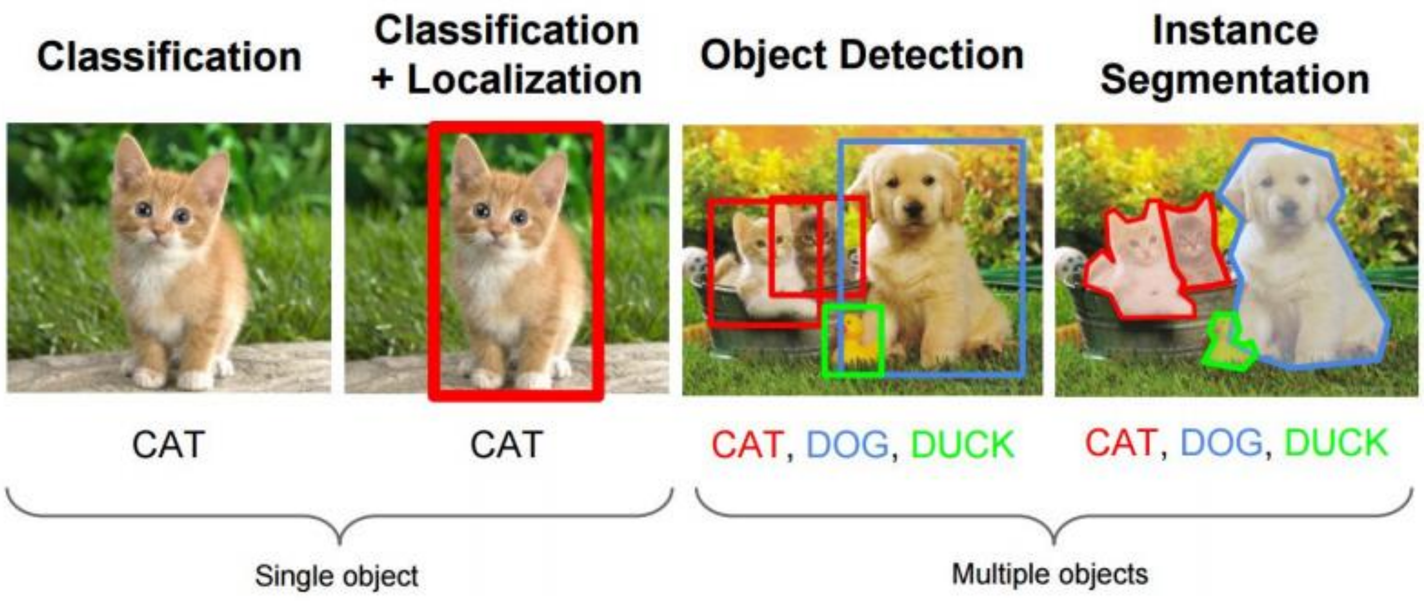
Object Detection

Prof. Seungchul Lee

Industrial AI Lab.

