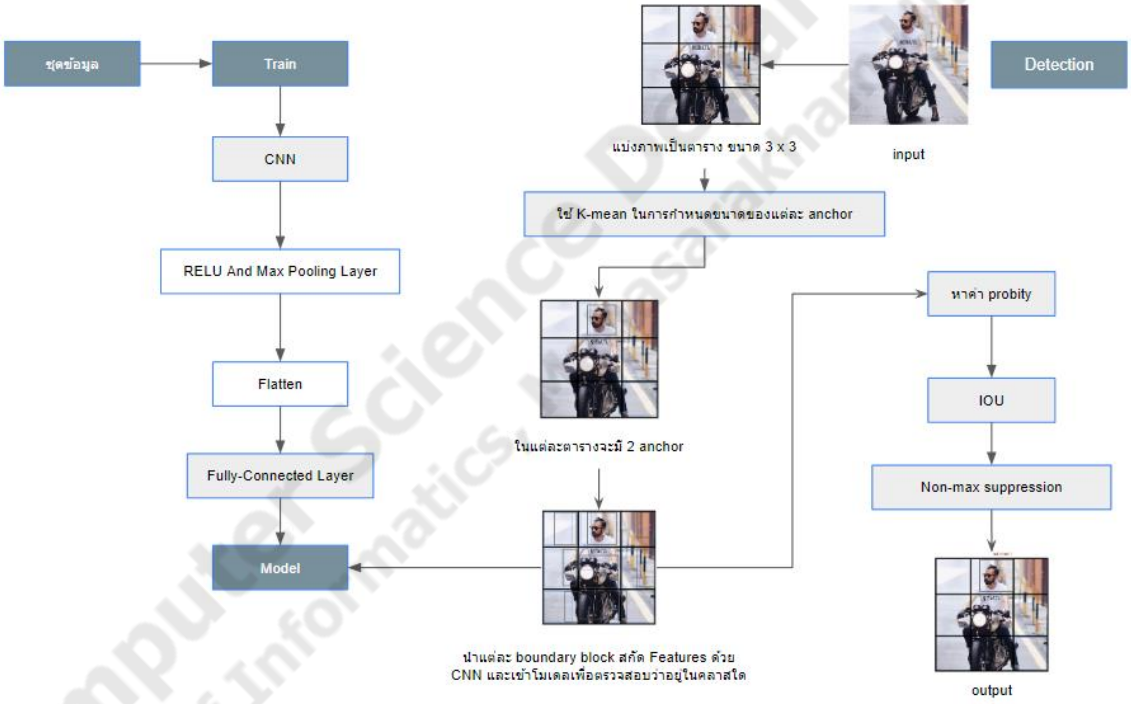
บทที่ 3

ขั้นตอนการดำเนินงาน

ในบทที่ 3 จะอธิบายการเตรียมข้อมูลและวิธีการตรวจจับหมวกนิรภัยด้วย YOLO หรือ You Only Look Once คือ Realtime Object Detection Model ที่มีความโดดเด่นเรื่องความเร็วและความถูกต้องหลักการของ YOLO คือจะพยายาม rectangle object เหล่านั้นไว้โดยตีกรอบเอาไว้และบอกออกมาว่าสิ่งนั้นคืออะไรโดยขั้นตอนการทำงานของ YOLO แสดงดังขั้นตอนต่อไปนี้ดังภาพที่ 3.1



ภาพประกอบที่ 3.1 ภาพตัวอย่างการทำงานของ YOLO

การทำงานของ YOLO จะมีสองส่วนคือ ส่วนของการ Train และ Detection

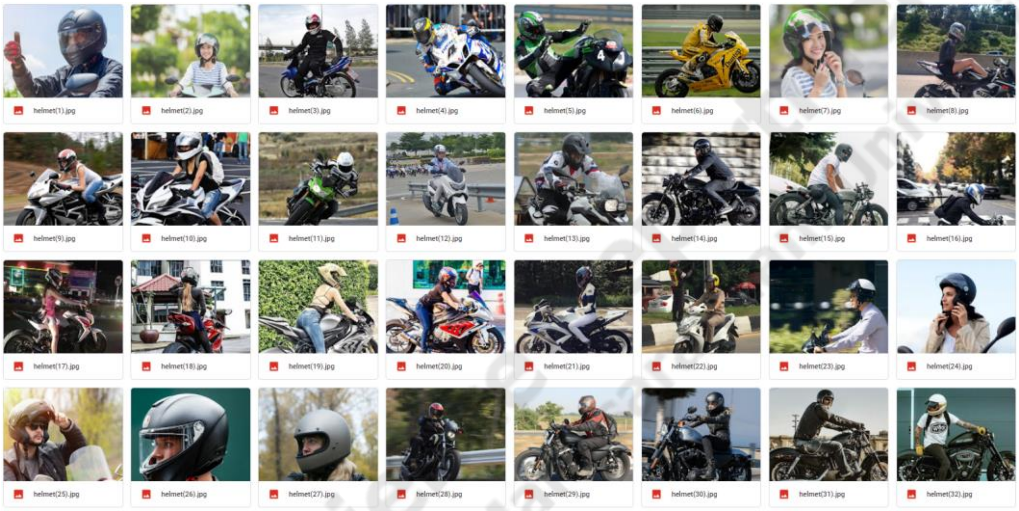
1.ส่วนของการ Train จะนำชุดข้อมูลที่เตรียมไว้ เข้าเทรนด้วย CNN , RELU, Max Polling Layer และ Fully-Connected Layer จึงจะได้ Model

2.ส่วนของการ Detection ขั้นตอนแรกจะรับ input เข้ามาและทำการแบ่งภาพเป็นกริด ต่อไปจะใช้ K-mean ในการกำหนดขนาดของ anchor โดยในแต่ละกริดจะมี 2 anchor เมื่อครบทุกกริดจะนำ boundary block สกัด Features(คุณลักษณะเด่นของภาพ)ด้วย CNN และเข้า Model เพื่อตรวจสอบว่าอยู่ในคลาสใดจากนั้นหาค่า probity โดยหาจากค่า IOU และใช้วิธี non-max suppression

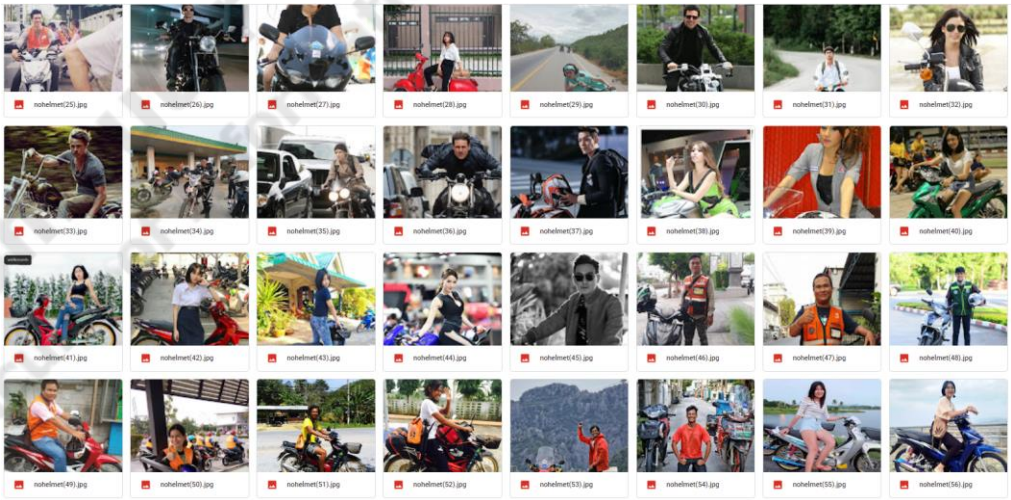
Computer Science Department
Faculty of Informatics
Mahasarakham University

3.1 การเตรียมชุดข้อมูล (Dataset)

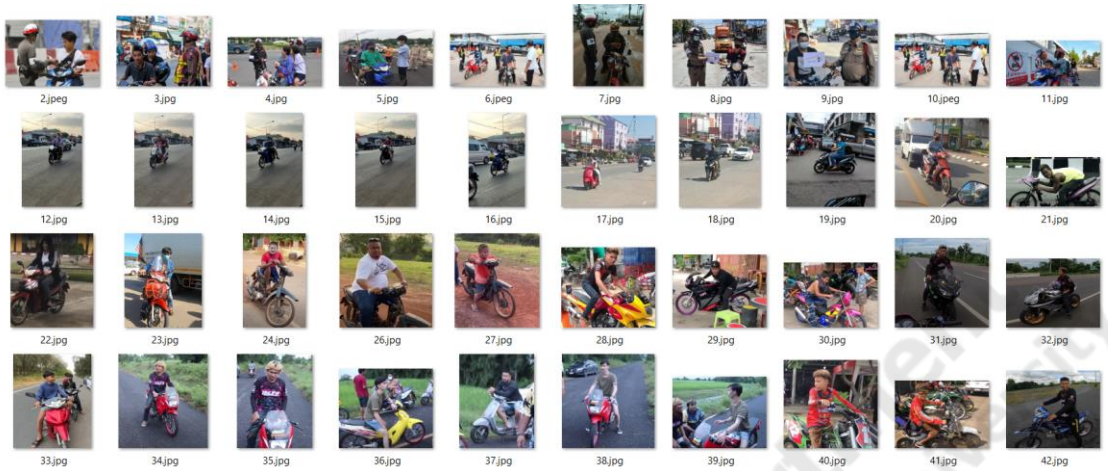
3.1.1 การเก็บข้อมูลที่จะนำมาสร้าง Dataset จำนวน 714 ภาพ แบ่งเป็นภาพที่สวมหมวกนิรภัย 286 ภาพ ไม่สวมหมวกนิรภัย 451 ภาพ ภาพที่ใช้ตรวจสอบความถูกต้อง 111 ภาพ ภาพทดสอบ 104 ภาพ และวีดีโอทดลอง 10 วีดีโอ



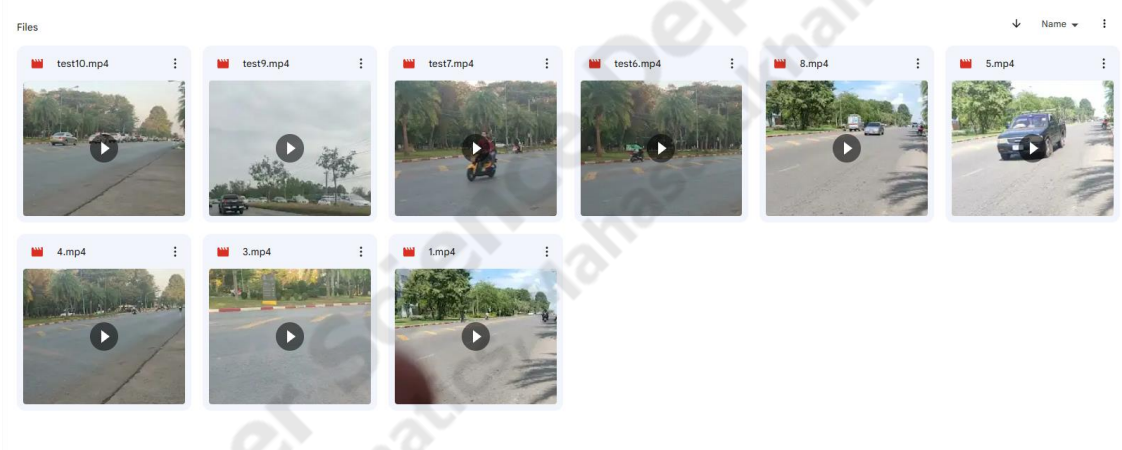
ภาพประกอบที่ 3.2 Dataset helmet



ภาพประกอบที่ 3.3 Dataset nohelmet



ภาพประกอบที่ 3.4 Dataset test



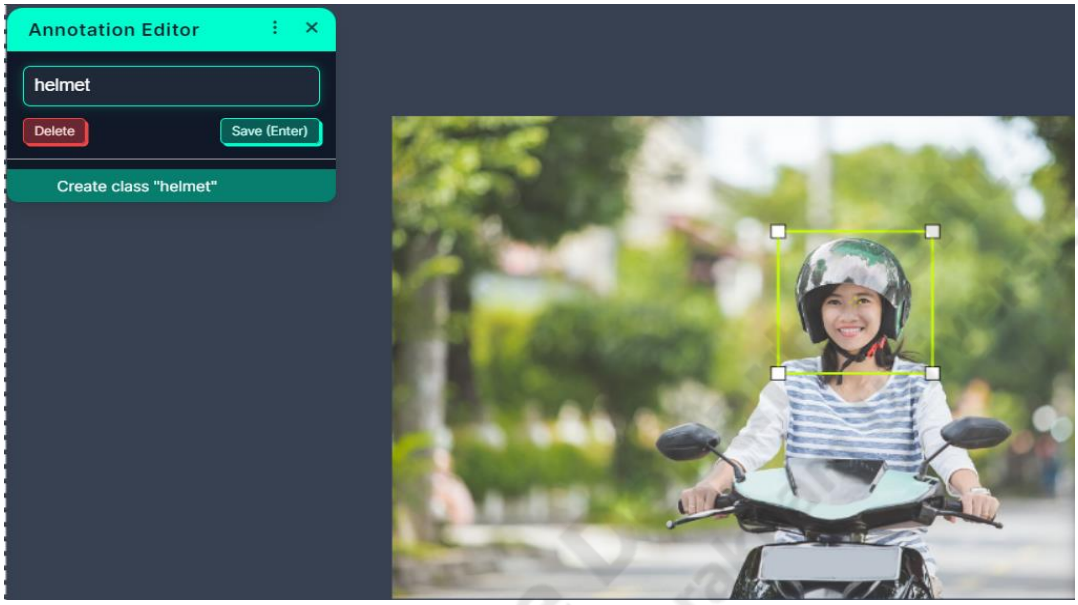
ภาพประกอบที่ 3.5 Dataset video test

3.1.2 การสร้างผลเฉลย โดย Roboflow

Roboflow เป็น computer vision developer framework สำหรับใช้จัดเก็บ เตรียมชุดข้อมูล และสร้างแบบจำลองต่างๆ ที่สามารถใช้งานผ่าน web browser ได้

