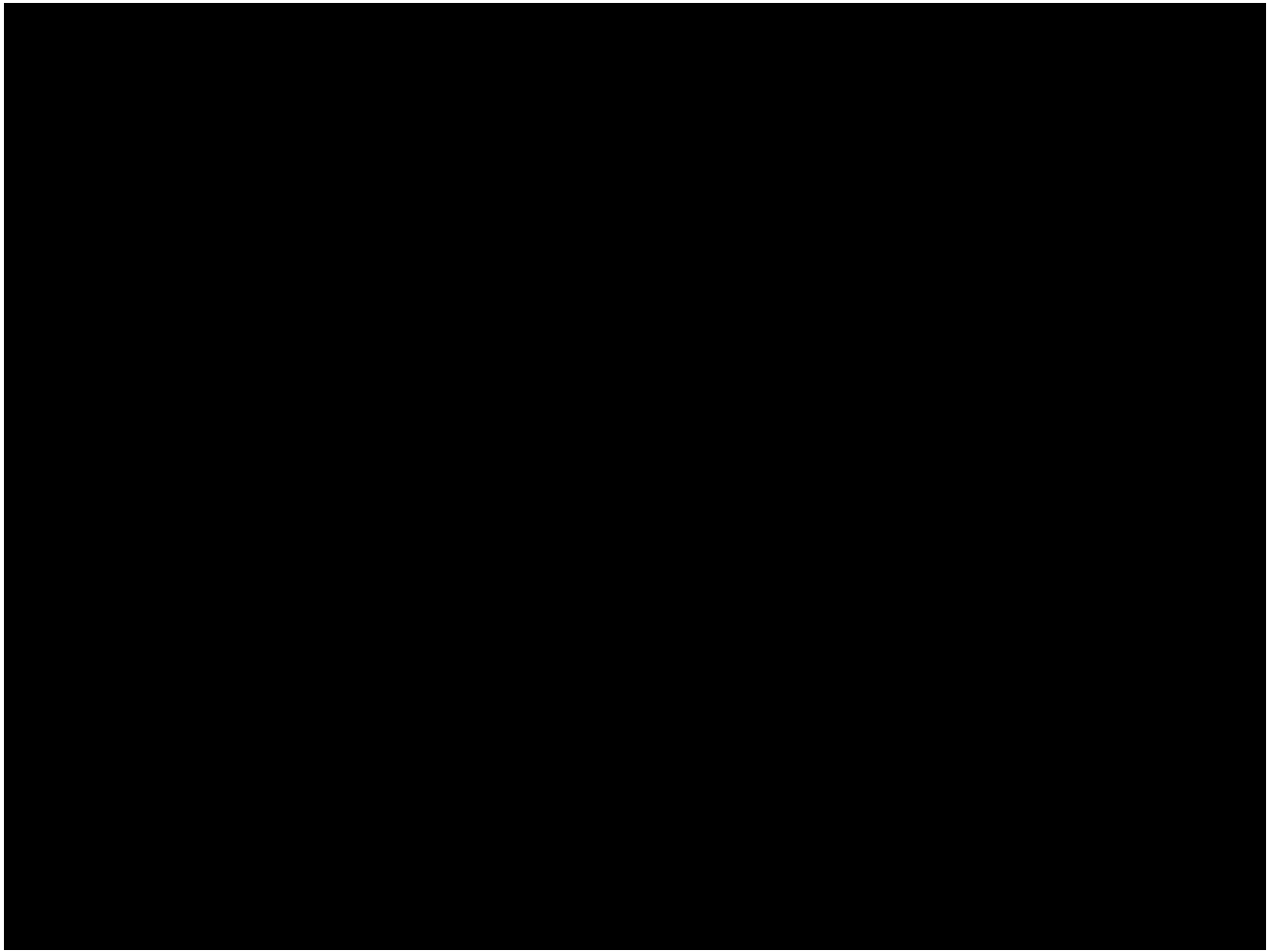
+30 +15 Base sample -15 -30

$$X = F\,\mathrm{diag}\left(\hat{\mathbf{x}}\right)\,F^H$$

Support vector machine (SVM)







```
(MaskRCNN) C:\ObjRec2\Tracking\object-tracking-dlib>python track_object.py --prototxt mobilenet_ssd/Mobi
leNetSSD_deploy.prototxt --model mobilenet_ssd/MobileNetSSD_deploy.caffemodel --video input/bottle.mov -
-label bottle --output output/bottle_output.avi
[INFO] loading model...
[INFO] starting video stream...
[INFO] elapsed time: 27.80
[INFO] approx. FPS: 21.87
```