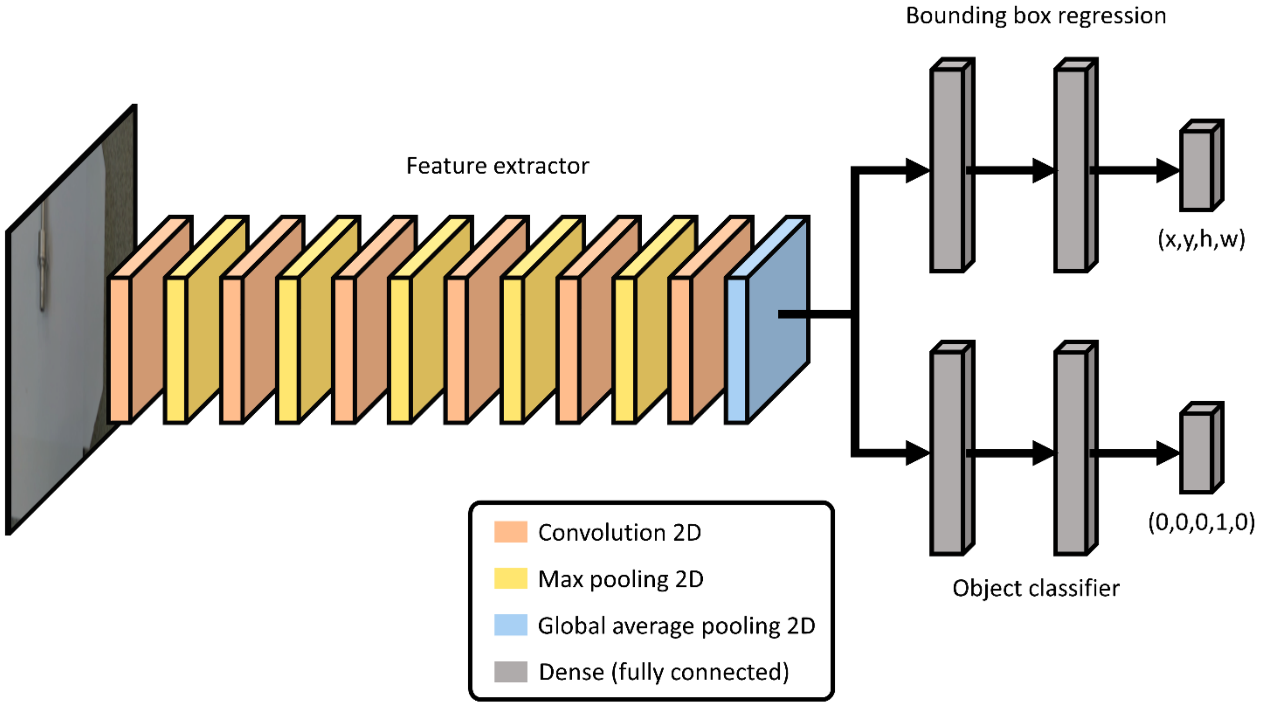
Object Detection

- Classification
  - Predicting the class of one object in an image
- Object localization
  - Identifying the location of one or more objects in an image and drawing a bounding box around their extent
- Object detection
  - Combining these two tasks and localizes and classifies one or more objects in an image



Lab: Object Detection

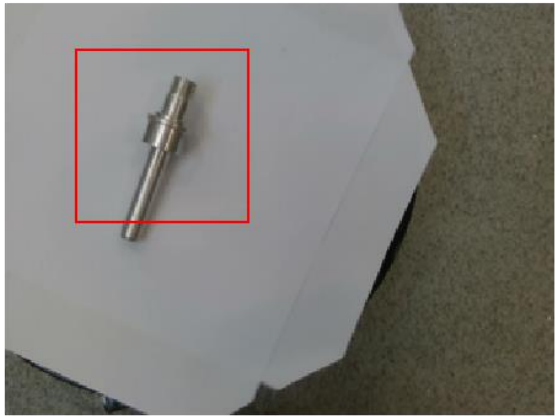
- Simplified version of two-stage object detection model for tutorial
  - 2D convolution layers extract features from the input image
  - Both classifier and bounding box regressor share the same features acquired from the 2D convolution layers
  - Extracted features are utilized for
    - Object classification and
    - Object bounding box detection



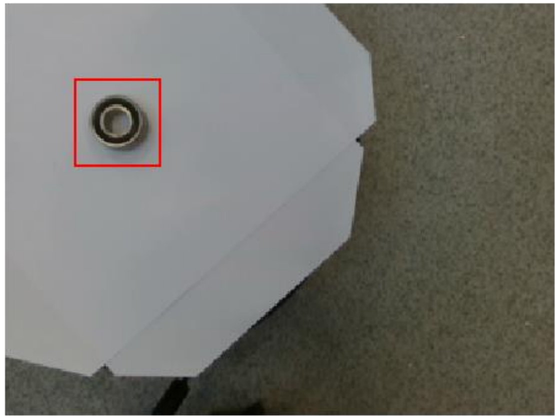
Data and Labels

- Input: 5 classes images (Axis, bearing, bearing box, distance tube, beam)
- Labeled with class and bounding box location (normalized): class, x, y, h, w