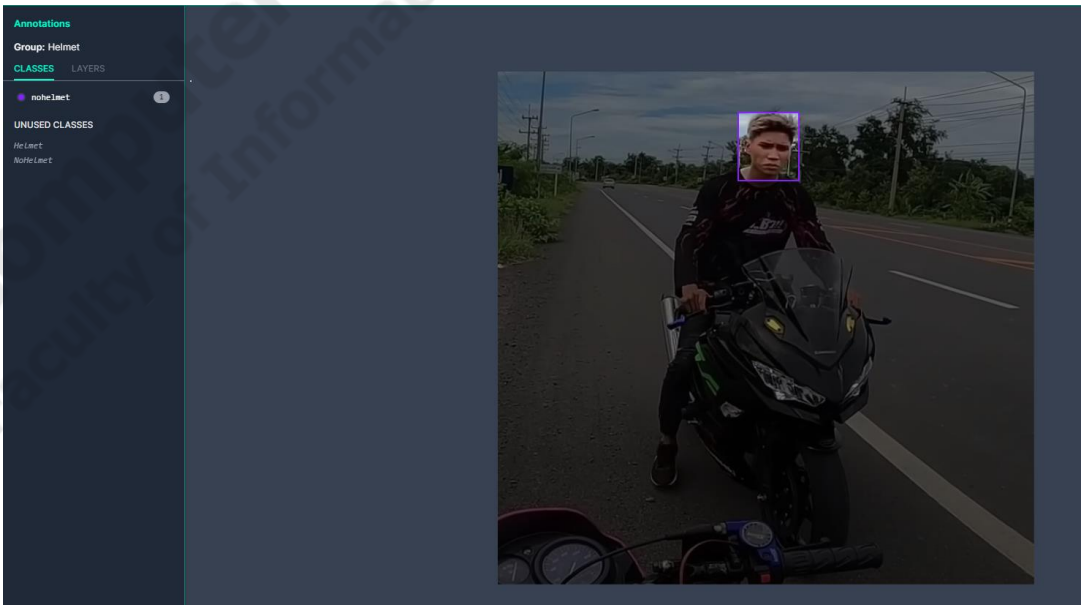
Computer College Department
Faculty of Informatics Mahasarakham University

3.1.2.1 การเตรียมข้อมูลใน Roboflow



ภาพประกอบที่ 3.6 การตีกรอบภาพคลาสสวมหมวกนิรภัย

จากภาพตัวอย่าง 3.6 เป็นการตีกรอบตำแหน่งที่เป็นหมวกนิรภัย และกำหนด class ของ วัตถุที่อยู่ในกรอบ

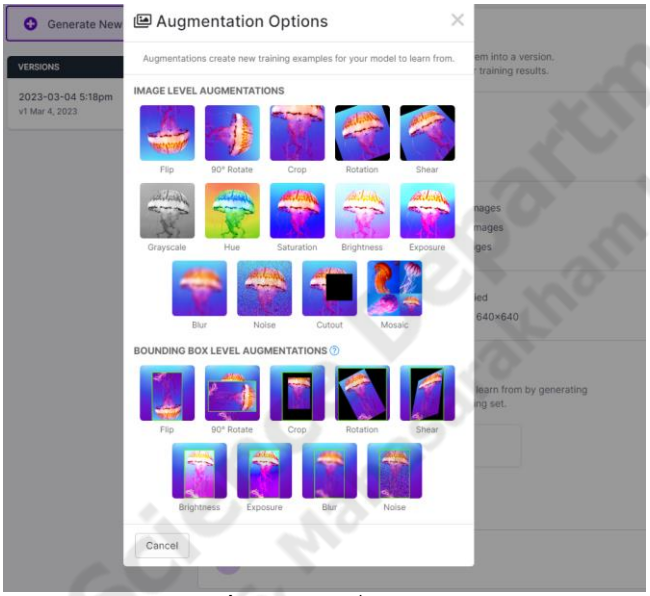


ภาพประกอบที่ 3.7 การตีกรอบภาพไม่สวมหมวกนิรภัย

จากภาพตัวอย่าง 3.7 เป็นการตีกรอบคนไม่สวมหมวกนิรภัย และกำหนด class ของ วัตถุที่อยู่ในกรอบ

3.1.2.2 การเพิ่มปริมาณ dataset

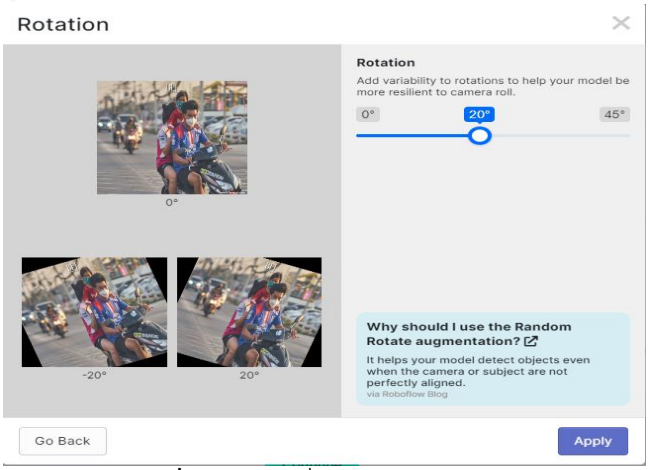
เป็นการเพิ่มรูปแบบของภาพให้มากขึ้นเพื่อเพิ่มความหลากหลายของภาพ เพื่อเพิ่มประสิทธิภาพในการตรวจจับได้ดียิ่งขึ้น



ภาพประกอบที่ 3.8 การเพิ่มประมาณ dataset

3.1.2.3 การเพิ่มปริมาณ data

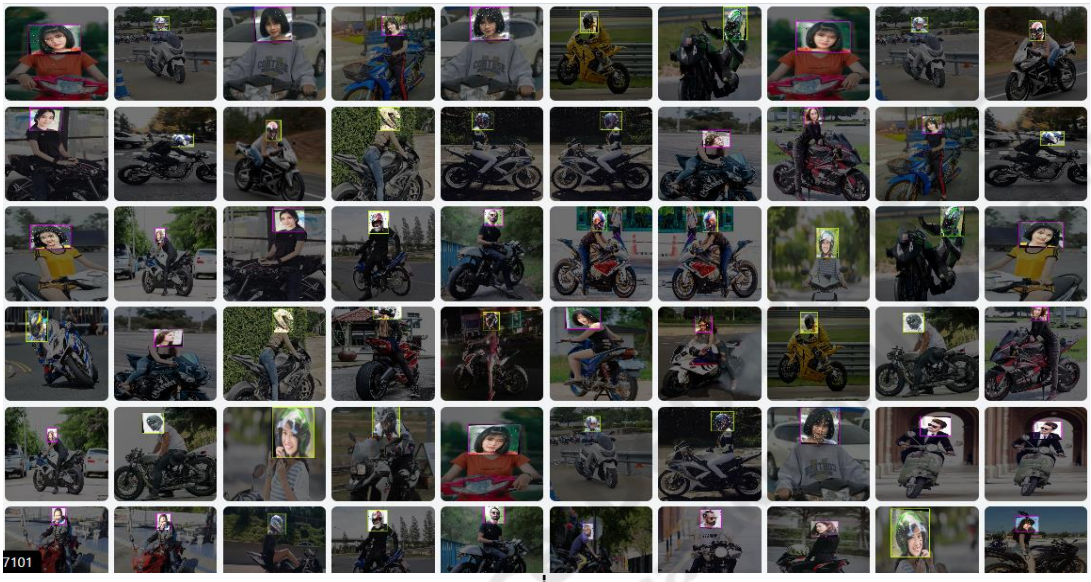
การเพิ่มปริมาณ data เช่น การทำให้ภาพหมุนเพื่อให้เกิดเป็นภาพมุมมองใหม่ๆ



ภาพประกอบที่ 3.9 การเพิ่มปริมาณ dataset Rotation

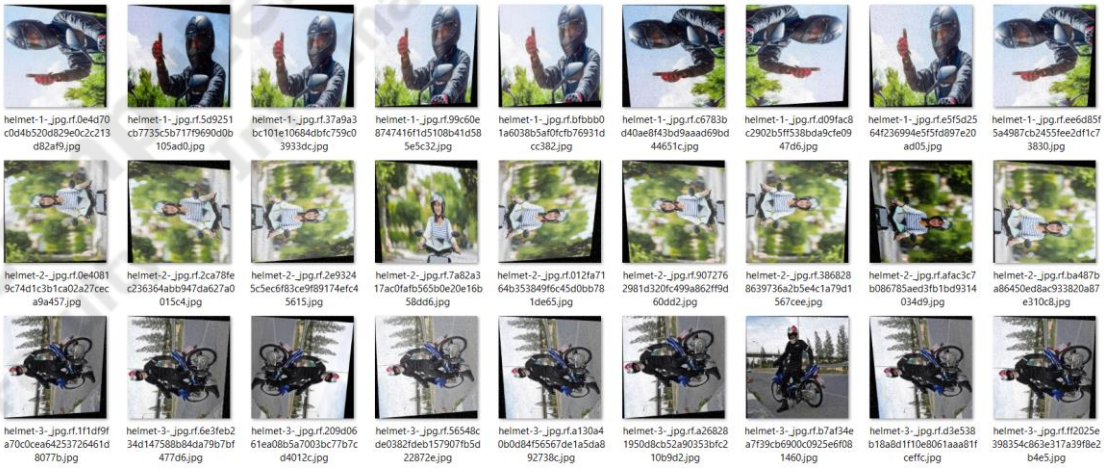
Computer Science Department Faculty of Information Technology Mahasarakham University

เมื่อเพิ่มรูปแบบของ data เสร็จก็ทำการ generate ภาพเพื่อให้ dataset มีรูปแบบที่หลากหลาย

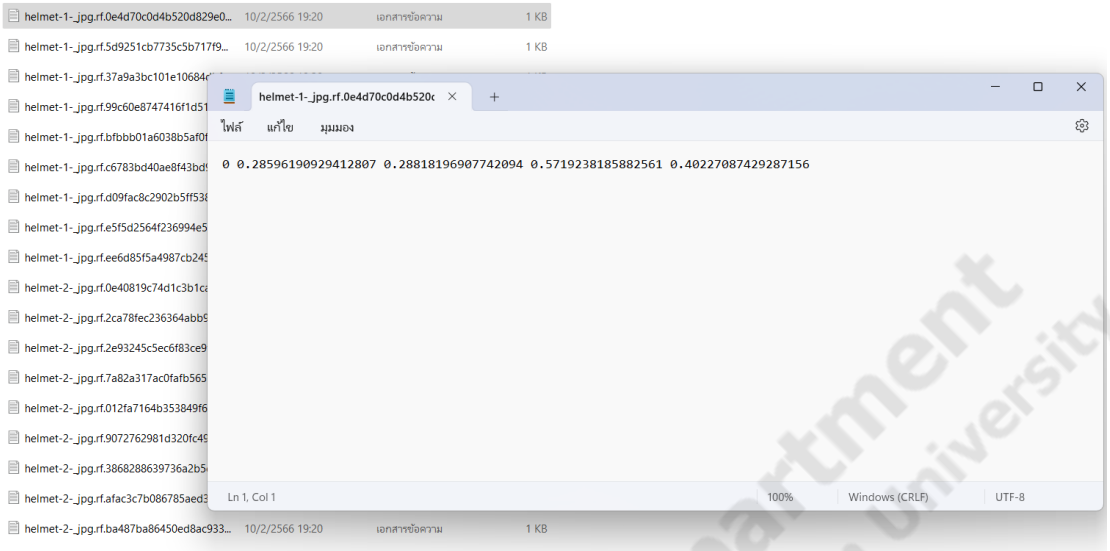


ภาพประกอบที่ 3.10 Dataset

การนำ Dataset ที่เตรียมใน Roboflow มาใช้เมื่อเรา generate รูปภาพเสร็จทำการ Export ไฟล์เราจะได้ไฟล์สองไฟล์ คือ images และ labels



ภาพประกอบที่ 3.11 ตัวอย่างไฟล์รูปภาพ



ภาพประกอบที่ 3.12 ตัวอย่างไฟล์ labels

จากภาพประกอบที่ 3.12 ไฟล์ labels จะประกอบด้วยข้อมูลตัวแรกจะเป็น class ของวัตถุ ตัวที่ 2 ตัวที่3 และตัวที่ 4 คือตำแหน่งของวัตถุนั้น

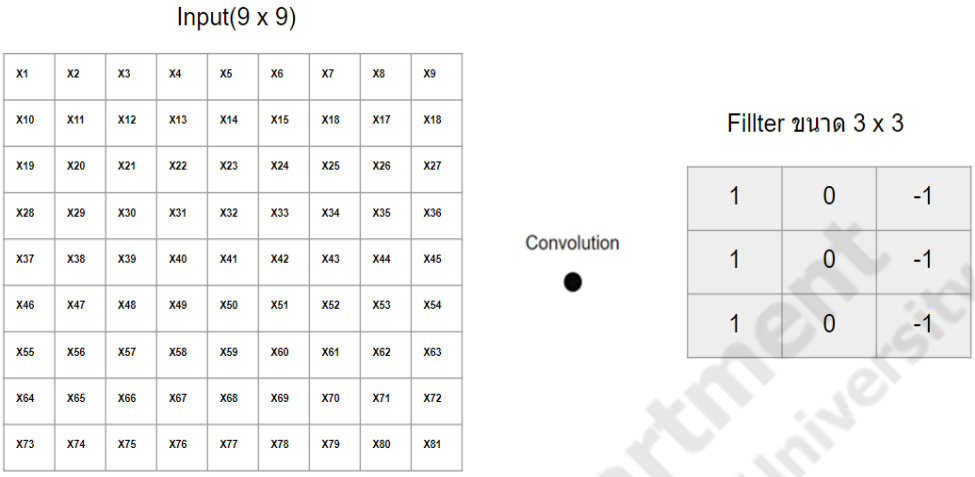
3.2 ขั้นตอนการทำงาน YOLO

3.2.1 การเรียนรู้ข้อมูลด้วย YOLO (Training) มีขั้นตอนการทำงานที่สำคัญคือ การสกัดคุณลักษณะเด่นของภาพด้วยคอนโวโลชันนิวรอลเน็ตเวริ์ก (Convolution neuron networks:CNNS) และสร้างโมเดลด้วย Fully connected layer: FC การทำงานอย่างละเอียดแสดงดังนี้