Detection + Tracking

```
(MaskRCNN) C:\ObjRec2\Tracking\object-tracking-dlib>python detect_object.py --prototxt mobilenet_ssd/Mob
ileNetSSD_deploy.prototxt --model mobilenet_ssd/MobileNetSSD_deploy.caffemodel --video input/bottle.mov
--label bottle --output output/bottle_detection_output.avi
[INFO] loading model...
[INFO] starting video stream...
[INFO] elapsed time: 168.19
[INFO] approx. FPS: 3.61
```

Pure Detection

```
############################################ I N T E R F A C E ##########################################################

MODEL_NAME = 'ssd_inception_v2_coco_2018_01_28'
VIDEO_NAME = 'videos/lab_video1.avi'

OUTPUT = 'output/lab_video1_output.avi'

CWD_PATH = os.getcwd()

PATH_TO_CKPT = os.path.join(CWD_PATH,MODEL_NAME,'frozen_inference_graph.pb')

PATH_TO_LABELS = os.path.join(CWD_PATH,'data','mscoco_label_map.pbtxt')

PATH_TO_VIDEO = os.path.join(CWD_PATH,VIDEO_NAME)

NUM_CLASSES = 90

skip_frames = 10
min_score_thresh = 0.5
max_boxes_to_draw = 20

#####################################################################################################################
```

```
label_map = label_map_util.load_labelmap(PATH_TO_LABELS)
categories = label_map_util.convert_label_map_to_categories(label_map, max_num_classes=NUM_CLASSES, use_display_name=True)
category_index = label_map_util.create_category_index(categories)

detection_graph = tf.Graph()
with detection_graph.as_default():
    od_graph_def = tf.GraphDef()
    with tf.gfile.GFile(PATH_TO_CKPT, 'rb') as fid:
        serialized_graph = fid.read()
        od_graph_def.ParseFromString(serialized_graph)
        tf.import_graph_def(od_graph_def, name='')

    sess = tf.Session(graph=detection_graph)

image_tensor = detection_graph.get_tensor_by_name('image_tensor:0')

detection_boxes = detection_graph.get_tensor_by_name('detection_boxes:0')

detection_scores = detection_graph.get_tensor_by_name('detection_scores:0')
detection_classes = detection_graph.get_tensor_by_name('detection_classes:0')

num_detections = detection_graph.get_tensor_by_name('num_detections:0')
```

```
vs = cv2.VideoCapture(PATH_TO_VIDEO)

(h, w) = (None, None)

if (vs.isOpened()== False):
    print("Error opening video stream or file")

writer = None

trackers = []
labels = []

totalFrames = 0

fps = FPS().start()
```

```
while True:
    (grabbed, frame) = vs.read()

    if frame is None:...

    frame = imutils.resize(frame, width=600)
    rgb = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
    frame_expanded = np.expand_dims(frame, axis=0)

    if OUTPUT is not None and writer is None:...

    if totalFrames % skip_frames == 0:...

    elif len(trackers) != 0:...

    if writer is not None:...

    cv2.imshow("Frame", frame)
    key = cv2.waitKey(1) & 0xFF

    if key == ord("q"):
        break

    totalFrames += 1
    fps.update()
```

```
while True:
    (grabbed, frame) = vs.read()

    if frame is None:···

    frame = imutils.resize(frame, width=600)
    rgb = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
    frame_expanded = np.expand_dims(frame, axis=0)

    if OUTPUT is not None and writer is None:···

    if totalFrames % skip_frames == 0:···

    elif len(trackers) != 0:···

    if writer is not None:···

    cv2.imshow("Frame", frame)
    key = cv2.waitKey(1) & 0xFF

    if key == ord("q"):
        break

    totalFrames += 1
    fps.update()
```

Part A

```
if w is None or h is None:
    (h, w) = frame.shape[:2]

(boxes, scores, classes, num) = sess.run(
    [detection_boxes, detection_scores, detection_classes, num_detections],
    feed_dict={image_tensor: frame_expanded})

boxes = np.squeeze(boxes)
classes = np.squeeze(classes).astype(np.int32)
scores = np.squeeze(scores)

trackers = []
```

Part B

```
for i in range(min(max_boxes_to_draw, boxes.shape[0])):
    if scores[i] > min_score_thresh:
        box = tuple(boxes[i].tolist())
        startY = int((box[0] * h))
        startX = int((box[1] * w))
        endY = int((box[2] * h))
        endX = int((box[3] * w))

        t = dlib.correlation_tracker()
        rect = dlib.rectangle(startX, startY, endX, endY)
        t.start_track(rgb, rect)

        if classes[i] in category_index.keys():
            class_name = category_index[classes[i]]['name']
        else:
            class_name = 'N/A'

        label = class_name
        labels.append(label)
        trackers.append(t)

        cv2.rectangle(frame, (startX, startY), (endX, endY), (0, 255, 0), 2)
        cv2.putText(frame, label, (startX, startY - 15),
            cv2.FONT_HERSHEY_SIMPLEX, 0.45, (0, 255, 0), 2)
```

```
while True:
    (grabbed, frame) = vs.read()

    if frame is None:···

    frame = imutils.resize(frame, width=600)
    rgb = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
    frame_expanded = np.expand_dims(frame, axis=0)

    if OUTPUT is not None and writer is None:···

    if totalFrames % skip_frames == 0:···

    elif len(trackers) != 0:···

    if writer is not None:···

    cv2.imshow("Frame", frame)
    key = cv2.waitKey(1) & 0xFF

    if key == ord("q"):
        break

    totalFrames += 1
    fps.update()
```

