Lecture Notes
Part II

CmpE 258
Spring 2023

March 21 (Tue).

Midterm is scheduled on the 23rd (Thu). 1 hr Exame.
16:30 - 17:30
Then 15 minutes for prep & Uploading the file.

Example: Softmax Activation Function. MNIST CNN

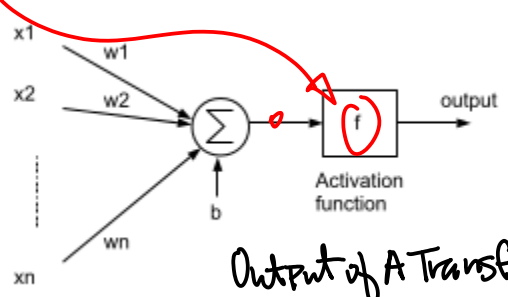
Note:

$$\sigma(\mathbf{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \quad \text{for } i = 1, \ldots, K \text{ and } \mathbf{z} = (z_1, \ldots, z_K) \in \mathbb{R}^K \quad \ldots (1)$$

Dimension

Let K=1

1° $f(z)$ Notation



Output of A Transfer function

2° Index

$$f(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \quad \ldots (2)$$

Total No. of Output Neurons = K
for Handwritten Digits Recognition
K = 10;

where

$$\sum_{j=1}^{K} e^{z_j} = e^{z_1} + e^{z_2} + \cdots + e^{z_K} \geq e^{z_i} \quad \ldots (3)$$

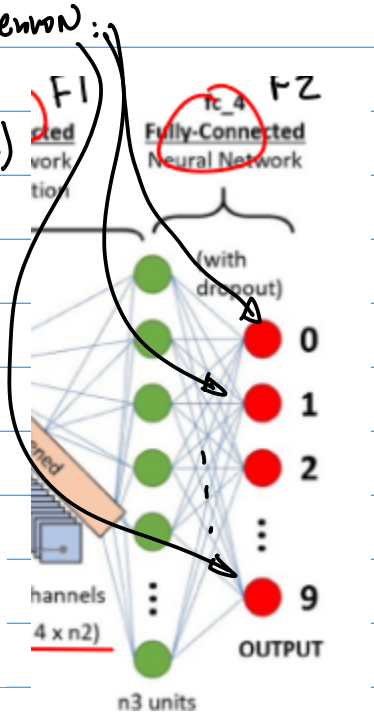
Property 1.

Output from Each Neuron:

$$f(z_1) = \frac{e^{z_1}}{\sum_{j=1}^{10} e^{z_j}} \quad \ldots (4)$$

for Digit "0" (1st Output)

And



$$f(z_2) = \frac{e^{z_2}}{\sum_{j=1}^{10} e^{z_j}} \quad \text{for Digit "1" (2nd output)} \quad \ldots (5)$$

$$\vdots$$

$$f(z_{10}) = \frac{e^{z_{10}}{\sum_{j=1}^{10} e^{z_j}} \quad \text{for Digit "9" (10th O/P)}$$

If Add Eqn (1) + Eqn (2) + ⋯

$$f(z_1) + f(z_2) + \cdots + f(z_{10})$$

$$= \frac{e^{z_1}}{\sum_{j=1}^{10} e^{z_j}} + \frac{e^{z_2}}{\sum_{j=1}^{10} e^{z_j}} + \cdots\cdots + \frac{e^{z_{10}}}{\sum_{j=1}^{10} e^{z_j}}$$

$$= \frac{\sum_{j=1}^{10} e^{z_j}}{\sum_{j=1}^{10} e^{z_j}} = 1. \quad \ldots (6)$$

The softmax function takes as input a vector z of K real numbers, and normalizes it into a probability distribution consisting of K probabilities proportional to the exponentials of

$$f(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}$$

See Eqn (6).

Now, move to the 2nd Half of DCNN.