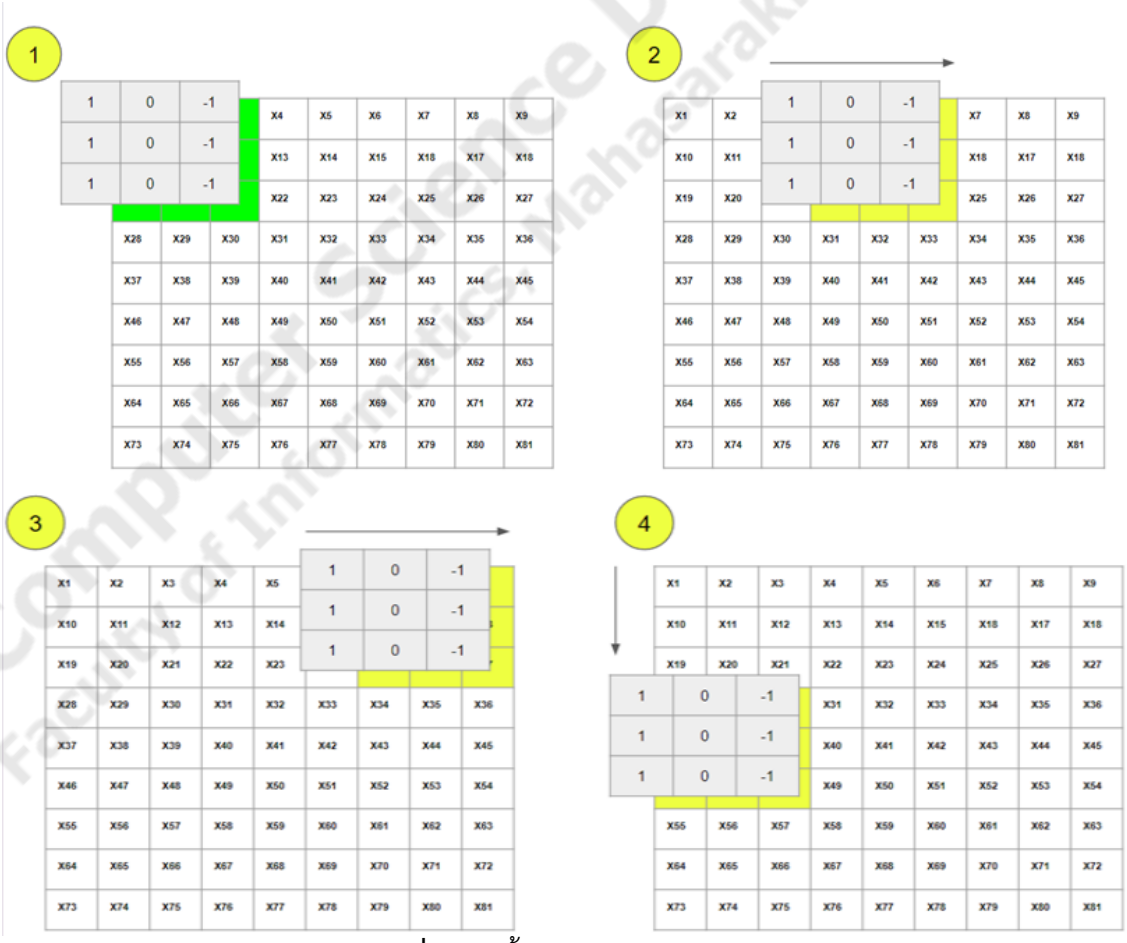
(1) คอนโวโลชันนิวรอลเน็ตเวริ์ก (Convolution neuron networks:CNNS) คือการสกัดคุณลักษณะของภาพ โดยการทำ convolution โดยใช้ Sliding Windows (Filter) มาสแกนภาพเพื่อทำการแยกองค์ประกอบ โดยวิธีดำเนินการ เริ่มจากนำ filter สแกนไปบนภาพ และทำการคำนวณโดยการนำเอาตำแหน่งที่ตรงกันของภาพต้นฉบับ และ filter มาคูณกัน แล้วจึงนำผลรวมของทุกตำแหน่งมาบวกกัน (ตำแหน่งที่ 1 + ตำแหน่งที่ 2 + ... + ตำแหน่งที่ n)

(2) หลักการทำงานของ Convolution ยกตัวอย่าง Input ขนาด 9x9 และขนาดของfillter เท่ากับ 3x3 ตัว filter จะใช้ค่าแสดงดังภาพ 3.13 ซึ่งเป็นค่ามาตรฐาน กำหนด Stride เท่ากับ 3 ให้ขยับ filter ทีละ3 ช่อง ดังนี้

Computer Science Department
Faculty of Informatics Mahasarakham University

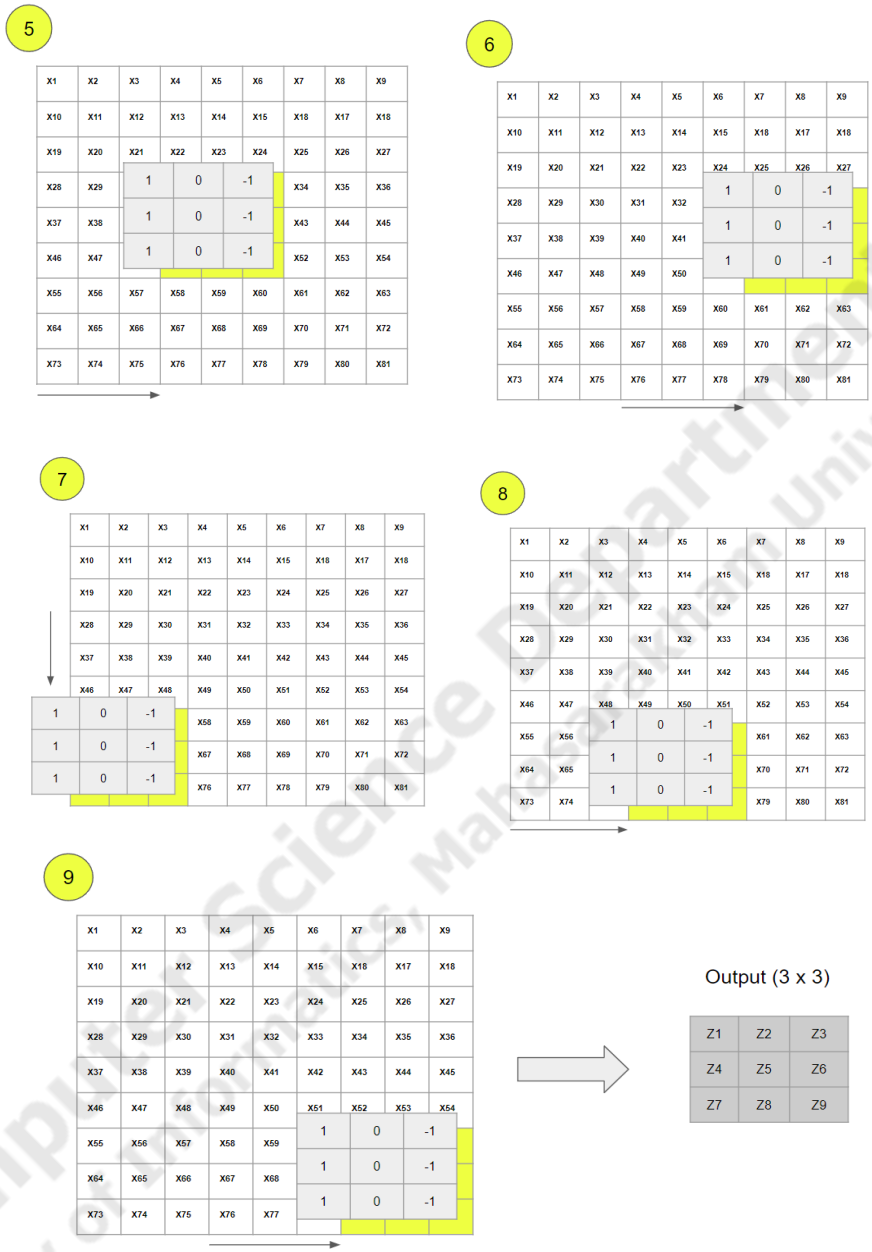


ภาพประกอบที่ 3.13 Convolution Input ขนาด 9x9 และ fillter ขนาด 3x3



ภาพประกอบที่ 3.14 ขั้นตอนการคำนวณ Convolution

Computer Science Department
Faculty of Informatics, Mahasarakham University



ภาพประกอบที่ 3.14 ขั้นตอนการคำนวณ Convolution (ต่อ)

ขั้นตอนการคำนวณคือ หาผลรวมของการคูณระหว่าง Input กับ fillter โดยใช้ fillter ชุด เดิม แล้ว สแกน ไปทั้ง Input จากซ้ายไปขวา และบนลงล่าง ผลลัพธ์ (Z) ที่ได้คือ

$$Z1 = (1)(x1) + (0)(x2) + (-1)(x3) + (1)(x10) + (0)(x11) + (-1)(x12) + (1)(x19) + (0)(x20) + (-1)(x21)$$

$$Z2 = (1)(x4) + (0)(x5) + (-1)(x6) + (1)(x13) + (0)(x14) + (-1)(x15) + (1)(x22) + (0)(x23) + (-1)(x24)$$

$$Z3 = (1)(x7) + (0)(x8) + (-1)(x9) + (1)(x16) + (0)(x17) + (-1)(x18) + (1)(x25) + (0)(x26) + (-1)(x27)$$

$$Z4 = (1)(x28) + (0)(x29) + (-1)(x30) + (1)(x37) + (0)(x38) + (-1)(x39) + (1)(x46) + (0)(x47) + (-1)(x48)$$

$$Z5 = (1)(x31) + (0)(x32) + (-1)(x33) + (1)(x40) + (0)(x41) + (-1)(x42) + (1)(x49) + (0)(x50) + (-1)(x51)$$

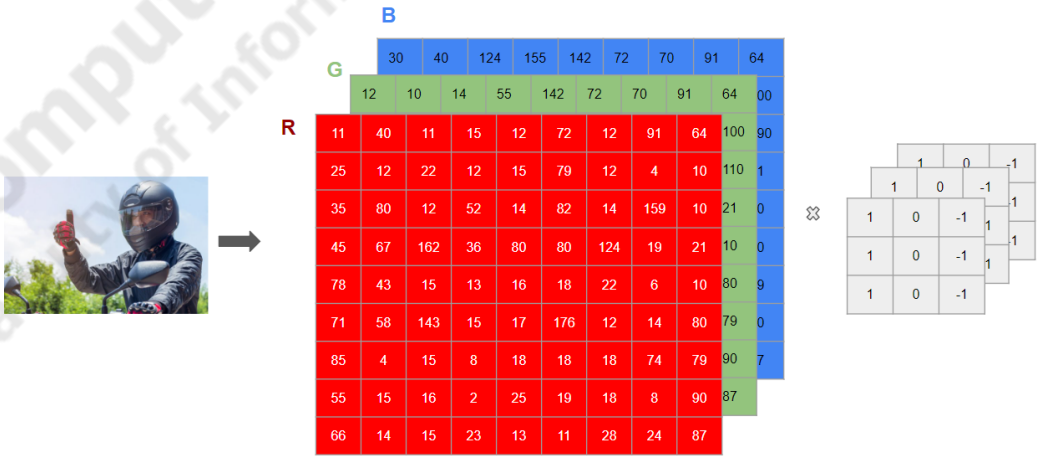
$$Z6 = (1)(x34) + (0)(x35) + (-1)(x36) + (1)(x43) + (0)(x44) + (-1)(x45) + (1)(x52) + (0)(x53) + (-1)(x54)$$

$$Z7 = (1)(x55) + (0)(x56) + (-1)(x57) + (1)(x64) + (0)(x65) + (-1)(x66) + (1)(x73) + (0)(x74) + (-1)(x75)$$

$$Z8 = (1)(x58) + (0)(x59) + (-1)(x60) + (1)(x67) + (0)(x68) + (-1)(x69) + (1)(x76) + (0)(x77) + (-1)(x78)$$

$$Z9 = (1)(x61) + (0)(x62) + (-1)(x63) + (1)(x70) + (0)(x71) + (-1)(x72) + (1)(x79) + (0)(x80) + (-1)(x81)$$

ตัวอย่างวิธีการคำนวณ Convolution Input เป็นภาพสี RGB ขนาด 9 x 9



ภาพประกอบที่ 3.15 ตัวอย่างค่าจากรูปภาพสี RGB นำเข้าก่อนทำ Convolution การทำ แบน R

Computer Science Department
Faculty of Informatics, Mahasarakham University

R

11	40	11	15	12	72	12	91	64
25	12	22	12	15	79	12	4	10
35	80	12	52	14	82	14	159	10
45	67	162	36	80	80	124	19	21
78	43	15	13	16	18	22	6	10
71	58	143	15	17	176	12	14	80
85	4	15	8	18	18	18	74	79
55	15	16	2	25	19	18	8	90
66	14	15	23	13	11	28	24	87

Fillter ขนาด 3 x 3

1	0	-1
1	0	-1
1	0	-1

ภาพประกอบที่ 3.16 ตัวอย่างการทำ Convolution แบน R

$$Z1 = (11\ x\ 1) + (40\ x\ 0) + \big(11\ x\ (-1)\big) + (25\ x\ 1) + (12\ x\ 0) + \big(22\ x\ (-1)\big) + (35\ x\ 1) + (80\ x\ 0) + \big(12\ x\ (-1)\big) = 26$$

$$Z2 = (15\ x\ 1) + (12\ x\ 0) + \big(72\ x\ (-1)\big) + (12\ x\ 1) + (15\ x\ 0) + \big(79\ x\ (-1)\big) + (52\ x\ 1) + (14\ x\ 0) + \big(82\ x\ (-1)\big) = -154$$

$$Z3 = (12\ x\ 1) + (91\ x\ 0) + \big(64\ x\ (-1)\big) + (12\ x\ 1) + (4\ x\ 0) + \big(10\ x\ (-1)\big) + (14\ x\ 1) + (159\ x\ 0) + \big(10\ x\ (-1)\big) = -46$$

$$Z4 = (45\ x\ 1) + (67\ x\ 0) + \big(162\ x\ (-1)\big) + (78\ x\ 1) + (43\ x\ 0) + \big(15\ x\ (-1)\big) + (71\ x\ 1) + (58\ x\ 0) + \big(143\ x\ (-1)\big) = -146$$

$$Z5 = (36\ x\ 1) + (80\ x\ 0) + \big(80\ x\ (-1)\big) + (13\ x\ 1) + (16\ x\ 0) + \big(18x\ (-1)\big) + (15\ x\ 1) + (17\ x\ 0) + \big(176\ x\ (-1)\big) = -210$$