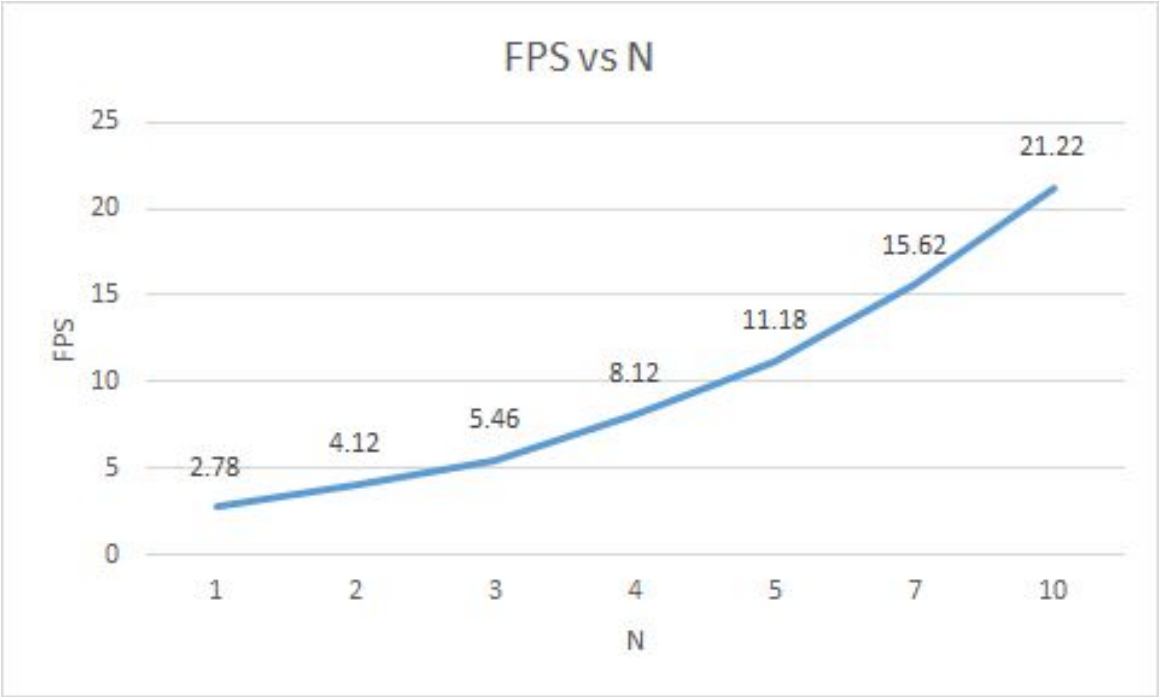
```
for t in trackers:
    t.update(rgb)
    pos = t.get_position()

    startX = int(pos.left())
    startY = int(pos.top())
    endX = int(pos.right())
    endY = int(pos.bottom())

    cv2.rectangle(frame, (startX, startY), (endX, endY),
        (0, 255, 0), 2)

    cv2.putText(frame, label, (startX, startY - 15),
        cv2.FONT_HERSHEY_SIMPLEX, 0.45, (0, 255, 0), 2)
```

```
while True:
    (grabbed, frame) = vs.read()

    if frame is None:···

    frame = imutils.resize(frame, width=600)
    rgb = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
    frame_expanded = np.expand_dims(frame, axis=0)

    if OUTPUT is not None and writer is None:···

    if totalFrames % skip_frames == 0:···

    elif len(trackers) != 0:···

    if writer is not None:···

    cv2.imshow("Frame", frame)
    key = cv2.waitKey(1) & 0xFF

    if key == ord("q"):
        break

    totalFrames += 1
    fps.update()
```