2022F-108a-Yolo-architecture-loss-function-2022-10-10.pdf

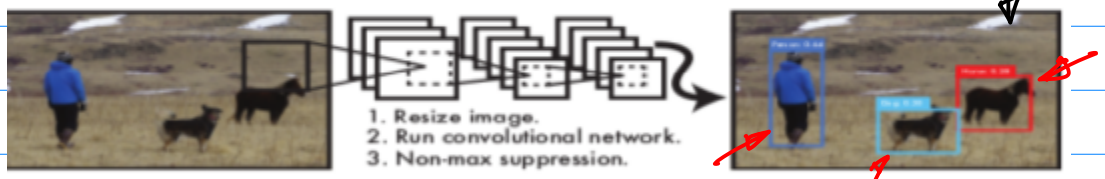
Typical Classification/Recognition Results are in Bounding Boxes

://arxiv.org/pdf/1506.02640v5.pdf

h Divvala, Ross Girshick, Ali Farhadi
, Allen Institute for AI,Facebook AI Research http://pjreddie.com/yolo/

You Only Look Once:
Unified, Real-Time Object Detection

Joseph Redmon*, Santosh Divvala*†, Ross Girshick¶, Ali Farhadi*†
University of Washington*, Allen Institute for AI†, Facebook AI Research¶
http://pjreddie.com/yolo/



del runs at 45 FPS. A smaller version, Fast YOLO, runs astounding 155
itperforms DPM (deformable parts models) and R-CNN.

Figure 1: The YOLO Detection Sy
image to 448 × 448, (2) runs a co
network, and (3) thresholds the re

Then, we would like to Achieve Semantic Segmentation.

Pixel by pixel Based Segmentation/Detection/Recognition

PART II (After the midterm)

April 4 (Tue)

Road map: Yolo (You-Only-Look-ONCE) → Semantic Segmentation.

{ Project Assignment = Yolo Implementation Due 2½ weeks. April 20th. (Thursday)

CMPE258

Spring 2023

Homework (In-Class Presentation) Due April 6 (Thu).

Requirements

1° One paragraph Description (Abstract) of the proposed Semester-Long Project.

2° Title

Team members : First Name, LastName, Major

Team Coordinator.

Contact E-mail.

3° Abstract Part.

Objective(s) : a) What is the proposed work;

b) What is the coding /training/ Testing Task involved in the project ?

c) Anticipated Result ? And deliverable ?

d) Tools, platform, Programming Language Version, T.F., Pytorch, ChatGPT etc.

Also, Define Python Packages, OpenCV.

Example: On Yolo.

Ref:

2022F-106-README-Tiny-Yolo4-GP...

Note1 : Readme for Yolo github 34 Installation & Testing. for the project.

Title: README Tiny Yolo v4 GPU Ubuntu

Document Number: 105-1b

CTI One Corporation

Table 1a. Document History

2022-10-6	Establish this document, document archive: (base) harry@workstation:~/yolo-2022-10-19$	YY

1. Setup YOLO v4 environment

1.1. Clone the GitHub folder;

$ git clone https://github.com/pythonlessons/TensorFlow-2.x-YOLOv3.git

1.2. Create YAML file for building the YOLO v4 Anaconda environment;

Create TensorFlow-2.x-YOLOv3/conda-gpu.yml as the following:

==========

name: yolov4-gpu

Ref : Introduction

2022F-108a-Yolo-architecture-loss-function-2022-10-10.pdf

Base-Line Ref for Yolo Technique

2022S-112-yolo-paper.pdf

You Only Look Once:
Unified, Real-Time Object Detection

Joseph Redmon*, Santosh Divvala*†, Ross Girshick¶, Ali Farhadi*†

University of Washington*, Allen Institute for AI†, Facebook AI Research¶

http://pjreddie.com/yolo/

Lecture Notes : Base Line Ref/Requirements.

2022F-101-cmpe258-note-2022-11-1.pdf

Example: Notations for YoLo.