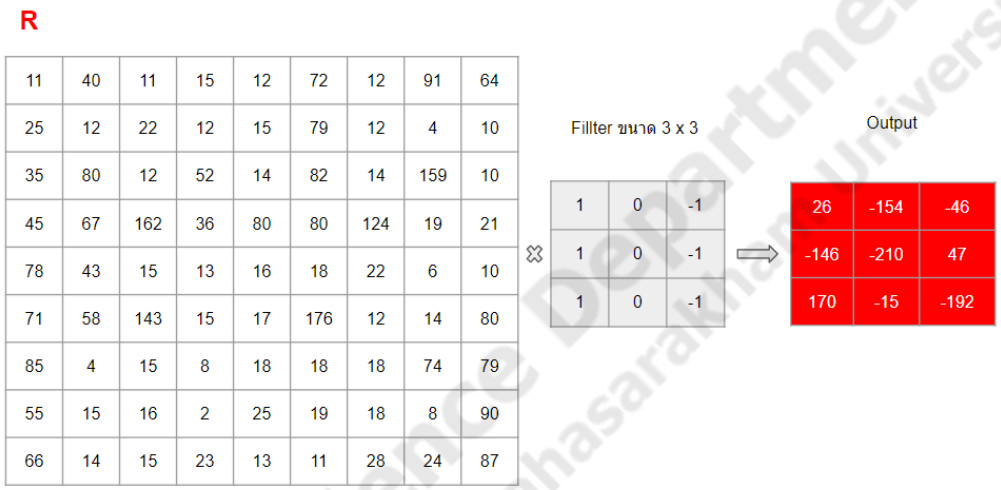
$$Z6 = (124\ x\ 1) + (19\ x\ 0) + \big(21\ x\ (-1)\big) + (22\ x\ 1) + (6\ x\ 0) + \big(10x\ (-1)\big) + (12\ x\ 1) + (14\ x\ 0) + \big(80\ x\ (-1)\big) = 47$$

$$Z7 = (85\ x\ 1) + (4\ x\ 0) + \big(15\ x\ (-1)\big) + (55\ x\ 1) + (15\ x\ 0) + \big(16x\ (-1)\big) + (66\ x\ 1) + (14\ x\ 0) + \big(15\ x\ (-1)\big) = 170$$

Computer Science Department
Faculty of Informatics, Mahasarakham University

$$Z8 = (8\ x\ 1) + (18\ x\ 0) + \big(18\ x\ (-1)\big) + (2\ x\ 1) + (25\ x\ 0) + \big(19x\ (-1)\big) + (23\ x\ 1) + (13\ x\ 0) + \big(11\ x\ (-1)\big) = -15$$

$$Z9 = (18\ x\ 1) + (74\ x\ 0) + \big(79\ x\ (-1)\big) + (18\ x\ 1) + (8\ x\ 0) + \big(90x\ (-1)\big) + (28\ x\ 1) + (24\ x\ 0) + \big(87\ x\ (-1)\big) = -192$$



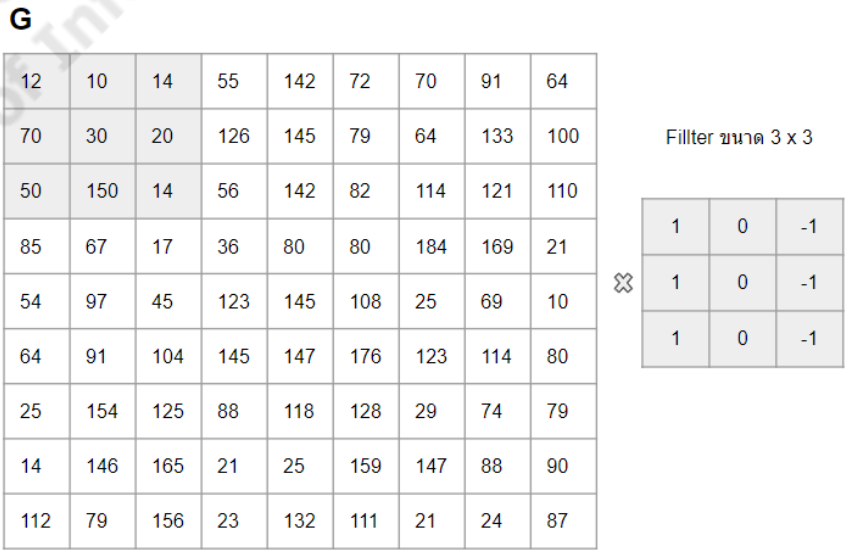
11	40	11	15	12	72	12	91	64
25	12	22	12	15	79	12	4	10
35	80	12	52	14	82	14	159	10
45	67	162	36	80	80	124	19	21
78	43	15	13	16	18	22	6	10
71	58	143	15	17	176	12	14	80
85	4	15	8	18	18	18	74	79
55	15	16	2	25	19	18	8	90
66	14	15	23	13	11	28	24	87

1	0	-1
1	0	-1
1	0	-1

26	-154	-46
-146	-210	47
170	-15	-192

ภาพประกอบที่ 3.17 ตัวอย่างผลลัพธ์การทำ Convolution แบน R

การทำ Convolution แบน G



12	10	14	55	142	72	70	91	64
70	30	20	126	145	79	64	133	100
50	150	14	56	142	82	114	121	110
85	67	17	36	80	80	184	169	21
54	97	45	123	145	108	25	69	10
64	91	104	145	147	176	123	114	80
25	154	125	88	118	128	29	74	79
14	146	165	21	25	159	147	88	90
112	79	156	23	132	111	21	24	87

1	0	-1
1	0	-1
1	0	-1

ภาพประกอบที่ 3.18 ตัวอย่างการทำ Convolution แบน G

Computer Science Department
Faculty of Informatics, Mahasarakham University

$$Z1 = (12\ x\ 1) + (10\ x\ 0) + \big(14\ x\ (-1)\big) + (70\ x\ 1) + (30\ x\ 0) + \big(20\ x\ (-1)\big) + (50\ x\ 1) + (150\ x\ 0) + \big(14\ x\ (-1)\big) = 50$$

$$Z2 = (55\ x\ 1) + (142\ x\ 0) + \big(72\ x\ (-1)\big) + (126\ x\ 1) + (145\ x\ 0) + \big(79\ x\ (-1)\big) + (56\ x\ 1) + (142\ x\ 0) + \big(82\ x\ (-1)\big) = 4$$

$$Z3 = (70\ x\ 1) + (91\ x\ 0) + \big(64\ x\ (-1)\big) + (64\ x\ 1) + (133\ x\ 0) + \big(100\ x\ (-1)\big) + (114\ x\ 1) + (121\ x\ 0) + \big(110\ x\ (-1)\big) = -20$$

$$Z4 = (85\ x\ 1) + (67\ x\ 0) + \big(17\ x\ (-1)\big) + (54\ x\ 1) + (97\ x\ 0) + \big(45\ x\ (-1)\big) + (64\ x\ 1) + (91\ x\ 0) + \big(104\ x\ (-1)\big) = 37$$

$$Z5 = (36\ x\ 1) + (80\ x\ 0) + \big(80\ x\ (-1)\big) + (123\ x\ 1) + (145\ x\ 0) + \big(108x\ (-1)\big) + (145\ x\ 1) + (147\ x\ 0) + \big(176x\ (-1)\big) = -90$$

$$Z6 = (70\ x\ 1) + (16\ x\ 0) + \big(21\ x\ (-1)\big) + (25\ x\ 1) + (69\ x\ 0) + \big(10x\ (-1)\big) + (66\ x\ 1) + (14\ x\ 0) + \big(80\ x\ (-1)\big) = 50$$

$$Z7 = (25\ x\ 1) + (154\ x\ 0) + \big(125\ x\ (-1)\big) + (14\ x\ 1) + (146\ x\ 0) + \big(165\ x\ (-1)\big) + (112\ x\ 1) + (79\ x\ 0) + \big(156x\ (-1)\big) = -9$$

$$Z8 = (88\ x\ 1) + (118\ x\ 0) + \big(128\ x\ (-1)\big) + (21\ x\ 1) + (25\ x\ 0) + \big(159\ x\ (-1)\big) + (23\ x\ 1) + (132x\ 0) + \big(111\ x\ (-1)\big) = -186$$

$$Z9 = (29\ x\ 1) + (74\ x\ 0) + \big(79\ x\ (-1)\big) + (147\ x\ 1) + (88\ x\ 0) + \big(90x\ (-1)\big) + (21\ x\ 1) + (24\ x\ 0) + \big(87\ x\ (-1)\big) = 61$$

Computer Science Department
Faculty of Informatics, Mahasarakham University

G

12	10	14	55	142	72	70	91	64
70	30	20	126	145	79	64	133	100
50	150	14	56	142	82	114	121	110
85	67	17	36	80	80	70	16	21
54	97	45	123	145	108	25	69	10
64	91	104	145	147	176	66	14	80
25	154	125	88	118	128	29	74	79
14	146	165	21	25	159	147	88	90
112	79	156	23	132	111	21	24	87

Fillter ขนาด 3 x 3

1	0	-1
1	0	-1
1	0	-1

Output

50	4	-20
37	-90	50
-9	-186	61

ภาพประกอบที่ 3.19 ตัวอย่างผลลัพธ์การทำ Convolution แบน G

B

30	40	124	155	142	72	70	91	64
50	30	20	126	145	79	64	133	100
90	150	14	56	142	82	164	129	190
80	67	167	36	80	80	84	169	21
55	97	145	123	165	138	25	69	10
65	91	144	145	147	176	23	114	20
125	154	125	88	188	148	29	74	79
145	146	165	21	25	159	147	88	90
112	179	156	23	132	111	21	24	87

Fillter ขนาด 3 x 3

1	0	-1
1	0	-1
1	0	-1

ภาพประกอบที่ 3.20 ตัวอย่างการทำ Convolution แบน B

$$Z1 = (30\ x\ 1) + (40\ x\ 0) + \big(124\ x\ (-1)\big) + (50\ x\ 1) + (30\ x\ 0) + \big(20\ x\ (-1)\big) + (90\ x\ 1) + (150\ x\ 0) + \big(14\ x\ (-1)\big)$$
$$= 12$$

$$Z2 = (155\ x\ 1) + (142x\ 0) + \big(72\ x\ (-1)\big) + (126\ x\ 1) + (145\ x\ 0) + \big(79\ x\ (-1)\big) + (56\ x\ 1) + (142\ x\ 0) + \big(82\ x\ (-1)\big)$$
$$= 104$$

$$Z3 = (70\ x\ 1) + (91\ x\ 0) + \big(64\ x\ (-1)\big) + (64\ x\ 1) + (133\ x\ 0) + \big(100\ x\ (-1)\big) + (164\ x\ 1) + (129\ x\ 0) + \big(190\ x\ (-1)\big)$$
$$= -53$$

Computer Science Department
Faculty of Informatics, Mahasarakham University

$$Z4 = (80\ x\ 1) + (67\ x\ 0) + \big(167\ x\ (-1)\big) + (55\ x\ 1) + (97\ x\ 0) + \big(145\ x\ (-1)\big) + (65\ x\ 1) + (91\ x\ 0) + \big(144\ x\ (-1)\big)$$
$$= -156$$

$$Z5 = (36\ x\ 1) + (80\ x\ 0) + \big(80\ x\ (-1)\big) + (123\ x\ 1) + (165\ x\ 0) + \big(138\ x\ (-1)\big) + (145\ x\ 1) + (147\ x\ 0) + \big(176\ x\ (-1)\big)$$
$$= -43$$

$$Z6 = (84\ x\ 1) + (169\ x\ 0) + \big(21\ x\ (-1)\big) + (25\ x\ 1) + (69x\ 0) + \big(10\ x\ (-1)\big) + (23\ x\ 1) + (114x\ 0) + \big(20\ x\ (-1)\big)$$
$$= 81$$

$$Z7 = (125\ x\ 1) + (154\ x\ 0) + \big(125\ x\ (-1)\big) + (145\ x\ 1) + (146\ x\ 0) + \big(165\ x\ (-1)\big) + (112\ x\ 1) + (179\ x\ 0) + \big(156\ x\ (-1)\big)$$
$$= -8$$

$$Z8 = (88\ x\ 1) + (188\ x\ 0) + \big(148\ x\ (-1)\big) + (21\ x\ 1) + (25\ x\ 0) + \big(159\ x\ (-1)\big) + (23\ x\ 1) + (132\ x\ 0) + \big(111\ x\ (-1)\big)$$
$$= -86$$

$$Z9 = (29\ x\ 1) + (74\ x\ 0) + \big(79\ x\ (-1)\big) + (147\ x\ 1) + (88\ x\ 0) + \big(90x\ (-1)\big) + (21\ x\ 1) + (24\ x\ 0) + \big(87\ x\ (-1)\big)$$
$$= -68$$

B

30	40	124	155	142	72	70	91	64
50	30	20	126	145	79	64	133	100
90	150	14	56	142	82	164	129	190
80	67	167	36	80	80	84	169	21
55	97	145	123	165	138	25	69	10
65	91	144	145	147	176	23	114	20
125	154	125	88	188	148	29	74	79
145	146	165	21	25	159	147	88	90
112	179	156	23	132	111	21	24	87