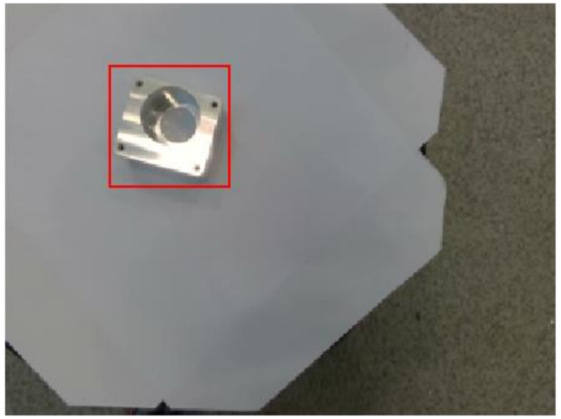
Axis



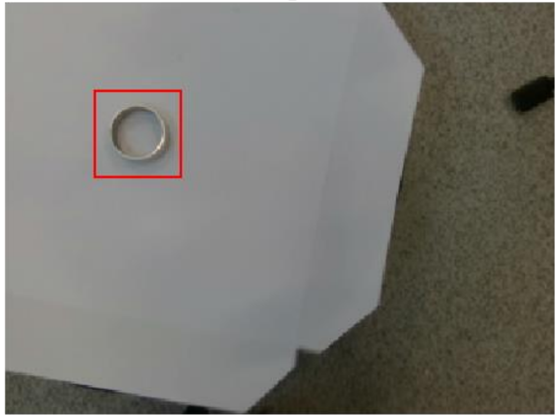
Bearing



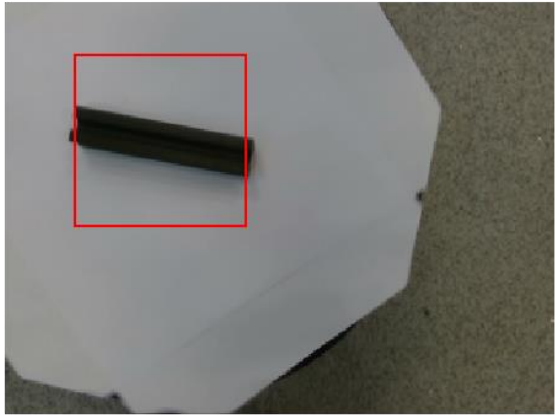
Bearing_Box



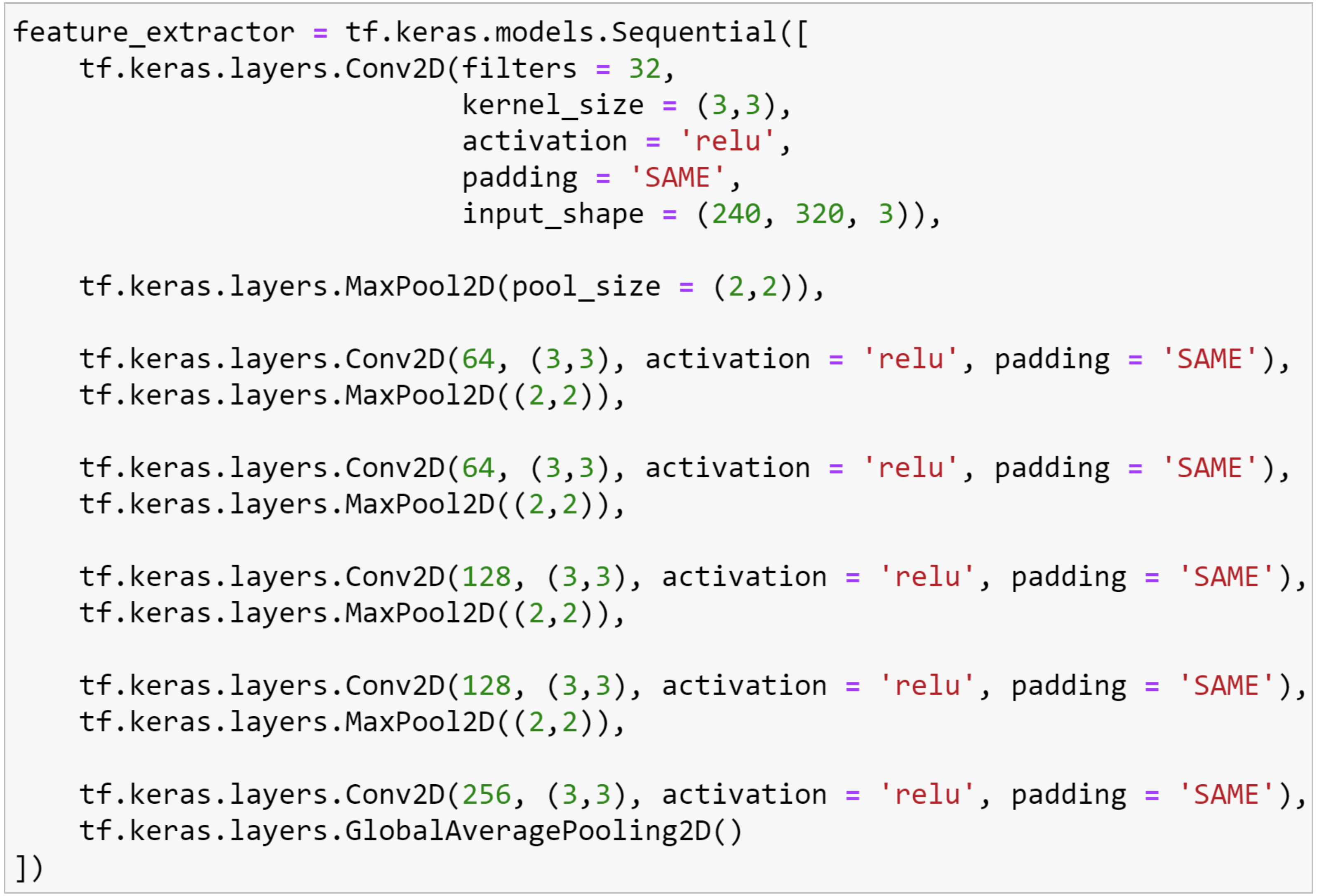
Distance_Tube



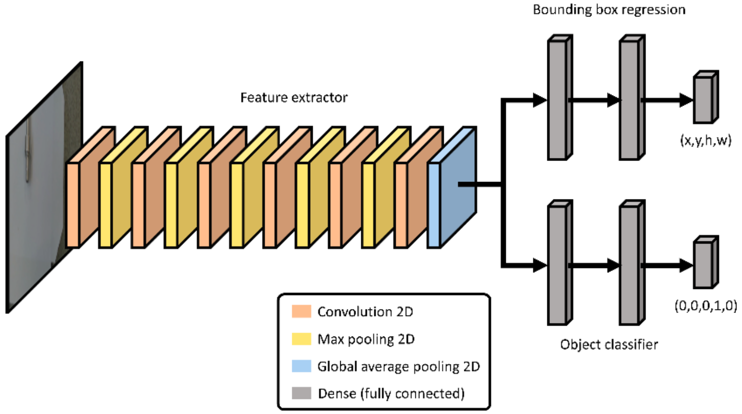
F20_20_B



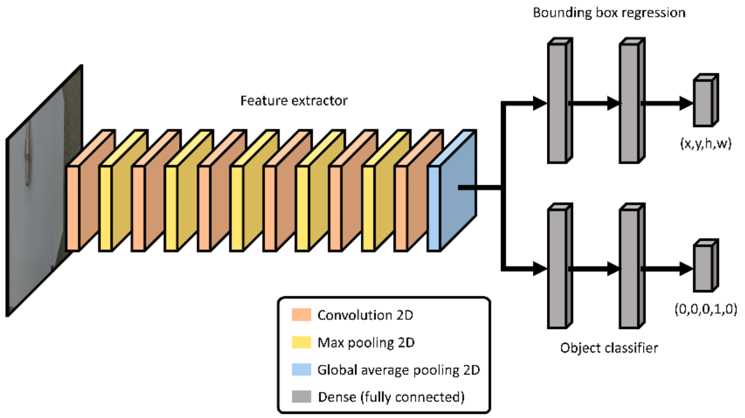
Common Feature Extractor



```
feature_extractor = tf.keras.models.Sequential([
    tf.keras.layers.Conv2D(filters = 32,
                           kernel_size = (3,3),
                           activation = 'relu',
                           padding = 'SAME',
                           input_shape = (240, 320, 3)),