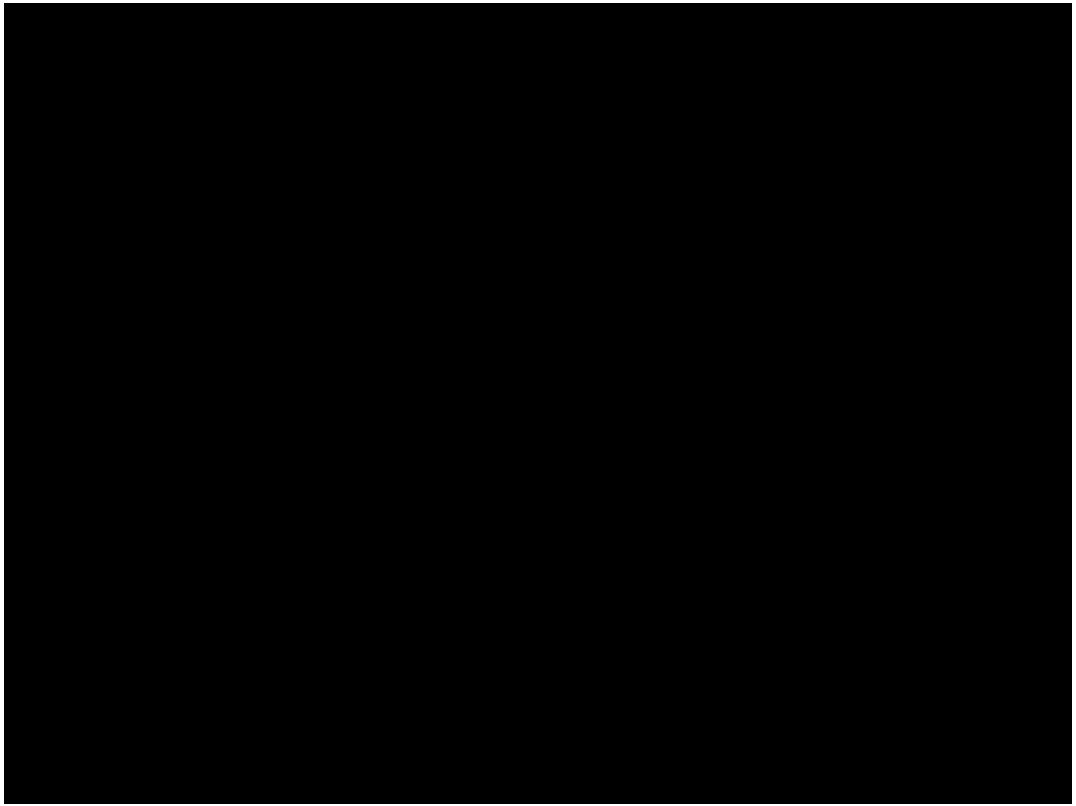

box
box
box

FPS change resulted by running detection every N frames

N	FPS
1	2.78
2	4.12
3	5.46
4	8.12
5	11.18
7	15.62
10	21.22



Drawbacks



References

W. Chen, T. Qu, Y. Zhou, K. Weng, G. Wang and G. Fu, "Door recognition and deep learning algorithm for visual based robot navigation," 2014 IEEE International Conference on Robotics and Biomimetics (ROBIO 2014), Bali, 2014,

João F. Henriques, Rui Caseiro, Pedro Martins, and Jorge Batista. *High-Speed Tracking with Kernelized Correlation Filters.* 5 Nov 2014

Daniel Gordon1 Ali Farhadi1,2 and Dieter Fox1. *Re3 : Real-Time Recurrent Regression Networks for Visual Tracking of Generic Objects.* 26 Feb 2018

Adam Coates, Paul Baumstarck, Quoc Le, and Andrew Y. Ng. *Scalable Learning for Object Detection with GPU Hardware*

Bryan Anenberg, Michela Meister. *Tracking-Learning-Detection*

Archive

https://github.com/BracoLi/Tensorflow-Object-Detection-api-with-dlib-KCF-Tracking

Self Evaluation

i. an ability to apply engineering design to create a product[1] that meets the specified needs of this engineering design experience with consideration of public health, safety, and welfare, as well as global, cultural, social, environmental, and economic factors.
ii. an ability to develop and conduct experimentation, analyze and interpret data, and use engineering judgment to draw conclusions related to the development of the product of this engineering design experience.
iii. an ability to identify, formulate, and solve complex engineering problems arising from this engineering design experience by applying principles of engineering, science, and mathematics.
iv. an ability to function effectively on a team whose members together provide leadership, create a collaborative and inclusive environment, establish goals, plan tasks, and meet objectives associated with this design experience
v. an ability to communicate effectively with a range of audiences appropriate to this design experience in both a written report and oral presentation.
vi. an ability to acquire and apply new knowledge as needed, using appropriate learning strategies to complete the engineering design experience associated with this course.
vii. an ability to recognize ethical and professional responsibilities associated with this engineering design experience and make informed judgments which must consider the impact of the product of this engineering design experience in global, economic, environmental, and societal contexts.