Fillter ขนาด 3 x 3

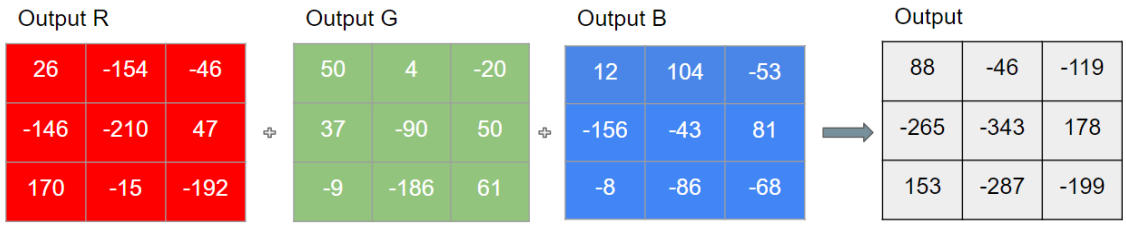
1	0	-1
1	0	-1
1	0	-1

Output

12	104	-53
-156	-43	81
-8	-86	-68

ภาพประกอบที่ 3.21 ตัวอย่างผลลัพธ์การทำ Convolution แบน B

Computer Science Department
Faculty of Informatics, Mahasarakham University



ภาพประกอบที่ 3.22 ตัวอย่างผลลัพธ์การทำ Convolution

หลักจากทำขั้นตอน Convolution เสร็จแล้วในส่วนต่อไปจะเป็นการทำ RELU เพื่อให้ค่าที่ได้อยู่ในช่วงที่ต้องการ และ Max Pooling Layer ซึ่งเป็นขั้นตอนในการลดขนาดของข้อมูล แต่ยังคงรายละเอียดของ input ไว้ ส่งผลให้ความเร็วในการคำนวณเพิ่มขึ้นและหลีกเลี่ยงการเกิด overfitting และหลังจากนั้นจะนำไปทำ Flatten เพื่อเปลี่ยนข้อมูลที่ได้เป็น Vector 1 มิติ

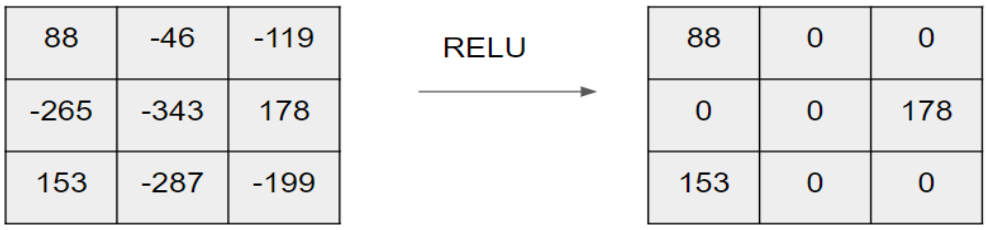
หลักการทำงานของ RELU คือการลบค่าลบทั้งหมดออกจาก Convolution ค่าบวกทั้งหมดยังคงเหมือนเดิม แต่ค่าลบทั้งหมดจะเปลี่ยนเป็นศูนย์

สูตรคำนวณ RELU

$$f(x) = \begin{cases} x, x > 0 \\ 0, x \leq 0 \end{cases}$$

ตาราง 3.1 ตัวอย่าง RELU

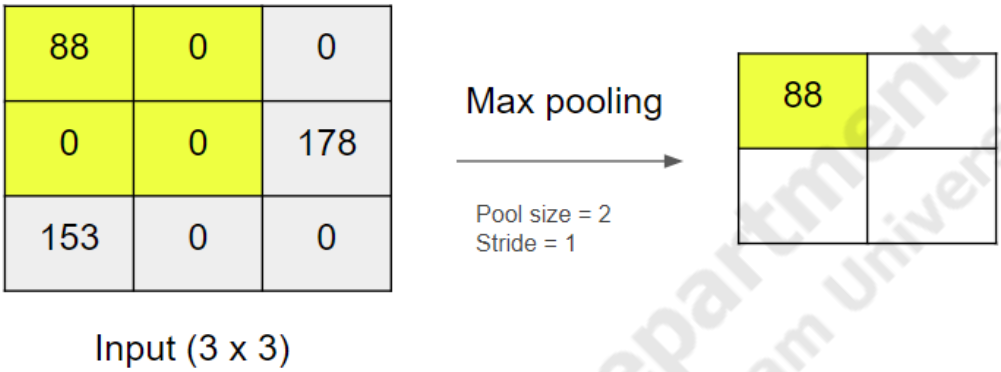
x	f(x) = x	f(x)
-3	f(x) = 0	0
0	f(x) = 0	0
1	f(x) = 1	1



ภาพประกอบที่ 3.23 ตัวอย่างการทำ RELU

Computer Science Department
Faculty of Informatics, Mahasarakham University

หลักการทำงานของ Max Pooling Layer ทำการกำหนดขนาดของ filter และกำหนดค่า stride จากนั้นวางฟิลเตอร์ที่มุมซ้ายสุดแล้วนำค่าที่มากที่สุดที่อยู่ในบริเวณของ filter มาเป็นค่าของรูปใหม่ จากนั้นขยับ filter ตามค่า stride เช่นค่า stride = 1 ก็ให้ขยับ filter ทีละ 1 ช่อง ดังนี้



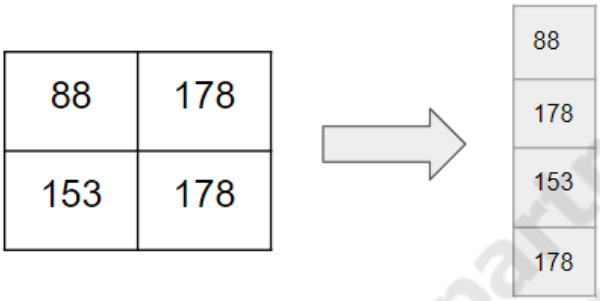
ภาพประกอบที่ 3.24 ตัวอย่างการทำ Max Pooling Layer

ภาพประกอบที่ 3.25 ตัวอย่างการทำ Max Pooling Layer

Computer Science Department
Faculty of Informatics, Mahasarakham University

จุดประสงค์ของการทำ Pooling คือ เพื่อลดขนาดของข้อมูล ทำให้โมเดลไม่จำเป็นต้องเรียนรู้ค่า weight จำนวนมากเกินไป

(1) หลักการทำงานของ Flatten การทำ Flatten คือการทำให้ค่าข้อมูลที่เป็นภาพประกอบที่ได้จาก Maxpooling 2 มิติ กลายเป็นข้อมูล Vector แบบ 1 มิติ เพื่อนำไปใช้ในขั้นตอนต่อไป



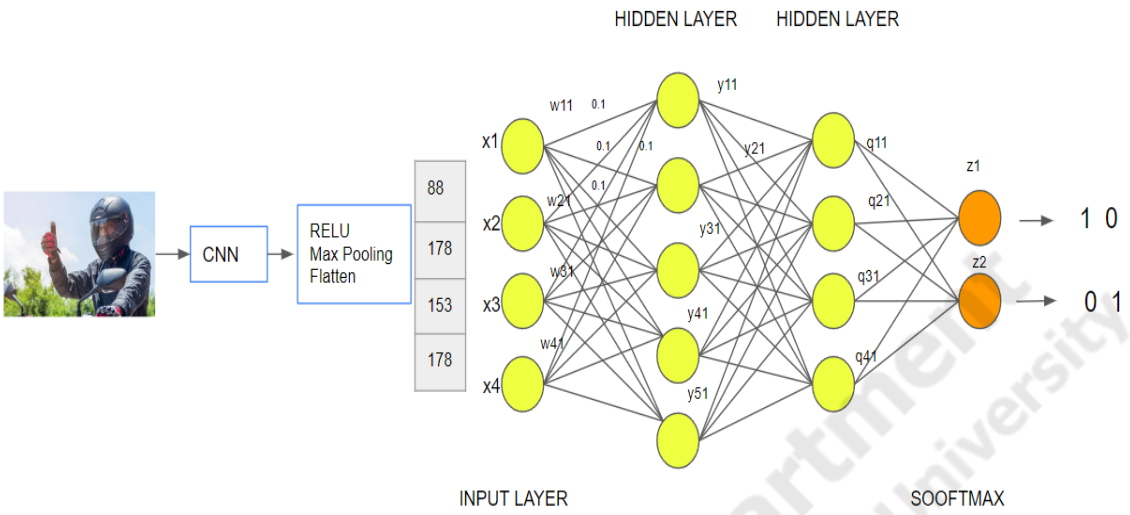
ภาพประกอบที่ 3.26 ตัวอย่างการทำ Flatten

ผลลัพธ์ CNN จะได้คุณลักษณะเด่นของภาพแต่ละภาพออกเป็น n*1 เวกเตอร์แสดงดังภาพประกอบที่ 3.27

ภาพประกอบที่ 3.27 ตัวอย่างผลลัพธ์ของ CNN

(2) การสร้างโมเดลด้วย FC (Fully-Connected Layer) หลักการทำงานของ Fully-Connected Layer เป็นชั้นที่เชื่อมโยงระหว่าง Feature Map และ Output แบบสมบูรณ์ นั่นหมายถึงทุกๆ Neuron ที่อยู่ในชั้นสุดท้ายของ Feature Map จะถูกนำไปเปลี่ยนรูป (Reshape) หรือเรียกว่า Flatten เพื่อส่งไปคำนวณต่อในชั้นถัดไป

Computer Science Department
Faculty of Informatics, Mahasarakham University



ภาพประกอบที่ 3.28 ตัวอย่างการทำงานของ Fully-Connected Layer

Softmax

Softmax Function เป็นฟังก์ชันที่จะแปลงคะแนนผลลัพธ์ของชั้นสุดท้ายในแต่ละโหนด ของโครงข่ายประสาทเทียม เป็นค่าความน่าจะเป็นตามสัดส่วนของคลาส คลาสไหนมีความน่าจะเป็นสูงสุดโมเดลจะเลือกคลาสนั้นเป็นคำตอบ

$$\sigma(z_i) = \frac{e^{Z_i}}{\sum_{j=1}^{k} e^{Z_i}}$$

โดยที่

Z คือ อินพุตของฟังก์ชัน softmax ซึ่งเป็นค่าของ pixel value(element) คูณกับ weight จนครบทุกโหนดแล้วนำไปบวกกับ bias จะได้ออกมาเป็นค่าโหนดของแต่ละคลาส

Z_i คือ ค่าของแต่ละโหนด

e^{Z_i} คือ ค่ามาตรฐานเอกซ์โพเนนเชียล (e) ที่นำมายกกำลังโดย Z_i แต่ละตัว

$\sum_{j=1}^{k} e^{Z_i}$ คือ ผลรวมของ e^{Z_i} ทุกตัวยกตัวอย่างการคำนวณให้ Z_i = (5.0, 3.0)

ขั้นตอนที่ 1 หาค่า e^{Z_i} แต่ละค่า โดยค่า e มีค่าประมาณ 2.71828

$e^{5.0}$ = 148.413159103

$e^{3.0}$ = 20.0855369232

ขั้นตอนที่ 2 หาค่า $\sum_{j=1}^{k} e^{z_i}$

$\sum_{j=1}^{k} e^{z_i}$ = (148.413159103+20.0855369232)

=168.498696026

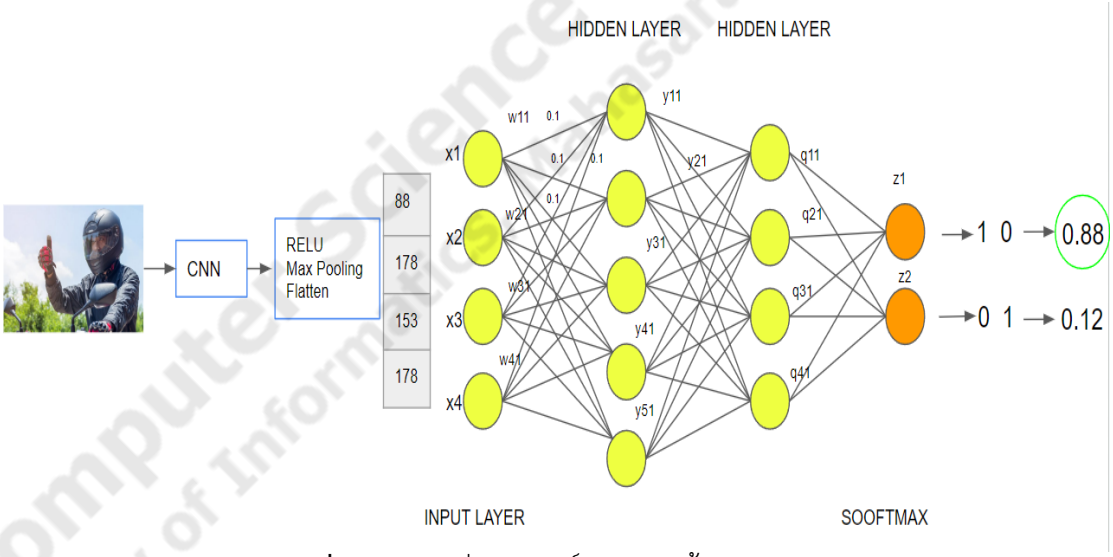
ขั้นตอนที่ 3 หาค่า Soft Max โดยแทนค่าในสมการ จะได้

$$\sigma(5.0) = \frac{148.413159103}{168.498696026} = 0.88079707797$$

$$\sigma(3.0) = \frac{20.0855369232}{168.498696026} = 0.11920292202$$

ซึ่งเมื่อนำค่า Soft Max ของทุกค่ามารวมกัน จะมีค่าเท่ากับ 1

โมเดลจะเลือกคลาสที่มีความน่าจะเป็นสูงสุดเป็นคำตอบ



ภาพประกอบที่ 3.29 ตัวอย่างผลลัพธ์หลังจากเข้า function Softmax

3.3 ขั้นตอนการค้นหาหมวกนิรภัย (Detection)

ในการค้นหาหมวกนิรภัยจะยกตัวอย่างภาพนำเข้า 2 แบบคือ ภาพนำเข้าที่ไม่สวมหมวกนิรภัย และ ภาพนำเข้าที่สวมหมวกนิรภัยและไม่สวมหมวกนิรภัย โดยมีรายละเอียดแสดงตามลำดับดังนี้

Computer Science Department
Faculty of Informatics Mahasarakham University

3.3.1 ตัวอย่าภาพนำเข้าไม่สวมหมวกนิรภัยแสดงขั้นตอนการคำนวณดังนี้

กำหนดขนาดภาพ 640 x 640 และแบ่งภาพเป็นตาราง 3 x 3

ภาพประกอบที่ 3.30 ตัวอย่างการแบ่งตาราง 3x3

3.3.2 กำหนดค่า y เพื่อเก็บคำตอบ

กำหนดค่า y เพื่อเก็บคำตอบ มีค่าเท่ากับ S x S x A x (5+จำนวน class) โดยให้ A=2 ซึ่ง A คือ จำนวน anchor ภายในแต่ละ grid ในตัวอย่างกำหนดจำนวน class เท่ากับ 2 คือ helmet และ nohelmet ดังนั้น y=3x3x2x(5+2) หรือ 3x3x2x7 การเก็บข้อมูลแสดงดังนี้

$$y = \begin{bmatrix} p(helmet) \\ bx \\ by \\ bh \\ bw \\ 1 \\ 0 \\ p(nohelmet) \\ bx \\ by \\ bh \\ bw \\ 0 \\ 1 \end{bmatrix}$$

