Project Milestone 2

Mohamed Kamal AbdEl rahman 60

Mohamed Elmaghraby Mohamed 55

Problem statement

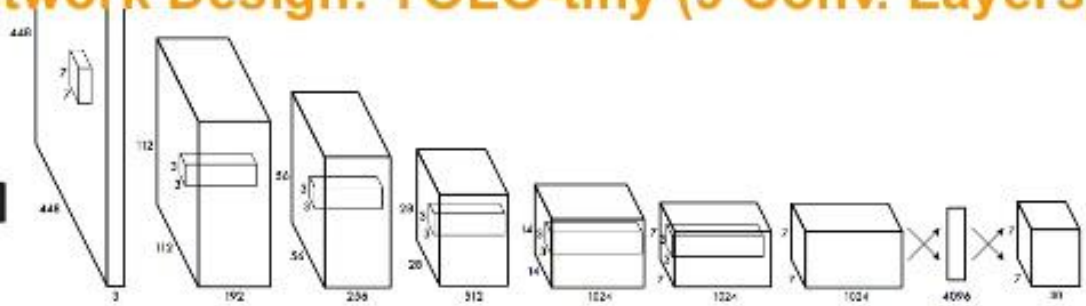
- Our project focuses on moving object detection which deals with identifying and locating objects of certain classes in a stream of videos.
- The best accuracy achieved so far is on coco dataset which is 37.4 by CenterNet on 52 fps.

We would use MOT video of labeled bounding boxes as a data sets and IOU and precession accuracy.

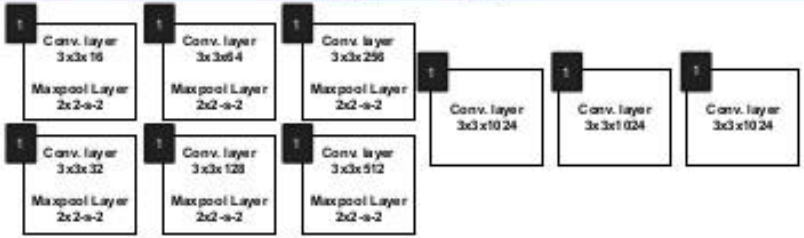
Original model

Network Design: YOLO-tiny (9 Conv. Layers)

YOLO



YOLO-Tiny



Yolo-tiny VS Yolo

1. Number of conv layers in yolo is more than it in yolo-tiny.
2. There is a variety in filter dimensions in yolo while in yolo-tiny they use only 3*3 filters.
3. Each block in yolo consist of many conv layers and one max pooling while in yolo-tiny each block consist of one con layer and one block.
4. Original model of both yolo and tiny yolo has skip connections while ours doesn’t.

Loss Function

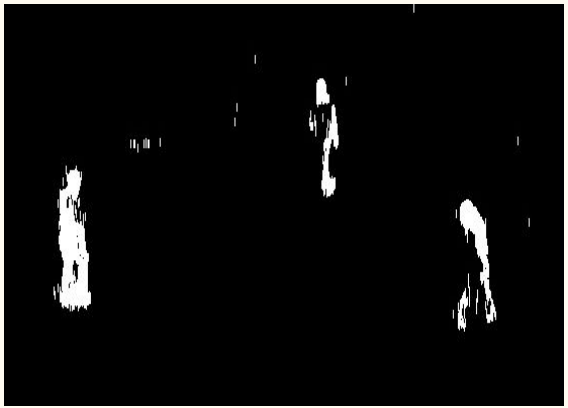
$$
\begin{aligned}
&\lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left(x_i - \hat{x}_i\right)^2 + \left(y_i - \hat{y}_i\right)^2 \\
&+ \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left(\sqrt{w_i} - \sqrt{\hat{w}_i}\right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i}\right)^2 \\
&+ \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left(C_i - \hat{C}_i\right)^2 \\
&+ \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{noobj}} \left(C_i - \hat{C}_i\right)^2 \\
&+ \sum_{i=0}^{S^2} \mathbb{1}_{i}^{\text{obj}} \sum_{c \in \text{classes}} \left(p_i(c) - \hat{p}_i(c)\right)^2 \qquad (3)
\end{aligned}
$$

proposed solution

Feature fusion?

Combining two or more different feature representation for the data two come up with combined feature that more representative than both of original features.

Example



Is that enough?

No, because the foreground mask is not always that accurate due to:

- Possibilities of noise.
- It is hard to accurately compute it in dynamic background.

Current models achieved respectable accuracy but face the real time constraint on this problem so we proposed

- since we are working on a stream of videos we use a background subtraction technique to use a foreground mask and integrated it as another channel to the image input.
- The main goal is to increase the accuracy without increasing the inference time of the model.