Ref, pp36.

CmpE258
Oct. 13, 22

Fig 1.

Note:

1. Image I(x,y) is divided into SxS Grids.
Denote it as $G_{p,q}(x,y)$, ... (1)
where p,q = 0,1,2,... indicate the

1) Image I(x,y). with Resolution M x N

M: No. of Col. N: No. of Rows.

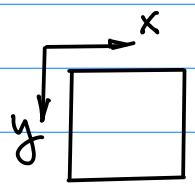


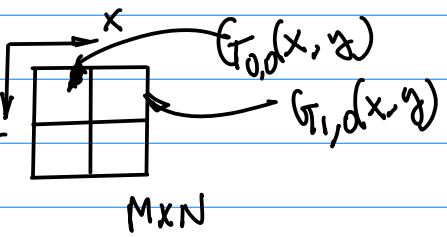
Fig. 1b.

Divide I(x,y) into SxS Grids.
Each Grid is Denoted as

$G_{pq}(x,y)$ ... 1)

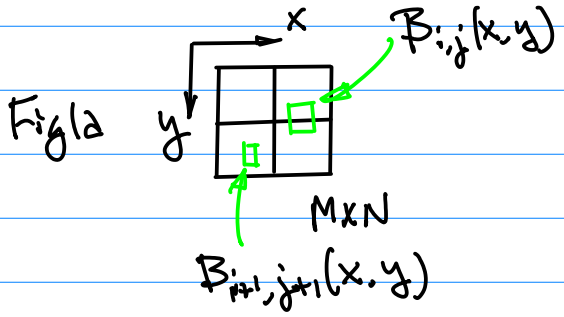
where p,q = 0,1,2,...,S-1 matches to x

Fig 1c



2. Bounding Boxes $B_{i,j}(x,y)$ ... (2)
i for x-direction, j for y-direction

Fig 1a



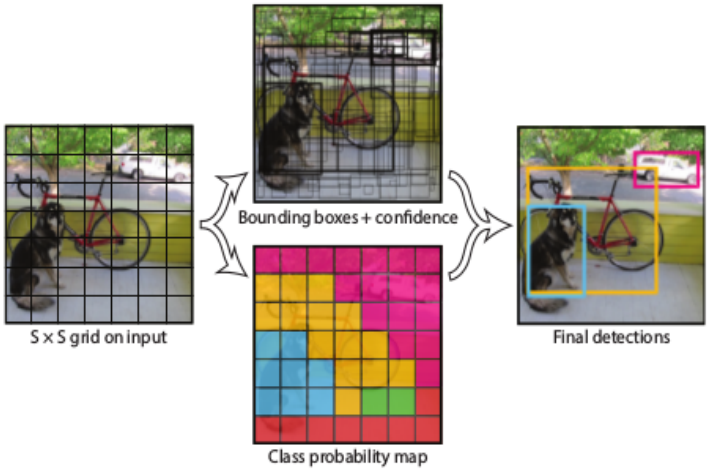
3. Five Parameters to define each Bounding Box.

(x,y): Location of the top Left Corner of $B_{i,j}(x,y)$

w,h: Width and Height of $B_{i,j}(x,y)$

f: Confidence Level, Probability distribution to Describe the likelihood of the B.B. (B^2) belongs to a certain Class of objects.

(x, y, w, h, f) ... (3)



Note 1. α. $G_{ij}(x,y)$ Grid.

Figure 2: The Model. Our system models detection as a regression problem. It divides the image into an $S \times S$ grid and for each grid cell predicts B bounding boxes, confidence for those boxes, and C class probabilities. These predictions are encoded as an $S \times S \times (B * 5 + C)$ tensor.

b. $B_{ij}(x,y)$

Class probability.

c. (x, y, w, h, f)

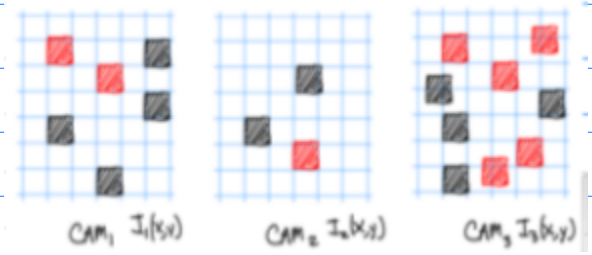
Probability

Confidence.