โดย p คือ ค่าความน่าจะเป็น (bx,by,bh,bw) คือตำแหน่งของวัตถุ และ (0,1)(1,0) คือ class ของวัตถุนั้น

3.3.3 ขนาดของ anchor ในแต่ละตารางจะหาได้จาก k-mean หากแต่ละตารางต้องการ 2 anchor แสดงดังภาพประกอบที่ 3.31

3.3.3.1 K-mean

K-Means เป็นวิธีที่นิยมใช้ในการแบ่งกลุ่มข้อมูล โดยเปรียบเทียบความคล้ายคลึงของข้อมูล กับจุดศูนย์กลางของแต่ละคลัสเตอร์ (Cluster) หรือค่าเฉลี่ย (Mean) เป็นการแบ่งแบบ Partitional clustering ด้วยการแบ่งข้อมูลออกเป็นส่วน ตามจำนวนกลุ่มที่ระบุ มี 4 ขั้นตอน

(1) mark - กำหนดจำนวนกลุ่ม K กลุ่ม และกำหนดจุดศูนย์กลางเริ่มต้นจำนวน K จุดด้วยการสุ่ม

(2) distance - นำวัตถุทั้งหมดจัดเข้ากลุ่มที่มีจุดศูนย์กลางที่อยู่ใกล้วัตถุนั้นมากที่สุด โดย คำนวณจากการวัดระยะห่างระหว่างจุดที่น้อยที่สุด

(3) center - คำนวณจุดศูนย์กลาง K จุดใหม่ โดยหาจากค่าเฉลี่ยทุกวัตถุที่อยู่ในกลุ่ม

(4) repeat - ทำซ้ำในข้อ 2 จนกระทั่งจุดศูนย์กลางไม่เปลี่ยนแปลงสมการสูตร Euclidean

$$dist = \sqrt{\sum_{k=1}^{n}(P_k - q_k)^2}$$

หรือ

$$\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

ตัวอย่างที่ K-mean นำภาพมาจากกริดที่ 2 โดยทำการสุ่ม C1 = 96 ,C2 = 201

20	190	195	178	168
78	96	10	160	170
90	46	150	190	199
28	42	57	201	241
25	15	17	214	216

ภาพประกอบที่ 3.31 ภาพตัวอย่างที่นำมาทำ k-mean

Computer Science Department
Faculty of Informatics, Mahasarakham University

จุดที่ 1

20	190	195	178	168
78	96	10	160	170
90	46	150	190	199
28	42	57	201	241
25	15	17	214	216

ภาพประกอบที่ 3.32 ตัวอย่างการจัดกลุ่มข้อมูลตัวอย่างที่ 1

$$dist = \sqrt{\sum_{k=1}^{n}(P_k - q_k)^2} \quad \text{หรือ} \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

$$dist1 = \sqrt{(20 - 96)^2}$$
$$= 76$$
$$dist2 = \sqrt{(20 - 201)^2}$$
$$= 181$$

ดังนั้น 20 จะถูกจัดในกลุ่มข้อมูลเดียวกับ C1=96 เนื่องจากมีระยะห่างที่น้อยกว่า C2=201 แสดงดังภาพที่ 3.33

20	190	195	178	168
78	96	10	160	170
90	46	150	190	199
28	42	57	201	241
25	15	17	214	216

ภาพประกอบที่ 3.33 ตัวอย่างผลลัพธ์การจัดกลุ่มข้อมูลตัวอย่างที่ 1

Computer Science Department
Faculty of Informatics Mahasarakham University

จุดที่ 2

20	190	195	178	168
78	96	10	160	170
90	46	150	190	199
28	42	57	201	241
25	15	17	214	216

ภาพประกอบที่ 3.34 ตัวอย่างการจัดกลุ่มข้อมูลตัวอย่างที่ 2

$$dist = \sqrt{\sum_{k=1}^{n}(P_k - q_k)^2} \quad \text{หรือ} \quad \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

$$dist1 = \sqrt{(190 - 96)^2}$$
$$= 94$$
$$dist2 = \sqrt{(190 - 201)^2}$$
$$= 11$$

ดังนั้น 190 จะถูกจัดในกลุ่มข้อมูลเดียวกับ C2 = 201 เนื่องจากมีระยะห่างที่น้อยกว่า 96 แสดงดังภาพที่ 3.35

20	190	195	178	168
78	96	10	160	170
90	46	150	190	199
28	42	57	201	241
25	15	17	214	216

ภาพประกอบที่ 3.35 ตัวอย่างผลลัพธ์การจัดกลุ่มข้อมูลตัวอย่างที่ 2

Computer Science Department
Faculty of Informatics, Mahasarakham University

ภาพประกอบที่ 3.36 ตัวอย่างผลลัพธ์การจัดกลุ่มข้อมูล

เมื่อคำนวณต่อไปเรื่อยๆจนครบทุกพิกเซลแล้ว จะทำการคำนวณจุดศูนย์กลาง ใหม่ โดยหาจากค่าเฉลี่ยทุกตัวที่อยู่ในกลุ่ม จนกว่าค่าจุดศูนย์กลางจะไม่เปลี่ยนแปลง

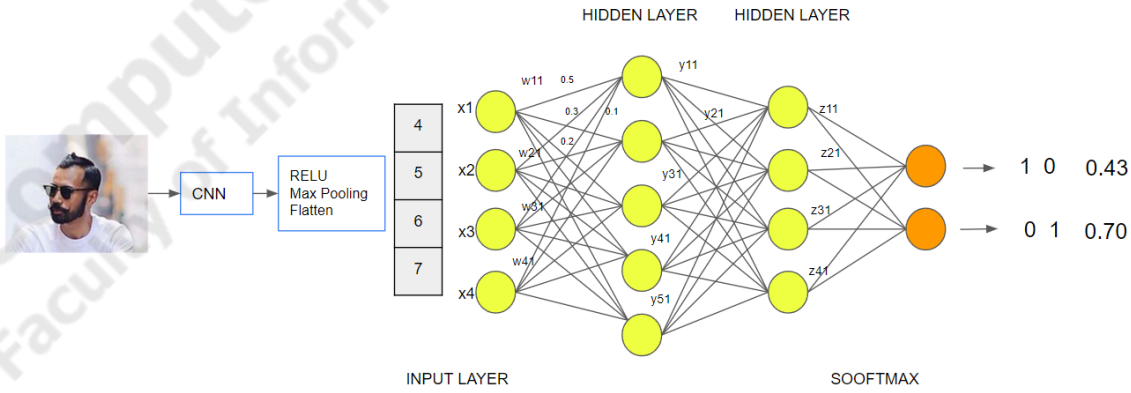
ภาพประกอบที่ 3.37 ตัวอย่างผลลัพธ์การจัดกลุ่มข้อมูล

Computer Science Department
Faculty of Informatics, Mahasarakham University

ในกริดอื่นๆก็ทำเช่นกัน จะได้ผลลัพธ์แสดงดัง ภาพประกอบที่ 3.38

ภาพประกอบที่ 3.38 การแบ่งกลุ่มข้อมูลในแต่ละช่อง โดยแต่ละช่องจะมี 2 anchor

3.3.4 นำแต่ละ boundary block สกัด Features ด้วย CNN และเข้าโมเดลเพื่อตรวจสอบว่าอยู่ในคลาสใดตัวอย่างเช่นนำกริดที่2 block สีน้ำเงิน



$$Yi = \sum WiXi$$

Y11 = (w11*x1)+(w21*x2)+(w31*x3)+(w41*x4)
= (0.5*4)+(0.3*5)+(0.2*6)+(0.1*7)
= 5.4

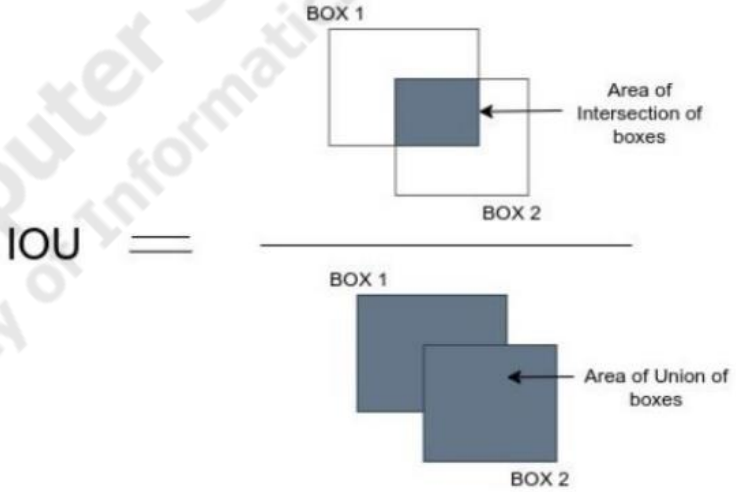
ภาพประกอบที่ 3.39 นำแต่ละ boundary block สกัด Features ด้วย CNN และเข้าโมเดล

ใช้ class ที่ค่า probity ที่มากที่สุด ทำกรอบสีแดง เหมือนกัน และ ทำทั้งหมดของภาพการเก็บค่า Y ของ กริดที่ 2 หลังจากเข้าโมเดล เราจะทราบว่าวัตถุที่อยู่ในกรอบนั้นเป็นคลาสอะไร

$$y = \begin{bmatrix} P = ? \\ bx \\ by \\ bh \\ bw \\ 0 \\ 1 \\ P = ? \\ bx \\ by \\ bh \\ bw \\ 0 \\ 1 \end{bmatrix}$$

จากนั้นต้องหาค่า probity โดยหาจากค่า IOU

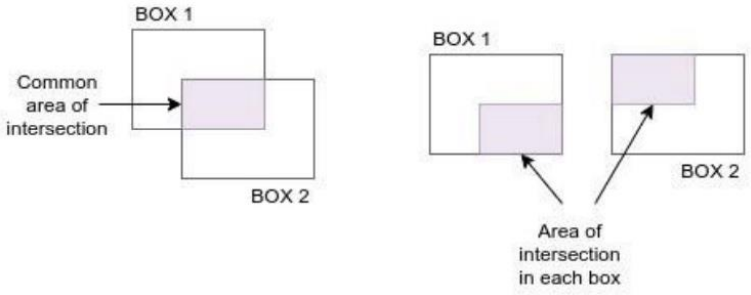
3.3.5 IOU



ภาพประกอบที่ 3.40 แผนภาพสูตรการคำนวณ IOU

พื้นที่ทั้งหมดที่ครอบคลุมโดยทั้งสองกล่องจะทราบว่าส่วนของจุดตัดนั้นครอบคลุมอยู่ในทั้งสองกล่องกล่าวคือพื้นที่ของจุดตัดจะรวมอยู่ใน area_box1 และ area_box2

Computer Science Department
Faculty of Informatics, Mahasarakham University



ภาพประกอบที่ 3.41 การแสดงพื้นที่ที่ครอบคลุมทั้งสองกรอบ

เนื่องจากเราต้องการคำนวณพื้นที่ส่วนกลางของจุดตัดเพียงครั้งเดียวเราจึงสามารถลบพื้นที่จุดตัดที่เราคำนวณออกจากพื้นที่ทั้งหมดของสองกล่อง สามารถคำนวณได้ดังสมการ 1 และ 2 ดังนี้

$$\text{areaunio} = \text{area}_{\text{box1}} + \text{area}_{\text{box2}} - \text{area}_{\text{intersection}} \quad (1)$$

$$\text{IOU} = \frac{\text{areaintersecion}_{\text{intersection}}}{\text{areaunion}} \quad (2)$$

(1) สูตรคำนวนพื้นที่ของการรวมกันของกล่อง

(2) สูตรคำนวณ IOU จากสูตรทั้งหมดสามารถสรุปออกมาเป็นตัวอย่างได้ตามตัวอย่างให้เข้าใจได้ดังต่อไปนี้

ตัวอย่างการคำนวณ IOU ของกริดที่ 2 กรอบสีน้ำเงิน

ผลเฉลย

1	1	1	1	1	1
1	0	1	1	1	1
1	0	1	1	1	1
0	1	1	1	1	1
1	0	1	1	1	1
1	0	1	1	1	1
0	1	1	1	1	1
0	1	1	1	1	1
1	1	1	1	1	1

box ตัวอย่าง

1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1
0	1	1	1	1	1
1	0	1	1	1	1
1	0	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1

ภาพประกอบที่ 3.42 ตัวอย่างข้อมูลของกรอบสีน้ำเงินและผลเฉลย

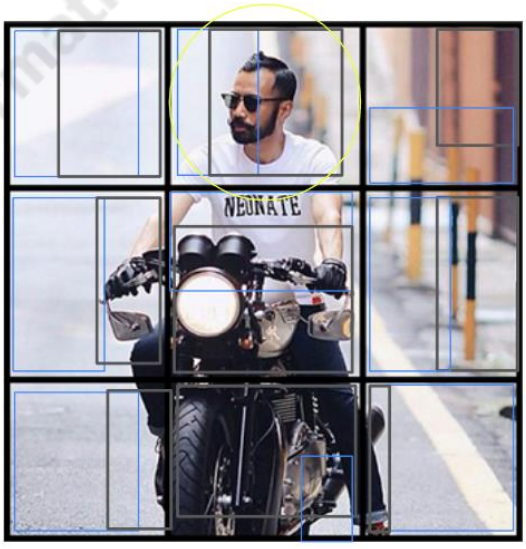
Computer Science Department
Faculty of Informatics, Mahasarakham University

1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1
0	1	1	1	1	1
1	0	1	1	1	1
1	0	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1