    tf.keras.layers.MaxPool2D(pool_size = (2,2)),

    tf.keras.layers.Conv2D(64, (3,3), activation = 'relu', padding = 'SAME'),
    tf.keras.layers.MaxPool2D((2,2)),

    tf.keras.layers.Conv2D(64, (3,3), activation = 'relu', padding = 'SAME'),
    tf.keras.layers.MaxPool2D((2,2)),

    tf.keras.layers.Conv2D(128, (3,3), activation = 'relu', padding = 'SAME'),
    tf.keras.layers.MaxPool2D((2,2)),

    tf.keras.layers.Conv2D(128, (3,3), activation = 'relu', padding = 'SAME'),
    tf.keras.layers.MaxPool2D((2,2)),

    tf.keras.layers.Conv2D(256, (3,3), activation = 'relu', padding = 'SAME'),
    tf.keras.layers.GlobalAveragePooling2D()
])
```



Classifier and Bounding Box Regressor

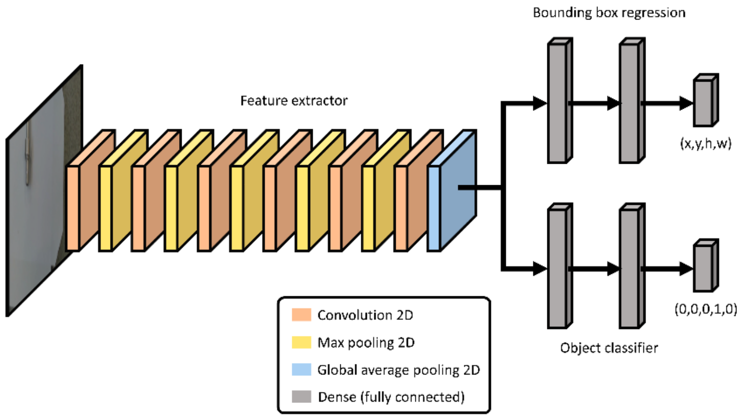


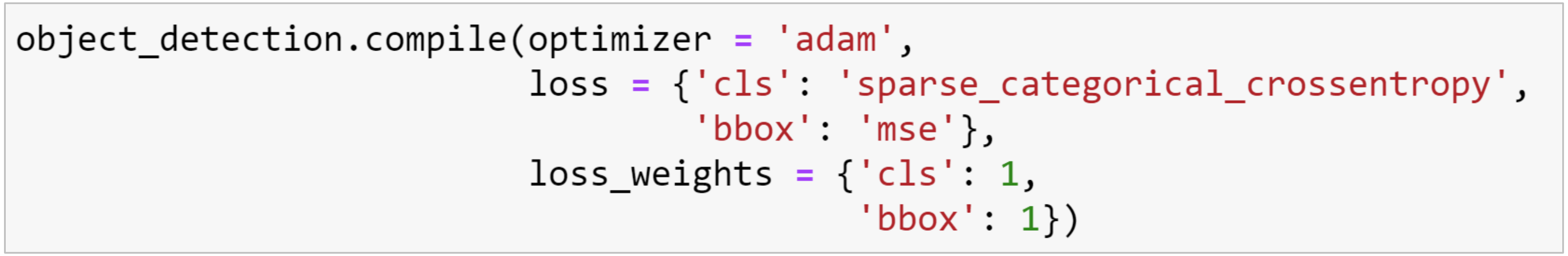
```
classifier = tf.keras.layers.Dense(256, activation = 'relu')(feature_extractor.output)
classifier = tf.keras.layers.Dense(256, activation = 'relu')(classifier)
classifier = tf.keras.layers.Dense(5, activation = 'softmax', name = 'cls')(classifier)
```

```
bb_regressor = tf.keras.layers.Dense(256, activation = 'relu')(feature_extractor.output)
bb_regressor = tf.keras.layers.Dense(256, activation = 'relu')(bb_regressor)
bb_regressor = tf.keras.layers.Dense(4, name = 'bbox')(bb_regressor)
```