Attn (Th).

Example: Discussion on Notation/Formulation.

Ref. 2022F-101-cmpe258-note-2022-11-1.pdf pp38



Camera 1: $I_1(x,y)$ Camera 2 $I_2(x,y)$ Cam 3. $I_3(x,y)$

$$R = RI_1 + RI_2 + RI_3 \quad \ldots (1)$$

R: Red Squares, Persons.

B: (Black) Vehicles.

for "Union" "$\cup$"

Intersection. "$\cap$"

Consider the probability of the event "R" (meaning Person(s) being Capture on any one of these images).

$$Prob(R) = Prob(RI_1) + Prob(RI_2) + Prob(RI_3) \quad \ldots (2)$$

$I_1 \cap I_2 \cap I_3 = \phi$ (Empty set)

Consider Each Individual Camera:

$$Prob(RI_1) = Prob(R|I_1)\,Prob(I_1) \quad \ldots (3a)$$

Similarly,

$$Prob(RI_2) = Prob(R|I_2)\,Prob(I_2) \quad \ldots (3b)$$

$$Prob(RI_3) = Prob(R|I_3)\,Prob(I_3) \quad \ldots (3c)$$

Rewrite Eqn (2):

$$Prob(R) = Prob(R|I_1)\,Prob(I_1) + Prob(R|I_2)\,Prob(I_2) + Prob(R|I_3)\,Prob(I_3) = \sum_{i=1}^{3} Prob(R|I_i)\,Prob(I_i) \quad \ldots (4)$$

CMPE258
Spring 2023

Ref: 2022S-112-yolo-paper.pdf

$$\Pr(\text{Class}_i|\text{Object}) * \Pr(\text{Object}) * \text{IOU}^{\text{truth}}_{\text{pred}} = \Pr(\text{Class}_i) * \text{IOU}^{\text{truth}}_{\text{pred}} \quad (1)$$

Note 1. ↑ Conditional Probability. Prob(C_i)

Prob(C_i|Obj) Prob(Obj)

1. IOU (Intersection of Union)

Index for the purpose of comparing 2 Bounding Boxes at a time

pp.37

Note: "GT" Ground Truth.

the Union of $B_{ij}(x,y)$ $B_{GT}(x,y)$

Note: $B_{ij}(x,y)$ for the i,j B.B.

The Union

The Intersection of $B_{ij}(x,y)$, $B_{GT}(x,y)$

Fig 1

The prediction matches the Ground Truth. (GT) is on the Left

$$\text{IOU} = \frac{\text{Intersection of } B_{ij}(x,y) \text{ and } B_{GT}(x,y)}{\text{The Union of } B_{ij}(x,y) \text{ and } B_{GT}(x,y)} \quad \cdots (5)$$

pp37

$$C = \frac{\square}{\square}$$

(IOU) Confidence, denoted as C

~~...(4)~~

Hand Calculation of IOU.