ภาพประกอบที่ 3.43 Intersection area

$$\text{areaunion} = 54 + 54 - 47$$
$$= 61$$
$$IOU = \frac{47}{61}$$
$$= 0.77$$

หลังจากการทำ IOU เราจะได้ค่า probity ทำกรอบสีแดง เหมือนกัน และ ทำทั้งหมดของภาพ



ภาพประกอบที่ 3.44 ตัวอย่างการเก็บค่าคนไม่สวมหมวกนิรภัย

การเก็บค่า Y ของ กริดที่ 2 หลังจากทำ IOU จะได้ค่า probity

Computer Science Department
Faculty of Informatics, Mahasarakham University

$$Y = \begin{bmatrix} P = 0.77 \\ 0.29 \\ 0.29 \\ 0.57 \\ 0.40 \\ 0 \\ 1 \\ P = 0.81 \\ 0.21 \\ 0.32 \\ 0.62 \\ 0.50 \\ 0 \\ 1 \end{bmatrix}$$

หลังจากหาค่า probity จาก IOU ทั้งหมด แสดงดังภาพตัวอย่างที่ 3.45

ภาพประกอบที่ 3.45 ตัวอย่างหลังจากการหาค่า probity ทั้งหมด

เมื่อหาค่า Y ได้ครบแล้วเราจะกำหนดว่าถ้าค่า probity น้อยกว่า 0.7 จะไม่นำค่านั้นมาคำนวณต่อ ดังภาพประกอบที่ 3.46

ภาพประกอบที่ 3.46 ตัดข้อมูลที่มีความน่าจะเป็นน้อยกว่าที่กำหนดออก

3.3.6 non-max suppression

โดยวิธีการนี้ จะใช้เมื่อมี box ที่ ซ้อนทับกัน โดยจะทำการคำนวณค่า IOU ซึ่งเป็นหาอัตราส่วน ของ Intersection area และ Union area ซึ่งเราจะกำหนดว่า ถ้าค่า IOU มากกว่า 0.5 เมื่อค่า IOU สูง แสดงมี box กำลังซ้อนกันอยู่เราจะใช้วิธีการ non-max suppression ซึ่งจะสนใจ box ที่มีความน่าจะเป็นสูงสุด ถ้าค่า IOU ต่ำกว่า 0.5 เราจะใช้คำตอบของทั้งสอง box

ภาพประกอบที่ 3.47 ตัวอย่างกริดที่ 2

Computer Science Department
Faculty of Informatics Mahasarakham University

สมการการคำนวณ IOU

$$IOU = \frac{Intersection\ area}{Union\ area}$$

ตัวอย่างการคำนวน IOU

ในการคำนวณค่า IOU ของขั้นตอนนี้จะเป็นการหาอัตราส่วน ของ Intersection area และ Union area ของ box ที่อยู่ในกริดตัวอย่าง

Box สีน้ำเงิน

1	1	1	1	1	1
1	1	1	1	1	0
1	1	1	1	1	0
1	1	1	1	1	0
1	1	1	1	1	0
1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1

Box สีแดง

1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1

ภาพประกอบที่ 3.48 ภาพตัวอย่างข้อมูลของ box สีแดงและ box สีน้ำเงิน

Intersection area

1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1

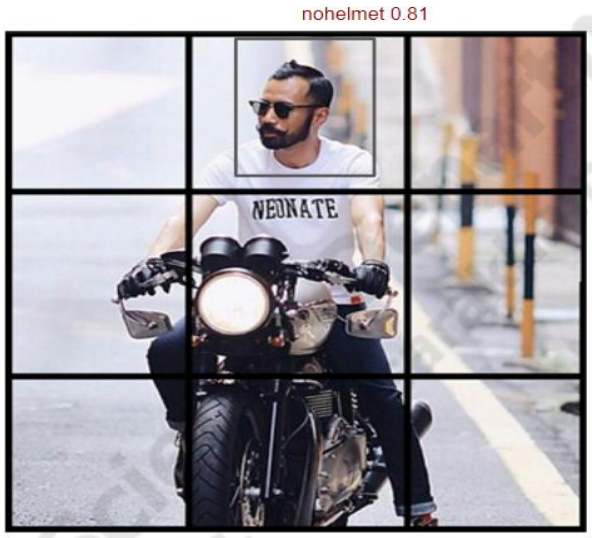
ภาพประกอบที่ 3.49 Intersection area ของ box สีแดงและ box สีน้ำเงิน

$$\begin{aligned} \text{areaunion} &= 54 + 54 - 50 \\ &= 58 \end{aligned}$$

Computer Science Department
Faculty of Informatics Mahasarakham University

$$\text{IOU} = \frac{50}{58}$$
$$= 0.86$$

จากการคำนวณ IOU มีค่าเท่ากับ 0.86 ซึ่งมากกว่าค่าที่เรากำหนดคือ 0.5 เราจึงทราบได้ว่า box ทั้งสองกำลังครอบวัตถุเดียวกันอยู่ ทำให้เราจะใช้วิธีการ non-max suppression ซึ่งจะสนใจ box ที่มีความน่าจะเป็นสูงสุด



ภาพประกอบที่ 3.50 ตัวอย่างกรอบที่มีความน่าจะเป็นสูงสุด

ตัวอย่างภาพนำเข้าที่กำหนดขนาดภาพ 640 x 640 และแบ่งภาพเป็นตาราง 3 x 3 สวมหมวกนิรภัยและไม่สวมหมวกนิรภัยแสดงขั้นตอนดังภาพประกอบที่ 3.51

ภาพประกอบที่ 3.51 ตัวอย่างการแบ่งตาราง 3x3

3.3.7 กำหนดค่า y เพื่อเก็บคำตอบ

กำหนดค่า y เพื่อเก็บคำตอบ มีค่าเท่ากับ S x S x A x (5+จำนวน class) โดยให้ A=2 ซึ่ง A คือ จำนวน anchor ภายในแต่ละ grid ในตัวอย่างกำหนดจำนวน class เท่ากับ 2 คือ helmet และ nohelmet ดังนั้น y=3x3x2x(5+2) หรือ 3x3x2x7 การเก็บข้อมูลแสดงดังนี้

$$y = \begin{bmatrix} p(helmet) \\ bx \\ by \\ bh \\ bw \\ 1 \\ 0 \\ p(nohelmet) \\ bx \\ by \\ bh \\ bw \\ 0 \\ 1 \end{bmatrix}$$

โดย p คือ ค่าความน่าจะเป็น (bx,by,bh,bw) คือตำแหน่งของวัตถุ และ (0,1) (1,0) คือ class ของวัตถุนั้นขนาดของ anchor ในแต่ละตารางจะหาได้จาก k-mean หากแต่ละกริดต้องการ 2 anchor แสดงดังภาพประกอบที่ 3.52

3.3.7.1 K-mean เป็นวิธีที่นิยมใช้ในการแบ่งกลุ่มข้อมูล โดยเปรียบเทียบความคล้ายคลึงของข้อมูล กับจุดศูนย์กลางของแต่ละคลัสเตอร์ (Cluster) หรือค่าเฉลี่ย (Mean) เป็นการแบ่งแบบ Partitional clustering ด้วยการแบ่งข้อมูลออกเป็นส่วน ตามจำนวนกลุ่มที่ระบุ มี 4 ขั้นตอน

(1) mark - กำหนดจำนวนกลุ่ม K กลุ่ม และกำหนดจุดศูนย์กลางเริ่มต้นจำนวน K จุดด้วยการสุ่ม

(2) distance - นำวัตถุทั้งหมดจัดเข้ากลุ่มที่มีจุดศูนย์กลางที่อยู่ใกล้วัตถุนั้นมากที่สุด โดย คำนวณจากการวัดระยะห่างระหว่างจุดที่น้อยที่สุด

(3) center - คำนวณจุดศูนย์กลาง K จุดใหม่ โดยหาจากค่าเฉลี่ยทุกวัตถุที่อยู่ในกลุ่ม

(4) repeat - ทำซ้ำในข้อ 2 จนกระทั่งจุดศูนย์กลางไม่เปลี่ยนแปลงสมการสูตร Euclidean

$$dist = \sqrt{\sum_{k=1}^{n}(P_k - q_k)^2}$$

Computer Science Department Faculty of Informatics Mahasarakham University

หรือ

$$\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

ตัวอย่างที่1 K-mean นำภาพมาจากกริดที่ 1 โดยทำการสุ่ม C1 = 25 ,C2 = 200

12	19	25	178	168
12	25	191	160	170
23	46	15	190	199
28	42	57	200	241
25	15	17	214	216

ภาพประกอบที่ 3.52 ภาพตัวอย่างที่นำมาทำ K-mean กริดที่1

ยกตัวอย่างคำนวณจุดที่ 1

12	19	25	178	168
12	25	191	160	170
23	46	15	190	199
28	42	57	200	241
25	15	17	214	216

ภาพประกอบที่ 3.53 ตัวอย่างการจัดกลุ่มข้อมูลตัวอย่างที่ 1

$$dist = \sqrt{\sum_{k=1}^{n}(P_k - q_k)^2} \quad \text{หรือ} \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

$$dist1 = \sqrt{(12 - 25)^2}$$
$$= 13$$
$$dist2 = \sqrt{(12 - 200)^2}$$
$$= 188$$

Computer Science Department
Faculty of Informatics, Mahasarakham University

ดังนั้น 12 จะถูกจัดในกลุ่มข้อมูลเดียวกับ C1 = 25 เนื่องจากมีระยะห่างที่น้อยกว่า C2 = 200 แสดงดังภาพประกอบที่ 3.54

12	19	25	178	168
12	25	191	160	170
23	46	15	190	199
28	42	57	200	241
25	15	17	214	216

ภาพประกอบที่ 3.54 ตัวอย่างผลลัพธ์การจัดกลุ่มข้อมูลตัวอย่างที่ 1

เมื่อคำนวณต่อไปเรื่อยๆจนครบทุกพิกเซลแล้วจะทำการคำนวณจุดศูนย์กลางใหม่โดยหาจากค่าเฉลี่ยทุกตัวที่อยู่ในกลุ่มจนกว่าค่าจุดศูนย์กลางจะไม่เปลี่ยนแปลง

12	19	25	178	168
12	25	191	160	170
23	46	15	190	199
28	42	57	200	241
25	15	17	214	216

ภาพประกอบที่ 3.55 ตัวอย่างผลลัพธ์การจัดกลุ่มข้อมูลกริดที่1

Computer Science Department
Faculty of Informatics, Mahasarakham University

ภาพประกอบที่ 3.56 ตัวอย่างผลลัพธ์การกำหนดขนาด anchor โดยใช้ K-mean

ตัวอย่างที่2 K-mean นำภาพมาจากกริดที่ 2 โดยทำการสุ่ม C1 = 50 ,C2 = 215

178	194	188	17	68
158	215	19	60	70
190	182	15	90	99
145	164	178	50	41
187	185	17	14	26

ภาพประกอบที่ 3.57 ภาพตัวอย่างที่นำมาทำ k-mean กริดที่2

ยกตัวอย่างคำนวณจุดที่ 1

178	194	188	17	68
158	215	19	60	70
190	182	15	90	99
145	164	178	50	41
187	185	17	14	26

ภาพประกอบที่ 3.58 ตัวอย่างการจัดกลุ่มข้อมูลตัวอย่างที่ 2

Computer Science Department
Faculty of Informatics, Mahasarakham University

$$dist = \sqrt{\sum_{k=1}^{n}(P_k - q_k)^2} \quad \text{หรือ} \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

$$dist1 = \sqrt{(178 - 50)^2}$$
$$= 128$$
$$dist2 = \sqrt{(178 - 215)^2}$$
$$= 37$$

ดังนั้น 178 จะถูกจัดในกลุ่มข้อมูลเดียวกับ C2=215 เนื่องจากมีระยะห่างที่น้อยกว่า C1=50 แสดงดังภาพประกอบที่ 3.59

178	194	188	17	68
158	215	19	60	70
190	182	15	90	99
145	164	178	50	41
187	185	17	14	26

ภาพประกอบที่ 3.59 ตัวอย่างผลลัพธ์การจัดกลุ่มข้อมูลตัวอย่างที่ 2

เมื่อคำนวณต่อไปเรื่อยๆจนครบทุกพิกเซลแล้ว จะทำการคำนวณจุดศูนย์กลาง ใหม่ โดยหาจากค่าเฉลี่ยทุกตัวที่อยู่ในกลุ่ม จนกว่าค่าจุดศูนย์กลางจะไม่เปลี่ยนแปลง

178	194	188	17	68
158	215	19	60	70
190	182	15	90	99
145	164	178	50	41
187	185	17	14	26

ภาพประกอบที่ 3.60 ตัวอย่างผลลัพธ์การจัดกลุ่มข้อมูล กริดที่2

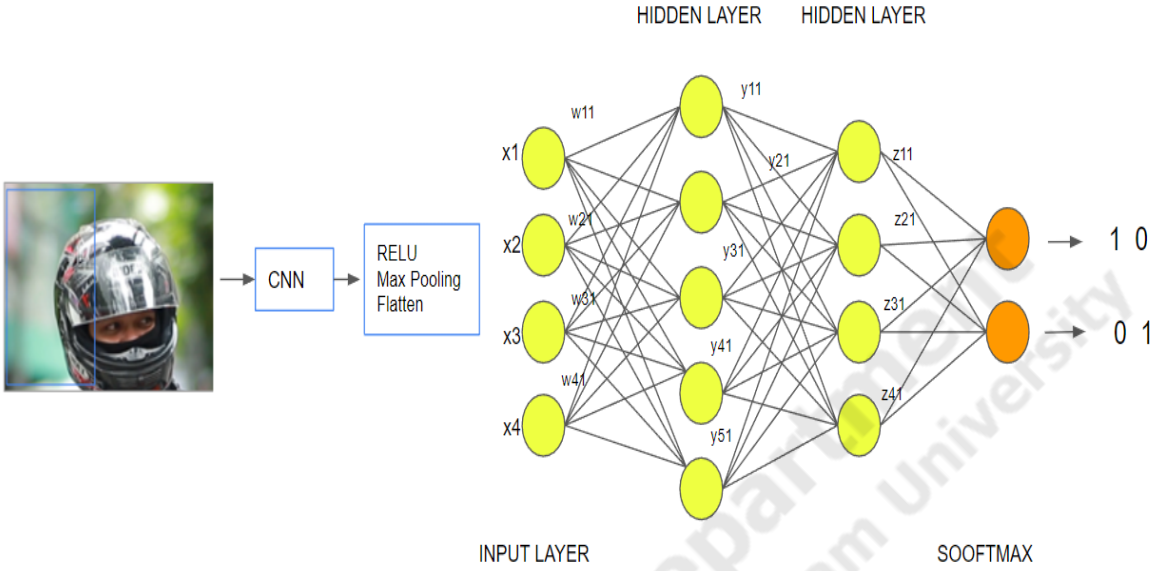
Computer Science Department
Faculty of Informatics, Mahasarakham University