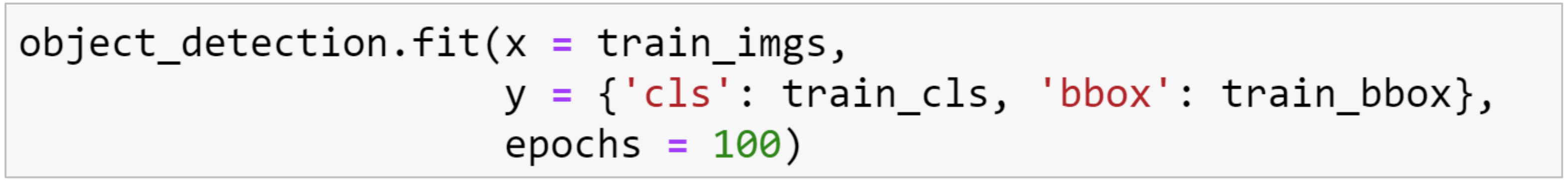
```
object_detection = tf.keras.models.Model(inputs = feature_extractor.input,
                                         outputs = [classifier, bb_regressor])
```

Losses and Optimization Configuration



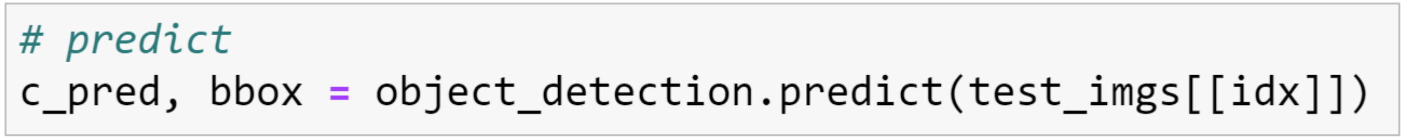


```
object_detection.compile(optimizer = 'adam',
                         loss = {'cls': 'sparse_categorical_crossentropy',
                                 'bbox': 'mse'},
                         loss_weights = {'cls': 1,
                                         'bbox': 1})
```

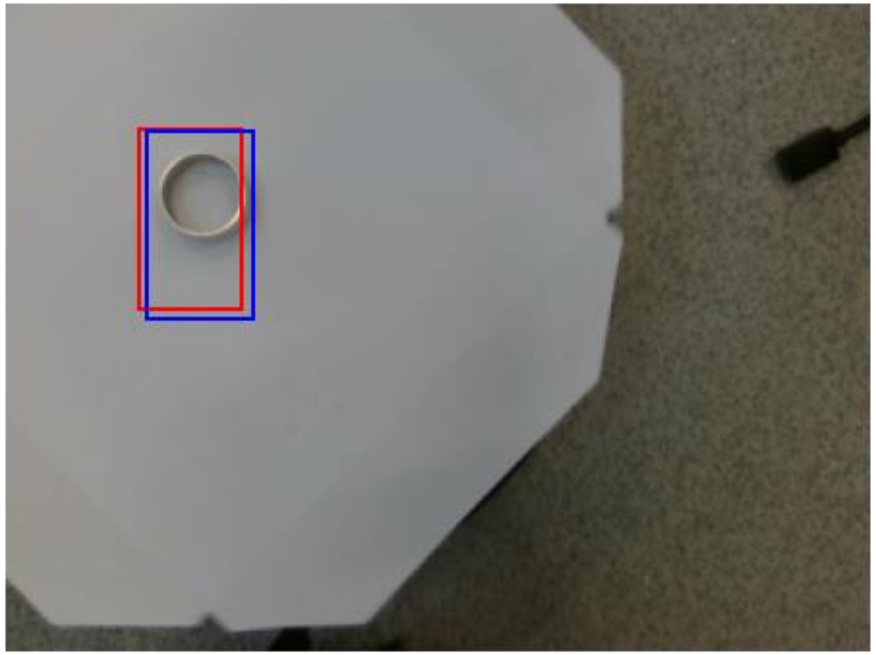


```
object_detection.fit(x = train_imgs,
                     y = {'cls': train_cls, 'bbox': train_bbox},
                     epochs = 100)
```