Example: pp36

5. IOU (Intersection of Union).

Example: Illustration of IOU

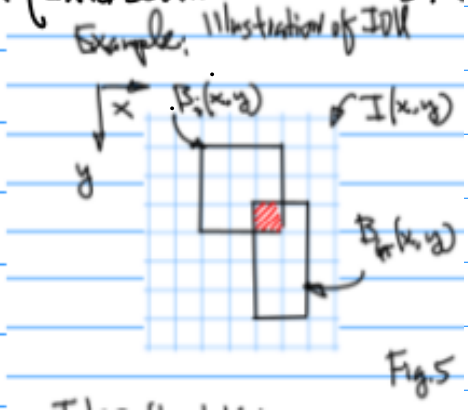


Fig 3

Fig 5

Sol. First, find the Number of pixels of the $B_{ij}(x,y) \cap B_{GT}(x,y)$

$B_{ij}(x,y) \cap B_{GT}(x,y) = 1$

then, Find the Union

$B_{ij}(x,y) \cup B_{GT}(x,y) =$

$N[B_{ij}(x,y)] + N[B_{GT}(x,y)]$

$- N[B_{ij}(x,y) \cap B_{GT}(x,y)]$

$= (3 \times 3) + (2 \times 4) - 1 = 9 + 8 - 1$

$= 17 - 1 = 16$

$\therefore$ IOU $= 1/16$

Now, from Eqn (1) of the Ref. (Research Paper)

Note

$$\Pr(\text{Class}_i|\text{Object}) * \Pr(\text{Object}) * \text{IOU}^{\text{truth}}_{\text{pred}} = \Pr(\text{Class}_i) * \text{IOU}^{\text{truth}}_{\text{pred}} \quad (1)$$

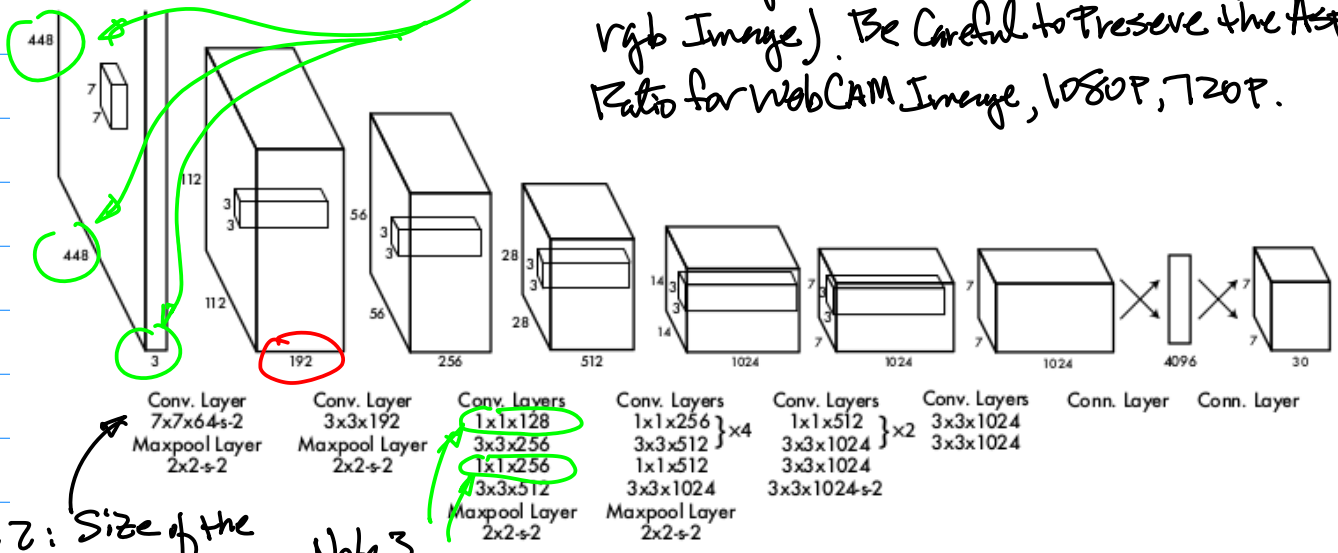
IOU truth — "G.T." B.B.

IOU pred — Predicted B.B.

YOLO

Example: Architecture:

Note 1. Input Image Resolution: 448x448x3 (Channels: rgb Image). Be Careful to Preseve the Aspect Ratio for WebCAM Image, 1080P, 720P.