ภาพประกอบที่ 3.61 ตัวอย่างผลลัพธ์การกำหนดขนาด anchor โดยใช้ K-mean

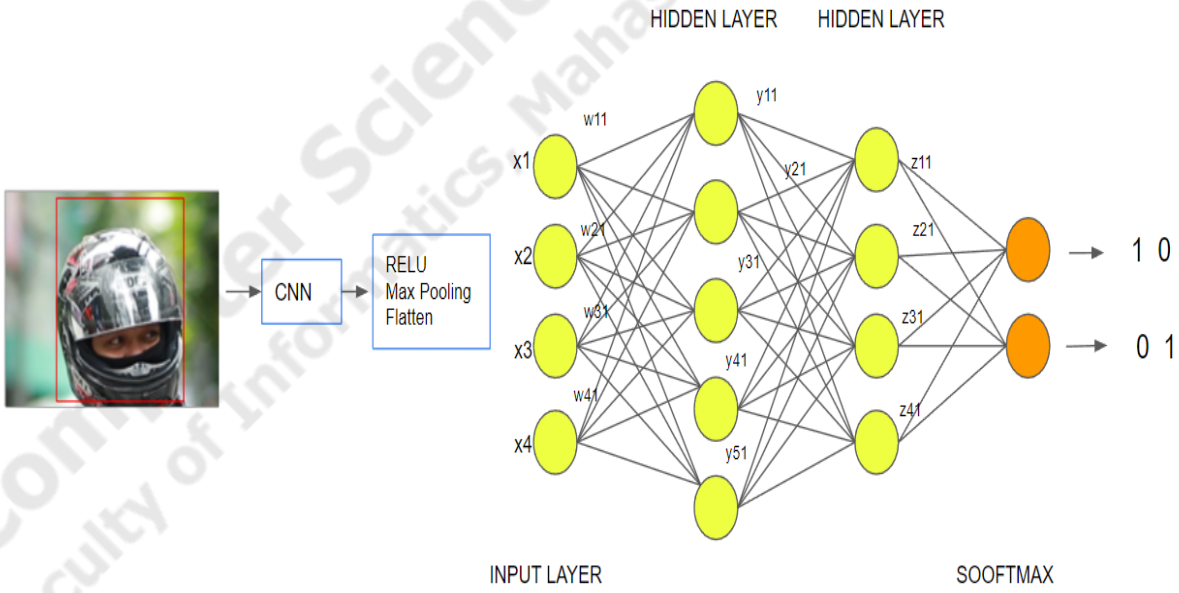
ในกริดอื่นๆก็ทำเช่นกัน จะได้ผลลัพธ์แสดงดัง ภาพประกอบที่ 3.62

ภาพประกอบที่ 3.62 การแบ่งกลุ่มข้อมูลในแต่ละช่อง โดยแต่ละช่องจะมี 2 anchor

นำแต่ละ boundary block สกัด Features ด้วย CNN และเข้าโมเดลเพื่อตรวจสอบว่าอยู่ในคลาสใด ตัวอย่างนำกริดที่ 1 block สีน้ำเงินและ block สีแดงแสดงดัง ภาพประกอบที่ 3.63 และภาพประกอบที่ 3.64



ภาพประกอบที่ 3.63 นำแต่ละ boundary block สีน้ำเงิน สกัด Features ด้วย CNN และเข้าโมเดล



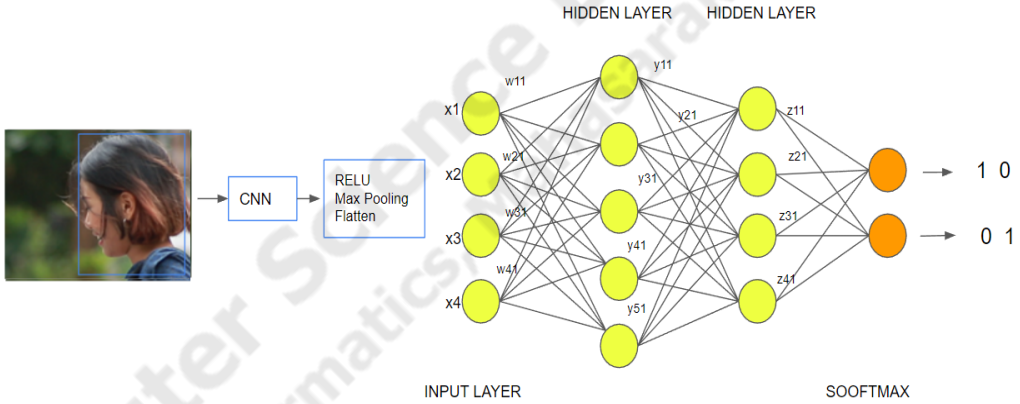
ภาพประกอบที่ 3.64 นำแต่ละ boundary block สีแดง สกัด Features ด้วย CNN และเข้าโมเดล

การเก็บค่า Y ของ กริดที่ 1 หลังจากเข้าโมเดลเราจะทราบว่าวัตถุที่อยู่ในกรอบนั้นเป็นคลาสอะไร

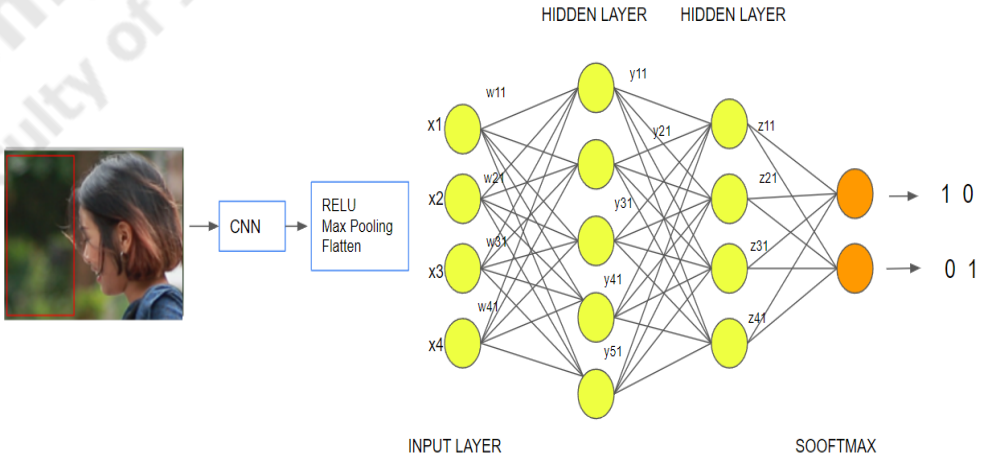
Computer Science Department
Faculty of Informatics Mahasarakham University

$$y = \begin{bmatrix} P = ? \\ bx \\ by \\ bh \\ bw \\ 1 \\ 0 \\ P = ? \\ bx \\ by \\ bh \\ bw \\ 1 \\ 0 \end{bmatrix}$$

ตัวอย่างนำกริดที่ 2 block สีน้ำเงินและ block สีแดง แสดงดังภาพประกอบที่ 3.65 และภาพประกอบที่ 3.66



ภาพประกอบที่ 3.65 นำแต่ละ boundary block สีน้ำเงิน สกัด Features ด้วย CNN และเข้าโมเดล



ภาพประกอบที่ 3.66 นำแต่ละ boundary block สีแดง สกัด Features ด้วย CNN และเข้าโมเดล

Computer Science Department
Faculty of Informatics Mahasarakham University

การเก็บค่า Y ของ กริดที่ 2 หลังจากเข้าโมเดล เราจะทราบว่าวัตถุที่อยู่ในกรอบนั้นเป็นคลาสอะไร

$$y = \begin{bmatrix} P = ? \\ bx \\ by \\ bh \\ bw \\ 0 \\ 1 \\ P = ? \\ bx \\ by \\ bh \\ bw \\ 0 \\ 1 \end{bmatrix}$$

ในกริดอื่นๆก็ทำเช่นเดียวกันจนครบทุกกริด จากนั้นต้องหาค่า probity โดยหาจากค่า IOU

3.3.8 IOU ในขั้นตอนนี้จะเป็นการนำแต่ละ boundary block ไปเปรียบเทียบกับผลเฉลย

สูตรคำนวณ IOU ดังสมการที่ 1 และ 2

$$IOU = \frac{area_{intersection}}{areaunion} \quad (1)$$

$$areaunio = area_{box1} + area_{box2} - area_{intersection} \quad (2)$$

ยกตัวอย่าง นำกริดที่หนึ่งมาหาค่า IOU

Computer Science Department
Faculty of Informatics, Mahasarakham University

ภาพประกอบที่ 3.67 ภาพตัวอย่างการคำนวณ IOU

ตัวอย่างการหาค่า IOU ของกรอบสีน้ำเงิน

Box สีน้ำเงิน

1	1	0	1	1	1
1	0	0	1	1	1
1	0	1	1	1	1
1	0	1	1	1	1
1	0	1	1	1	1
1	1	1	1	1	1
0	1	1	1	1	1
0	1	1	1	1	1
0	1	1	1	1	1

ผลเฉลย

1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1
0	1	1	1	1	1
0	1	1	1	1	1

ภาพประกอบที่ 3.68 ตัวอย่างข้อมูลของกรอบสีน้ำเงินและผลเฉลย

Intersection area

1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1
0	1	1	1	1	1
0	1	1	1	1	1

Computer Science Department
Faculty of Informatics Mahasarakham University

ภาพประกอบที่ 3.69 Intersection area

จากสูตร IOU คือ

$$IOU = \frac{area_{intersection}}{areaunion} \quad (1)$$

$$areaunio = area_{box1} + area_{box2} - area_{intersection} \quad (2)$$

วิธีการคำนวน

$$areaunio = 54 + 54 - 45 = 63$$

$$IOU = \frac{45}{63} = 0.71$$

ดังนั้นค่า probity ของ box สีน้ำเงินเท่ากับ 0.71 ในกรอบสีแดงใช้วิธีการเดียวกัน และคำนวณทุกกริดการเก็บค่า Y ของ กริดตัวอย่าง กรอบสีน้ำเงินและกรอบสีแดง หลังจากทำ IOU จะได้ค่า probity

$$Y = \begin{bmatrix} P = 0.71 \\ 0.56 \\ 0.91 \\ 0.57 \\ 0.40 \\ 1 \\ 0 \\ P = 0.81 \\ 0.21 \\ 0.32 \\ 0.62 \\ 0.50 \\ 1 \\ 0 \end{bmatrix}$$

ตัวอย่างการเก็บค่า Y ของ กริดที่2 กรอบสีน้ำเงินและกรอบสีแดง หลังจากทำ IOU *จะได้ค่า* probity

Computer Science Department
Faculty of Informatics, Mahasarakham University

$$Y = \begin{bmatrix} P = 0.73 \\ 0.93 \\ 0.81 \\ 0.77 \\ 0.50 \\ 0 \\ 1 \\ P = 0.93 \\ 0.102 \\ 0.32 \\ 0.121 \\ 0.50 \\ 0 \\ 1 \end{bmatrix}$$

หลังจากหาค่า probity จาก IOU ทั้งหมด แสดงดังภาพตัวอย่างที่ 3.70

ภาพประกอบที่ 3.70 ตัวอย่างหลังจากการหาค่า probity ทั้งหมด

เมื่อหาค่า Y ได้ครบแล้วเราจะกำหนดว่าถ้าค่า probity น้อยกว่า 0.7 จะไม่นำค่านั้นมาคำนวณ

ภาพประกอบที่ 3.71 ตัวอย่างหลังจากการหาค่า probity ที่มากกว่า 0.7

3.3.9 non-max suppression

โดยวิธีการนี้จะทำการคำนวณค่า IOU ซึ่งเป็นหาอัตราส่วน ของ Intersection area และ Union area ซึ่งเราจะกำหนดว่า ถ้าค่า IOU มากกว่า 0.5 เมื่อค่า IOU สูง แสดงมี box กำลังซ้อนกันอยู่ เราจะใช้วิธีการ non-max suppression ซึ่งจะสนใจ box ที่มีความน่าจะเป็นสูงสุด ถ้าค่า IOU ต่ำกว่า 0.5 เราจะใช้คำตอบของทั้งสอง box ใช้สูตรดังนี้

$$IOU = \frac{Intersection\ area}{Union\ area}$$

ภาพประกอบที่ 3.72 ตัวอย่างกริดที่ 1

Box สีน้ำเงิน

1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1
0	1	0	1	1	1
0	1	1	1	1	1
0	1	1	1	1	1

Box สีแดง

1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1
0	1	1	1	1	1
0	1	1	1	1	1

ภาพประกอบที่ 3.73 ตัวอย่างข้อมูลของกรอบสีน้ำเงินและกรอบสีแดง

Intersection area

1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1
0	1	1	1	1	1
0	1	1	1	1	1

ภาพประกอบที่ 3.74 Intersection area ของกรอบสีน้ำเงินและกรอบสีแดง

จากสูตร IOU คือ

$$IOU = \frac{area_{intersection}}{areaunion} \quad (1)$$

$$areaunio = area_{box1} + area_{box2} - area_{intersection} \quad (2)$$

Computer Science Department
Faculty of Informatics, Mahasarakham University

$$areaunio = 54 + 54 - 50 = 58$$

$$IOU = \frac{50}{58} = 0.86$$

จากการคำนวณ IOU ของกริดตัวอย่างได้ค่า IOU เท่ากับ 0.86 ซึ่งมากกว่าที่เรากำหนดไว้ แสดงว่า กรอบสีน้ำเงินและกรอบสีแดงกำลังครอบวัตถุเดียวกันอยู่ จึงใช้วิธีการ non-max suppression ซึ่งจะสนใจ box ที่มีความน่าจะเป็นสูงสุด ในกริดที่ 2 ก็ทำเช่นกัน จนครบทุกกริด

ภาพประกอบที่ 3.75 ตัวอย่างกรอบที่มีความน่าจะเป็นสูงสุด

เมื่อคำนวณครบทุกกริดแล้วจะได้ผลลัพธ์แสดงดังภาพประกอบที่ 3.76

ภาพประกอบที่ 3.76 ตัวอย่างผลลัพธ์ non-max suppression

Computer Science
Faculty of Informatics, Mahasarakham University