Result

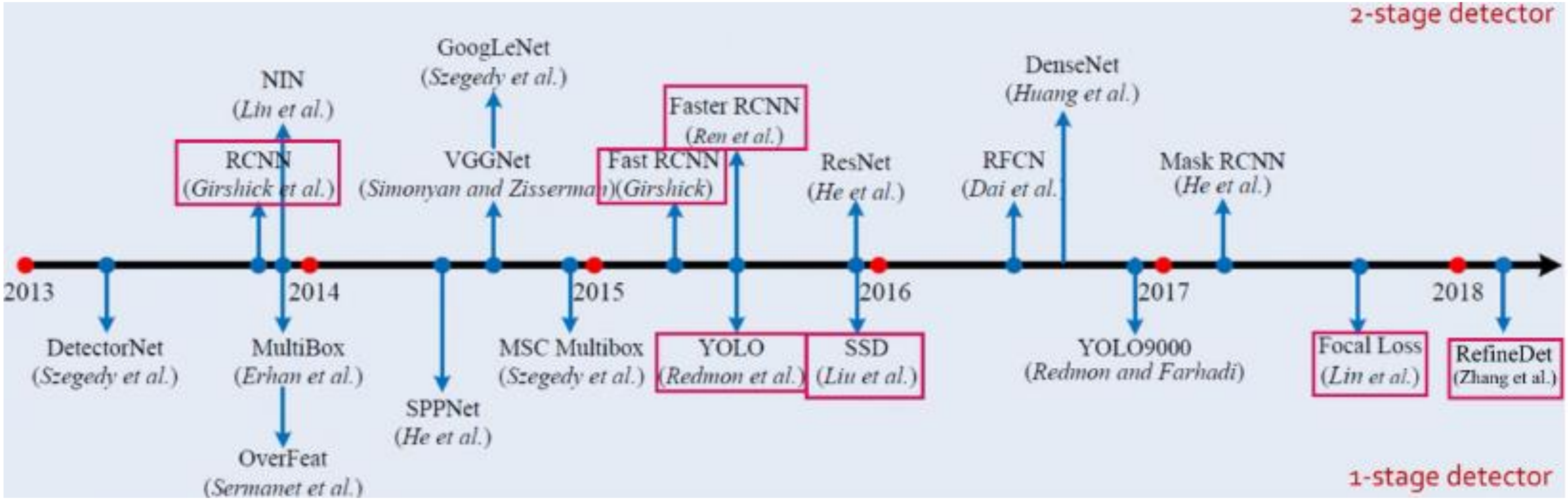


```
# predict
c_pred, bbox = object_detection.predict(test_imgs[[idx]])
```



In Practice

- Use pre-trained models and transfer learning technique for object detection task


2-stage detector
GoogLeNet
(Szegedy et al.)
NIN
(Lin et al.)
RCNN
(Girshick et al.)
VGGNet
(Simonyan and Zisserman)
Faster RCNN
(Ren et al.)
Fast RCNN
(Girshick)
DenseNet
(Huang et al.)
ResNet
(He et al.)
RFCN
(Dai et al.)
Mask RCNN
(He et al.)
2013
2014
2015
2016
2017
2018
DetectorNet
(Szegedy et al.)
MultiBox
(Erhan et al.)
OverFeat
(Sermanet et al.)
SPPNet
(He et al.)
MSC Multibox
(Szegedy et al.)
YOLO
(Redmon et al.)
SSD
(Liu et al.)
YOLO9000
(Redmon and Farhadi)
Focal Loss
(Lin et al.)
RefineDet
(Zhang et al.)
1-stage detector