Note 2: Size of the Kernel: 7x7, 64 of them.

Note 3. 1x1 Convolution is utilized here

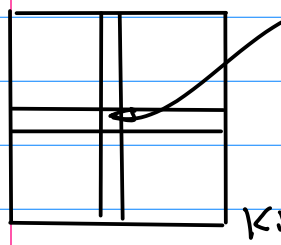


Fig. 1

for KxK 2D Convolution, the output of the convolution. "Spatial Information", Neighbouring pixels under the KxK kernel are Counted for (for feature extraction @ the center of the Kernel)

April 11 (Tue).

Presentation (Brief) on Each Team Project.

Example: Continuation On Math Formulation.

1x1 Convolution.

Note 1. Background On KxK Convolution.

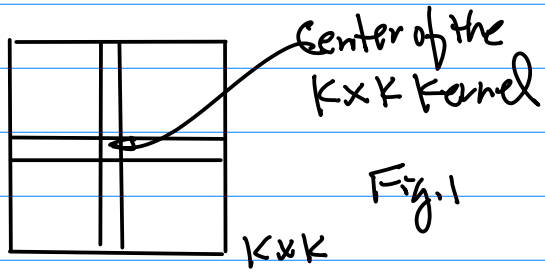


Fig. 1

Output: 1 pixel
Input: KxK pixels.

Captures All Neighbouring pixels at a time, And Produce One pixel Output.

Note 2. for each convolution Kernel, the Convolution Conducted will result in One Output Feature Layer

As we continue the Convolution Process, the Number of Output feature Layers will grow Significantly. Therefore, there's a need to Reduce the Number of Layers without missing crucial features.

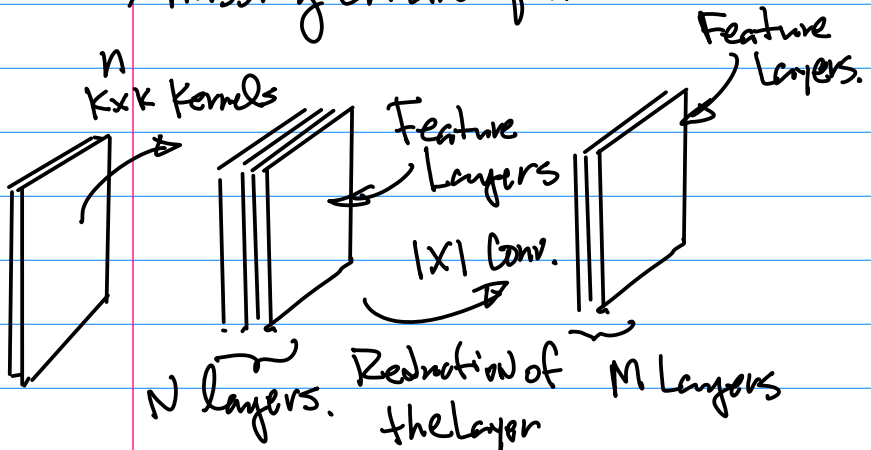


Fig.1.

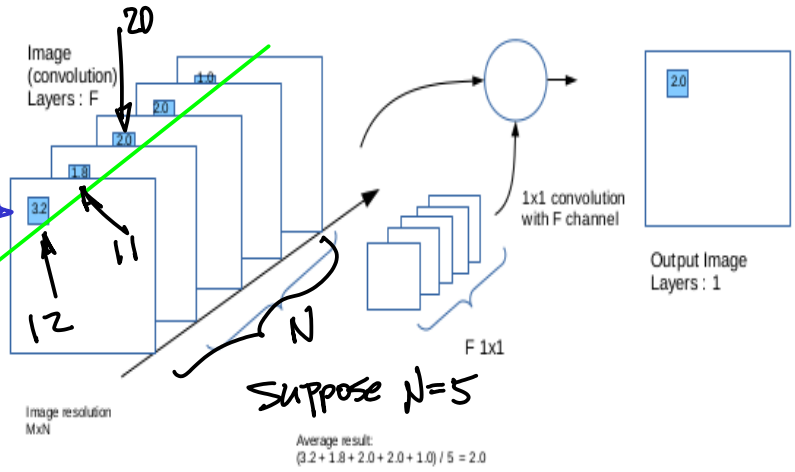
To Be Able to Extract/Preserve the key features to Achieve Reduction of Layers. We are using the following Technique.