We selected to use Tiny Yolo as

- Yolo considered the first model to achieve good performance near real-time.
- Use the tiny version for speed of training and inference.
- And has a simple architecture

Difference

Input Format

- In original model the input images are 3 channels (RGB)
- While in our model the input images are 4 channels, the extra one is a mask for foreground to have the input image as foreground only
- The foreground is computed by background subtraction using mixture of gaussian.

Loss Function

- The original model works with many object types (person, dog,...)
- In our model we only focus on persons to simplify our training process.

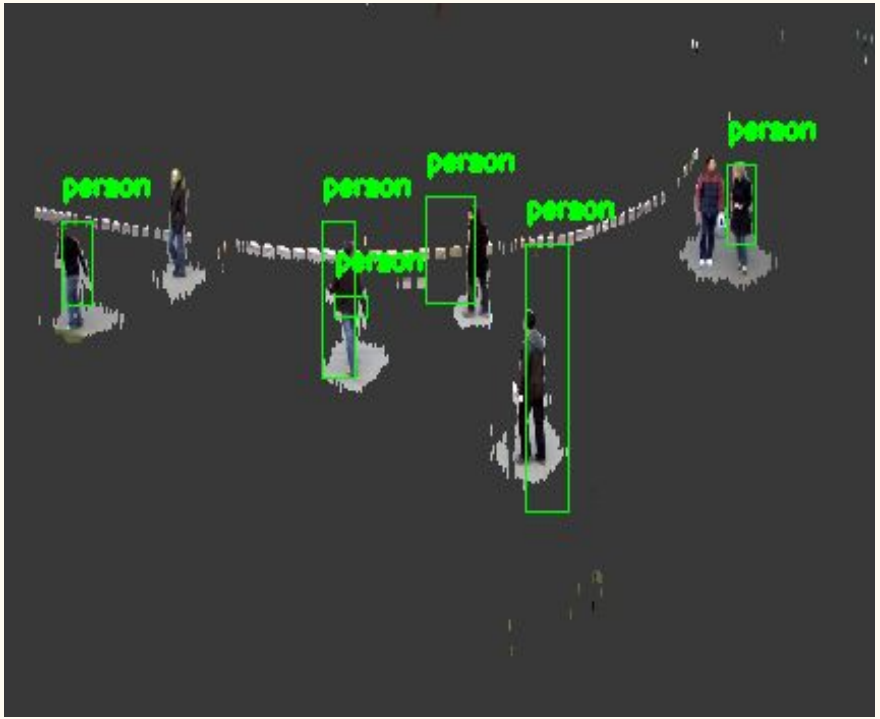
Evaluation

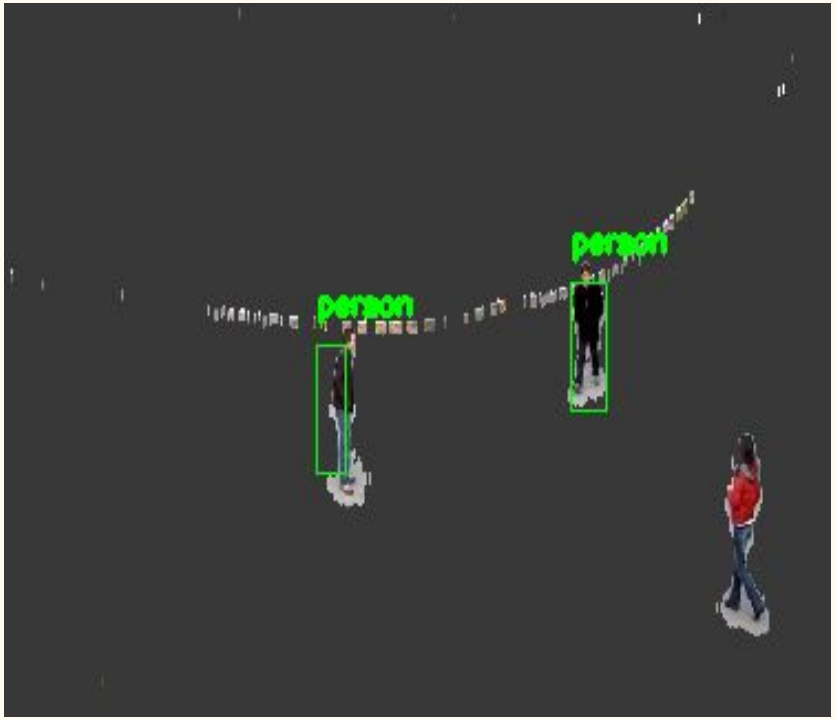
Model training

1. Training on sample of dataset.
2. Starting from random weights.
3. Hyper parameter

5 Anchor boxes	`'1.08,1.19, 0.44,1.31, 0.49,1.51, 0.59,1.85, 0.82,2.51'`
IOU threshold	0.5
Non-max suppression	0.3

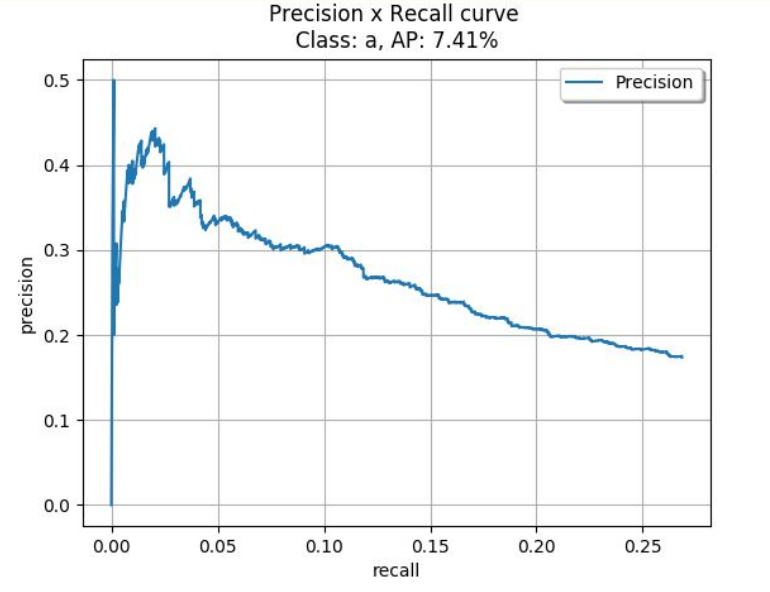
Sample results



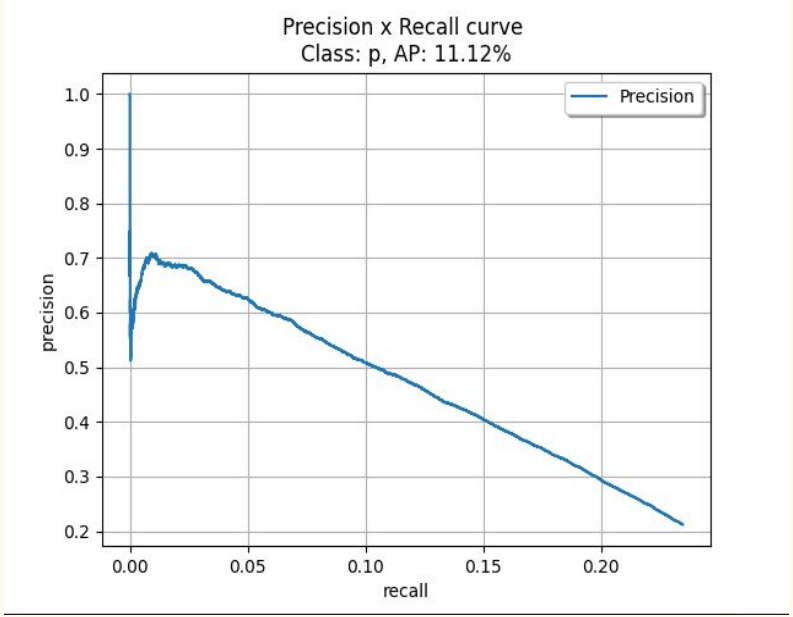


Our Model vs Tiny Model

Our model result



Yolo-tiny result



Conclusion

1. Works well on stationary camera.
2. Predict pounding box with low IOU.
3. Need more investigation to the technique as it is new to solve this problem.
4. The model need to be tested against images with bad foreground mask.

Current progress

- Implemented the utility functions
  - Background subtraction
  - Resize image
  - Evaluate accuracy function
- Download Yolo Tiny weights and configurations
- Download MOT datasets
- Run existing models
- Transform images to 4 channel format
- Build tiny-yolo architecture in keras
- Build loss functions and helper functions
- Train our model on the new data set

Next steps

- **Train the model on large set of images.**
- **Make our implementation as a program.**
- **Test our approach with larger architecture models (Yolo, RetinaNet, etc) (may be in near future not in next milestone).**

Resources

1. Tensorflow object detection API

1. Pakorn KaewTraKulPong and Richard Bowden. An improved adaptive background mixture model for real-time tracking with shadow detection. In Video-Based Surveillance Systems, pages 135–144. Springer, 2002.

1. mAP (mean Average Precision) for Object Detection

1. Object-Detection-Metrics

1. Yolo implementaion

1. Tiny-yolo implementation on keras

Thank You