2022F-108a-Yolo-architecture-loss-function-2022-

1x1 Convolution for Dimension Reduction and Pooling

The 1x1 convolution enables dimension reduction by reducing the number of channels in convolution layers

1. Suppose the input layers is C*H*W, where C is its channels. The 1x1 convolution generates one average result in shape H*W. The 1x1 (filter) is a vector of length C.
2. Now if you have F 1x1 filters, you get F layers of output, the output shape is F*H*W. For input layer C*H*W with F 1x1 convolution (with channel is C), you will get F*H*W layers.



Harry Li, Ph.D.

Note1.

One pixel across the entire stack of the feature layer.

Reduction Requirement: Combine N=5 layers into 1 layer.

To preserve the feature in this process.

Question: What is the technique to combine them (pixels at different layers) with equal contribution from Each layer?

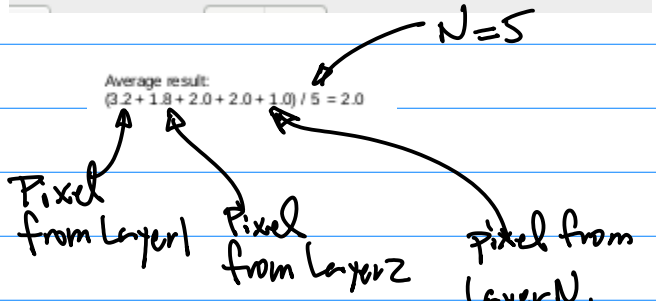
$$\frac{1}{N}\left[I_1(x_1,y_1) + I_2(x_1,y_1) + \cdots + I_5(x_1,y_1)\right] \quad \cdots (1)$$

$$\frac{1}{N}I_1(x_1,y_1) + \frac{1}{N}I_2(x_1,y_1) + \cdots + \frac{1}{N}I_5(x_1,y_1)$$

April 12(th).

Example: 1x1 Convolution.

2022F-108a-Yolo-architecture-loss-function-2022-10-10.pdf



Note: The Average Operation treats the feature Equally from Each Layer.

More General Case:

$$\alpha_1 I_1(x_1,y_1) + \alpha_2 I_2(x_1,y_1) + \cdots + \alpha_N I_N(x_1,y_1)$$

where $\alpha_1 + \alpha_2 + \cdots + \alpha_N = 1$ $\cdots (2)$

Note: Project Assignment (10 pts)
on Object Recognition.
Due April 30th (Sunday, 11:59pm).

CMPE258
YOLACT Semantic Segmentation and Comparison with YOLO
HL

PART I YOLO (5 points)

test run YOLO v4 based on the readme document given in the class githu
.com/hualili/opencv/blob/master/deep-learning-2022s/2022F-106-READ
-YY-HL-v2-2022-10-20.pdf

nart phone to take record a 15 - 30 second video clip for YOLO v4 object

Note: To config Anaconda Enviroment,
use the Pre-Created Configuration
files here, from the github

2022S-104b-conda-gpu.yml

2022S-104c-conda-cpu.yml

Then, Create Conda-enviroment Config. file

```
$ conda env create -f conda-gpu.yml
```

Next, Activate the Conda Env.

```
$ conda activate yolo4-gpu
```

Then, perform the following task

1.5. Download Tiny YOLO v4 model files;

```
$ wget -P model_data
https://github.com/AlexeyAB/darknet/
```

1.6. Modify the configuration file, TensorFlow-2.x-YOLOv3/yolov3/configs.py;

```
==========================================
13 YOLO_TYPE                = "yolov4" # yolov4 or yolov3

37 TRAIN_YOLO_TINY          = True
==========================================
```

Now, To Run the Yolo Program.

The Env.

(1.) 2.1. Activate the Anaconda environment

```
conda activate yolo4-gpu
```

(2) 2.2. Execute the demo program;

```
$ python detection_demo.py
```

Make the following modification to Run Yolo for the Video file Input. Need to modify the python Code.

3. Execute Tiny YOLO v4 with a video file

3.1. Modify TensorFlow-2.x-YOLOv3/detection_demo.py

Make Change by Replacing Image file input (Line 23) by Video file input (Line 24).

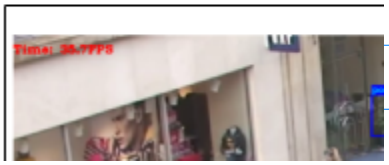
Then, Repeat the Same steps as you did for Yolo Execution with image input.

Note: Execution Speed,

```
$ python detection_demo.py
```

FPS is around 35 on RTX 2070.

FPS : Frames per Second.

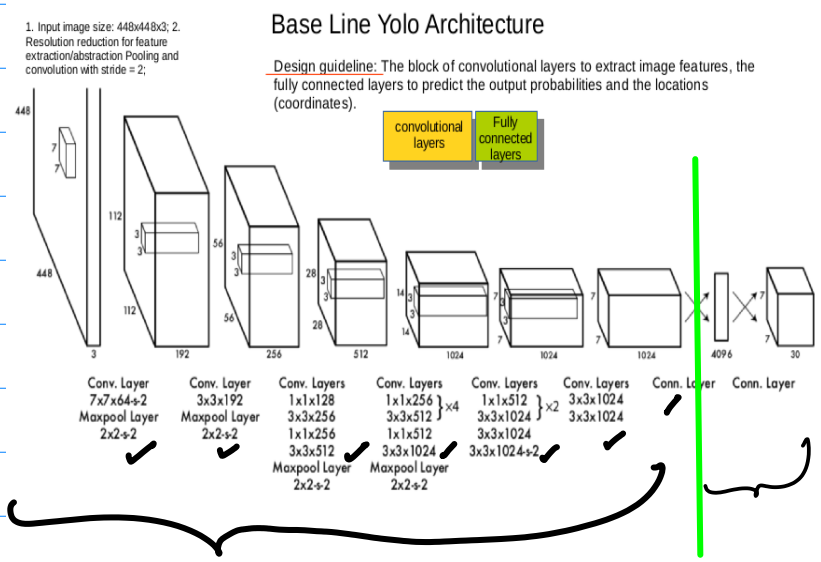


Be Sure to Record Video Clip(s) yourself for Yolo Testing, you Need to use these Videos in the future.

Example: Yolo Architecture.

Note: 1° Understand the Composition of the Yolo Architecture. To Be Able to Describe the function(s) of Each Block;

2° Analyze the Parameter(s) of Each Block.



For Feature Extraction Convolutional Operations. Like/Similar to the Creation of high Dimensional feature Vectors.

$X(x_1, x_2, \cdots, x_N)$

For Classification

Note: Image/Video Input should be Square with preservation of the Original Aspect Ratio.

1. Input image size: 448x448x3; 2. Resolution reduction for feature extraction/abstraction Pooling and convolution with stride = 2;

Stride = 1 Convolution.

"default" Version of Convolution.

Shift/move the mask from Top left Corner Left to Right ONE pixel at a time, And Top to Bottom ONE Row at a time.

Stride = 2.

Shift/move the mask from Top left Corner Left to Right 2 pixel at a time, And Top to Bottom 2 Row at a time.

Example/Exercise Off-Line.

Sub-Sampling + Convolution → Reduction of feature Layers.