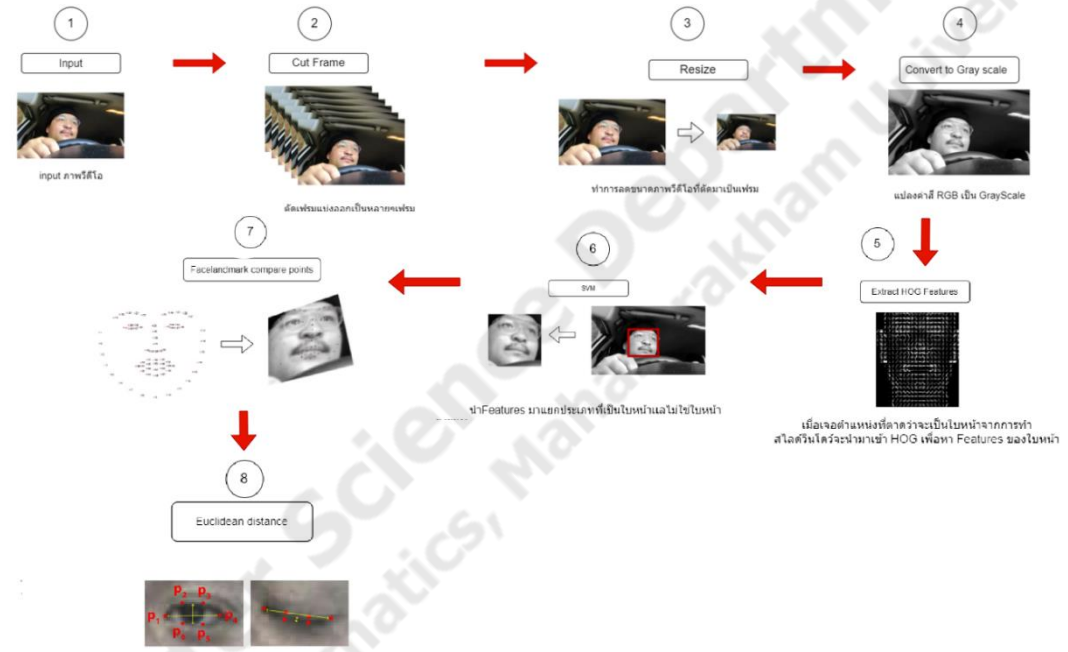
บทที่ 3

วิธีดำเนินงานวิจัย

ในบทนี้จะกล่าวถึงขั้นตอนในการดำเนินงานของโครงงานปริญญานิพนธ์ ซึ่งจะทำให้ทราบถึงการวิเคราะห์ระบบ กระบวนการประมวลผลของระบบ และขั้นตอนการทำงานของระบบเป็นอย่างไร โดยขั้นตอนในการดำเนินงาน ระบบของ Drowsiness Detection มีรายละเอียด ดังภาพประกอบที่ 3.1



ภาพประกอบที่ 3.1 รายละเอียดขั้นตอนการทำงานของระบบ

จากภาพประกอบที่ 3.1 มีขั้นตอนการทำงานดังนี้

ขั้นตอนที่ 1 Input นำภาพวีดีโอ ที่มีความยาว 1 นาที 30 เฟรมต่อวินาที

ขั้นตอนที่ 2 ทำการตัดเฟรมจาก Input ที่นำเข้าแบ่งออกเป็นหลาย ๆ เฟรม

ขั้นตอนที่ 3 ทำการลดขนาด Input (Resize)

ขั้นตอนที่ 4 แปลงค่าสีจาก RGB เป็น Gray scale

ขั้นตอนที่ 5 นำภาพที่ได้จากการแปลงค่าสีเสร็จแล้ว เข้ากระบวนการ HOG เพื่อคำนวณหาฟีเจอร์ของภาพหรือเฟรมที่นำเข้ามา

ขั้นตอนที่ 6 นำฟีเจอร์เข้ากระบวนการ SVM เพื่อจัดกลุ่มของข้อมูลภาพที่ได้จากขั้นตอนที่ 3 ว่าภาพที่ได้เป็นใบหน้าหรือไม่ใช่ใบหน้า

Computer Science Department
Faculty of Informatics, Mahasarakham University

ขั้นตอนที่ 7 นำภาพที่ได้มาเข้ากระบวนการหาจุดบนใบหน้าด้วย Facial Landmark เพื่อใช้ในการตรวจสอบสถานะของดวงตาในขั้นตอนถัดไป

ขั้นตอนที่ 8 ตรวจสอบหาค่าสถานะดวงตาด้วย Euclidean distance โดยรายละเอียดการทำงานอธิบายในหัวข้อที่ 3.2

3.1 ขั้นตอนการจัดเตรียมชุดข้อมูล

3.1.1 จัดเตรียมชุดข้อมูล

ระบบบใช้ชุดข้อมูลจากการเก็บข้อมูลเองของผู้พัฒนาจะเป็นภาพวิดีโอที่มีบุลคลกำลังขับขี่รถยนต์ในสถานะการณ์ที่แตกต่างกัน เวลาต่างกันแสง หรือความสว่างไม่เท่ากัน โดยมีจำนวณ 10 วิดีโอ โดยแต่ละวิดีโอจะมีความยาวประมาณ 1 - 2 นาที ในแต่ละวิดีโอจะมี 30 เฟรมต่อวินาที โดยจะแบ่งเป็นชุดวิดีโอกลางวันจำนวน 5 ชุด และกลางคืน 5 ชุด ในแต่ละวิดีโอจะมีภาพบุลคลที่มีภาวะหลับในหรืออาจจะไม่เกิด

ทางผู้พัฒนาได้เห็นตรงกันว่าชุดวิดีโอในการทดลองยังไม่เพียงพอ ทางผู้พัฒนาจึงได้หาชุดข้อมูลเพิ่มเติมจากเว็บมาใช้ได้จำนวน 10 วีดีโอ ชุดวิดีโอที่นำมามีความยาว 1 – 2 นาที มีจำนวนเฟรม 30 เฟรม ต่อนาทีโดยจะมีสถานะการณ์เป็นตอนกลางคืนทั้งหมด ดังภาพประกอบที่ 3.2

ภาพประกอบที่ 3.2 ภาพตัวอย่างข้อมูล

3.2 การทำงานของระบบ

3.2.1 ข้อมูลนำเข้า นำข้อมูลที่ใช้เข้าสู่กระบวนการแปลงจากวิดีโอที่มีค่าสีเป็น RGB ให้เป็น Gray Scale การแปลงทำได้โดย Gray Scale = 0.299*R + 0.587*G + 0.114*B และนำเข้าสู่กระบวนการ HOG (Histogram of Oriented Gradients) เมื่อเปลี่ยนจาก ภาพ RGB เป็น Gray scale ข้อมูลตัวเลขมีค่าความสว่างอยู่ที่ 0 – 255 ที่นำมาจัดเรียงเป็น Matrix ดังภาพประกอบที่ 3.3

ภาพประกอบที่ 3.3 ภาพที่แปลงเป็น GrayScale

3.2.2 HOG ย่อมาจาก Histogram of Oriented Gradients ซึ่งเป็นเทคนิคการคำนวณฟีเจอร์ (Feature) ที่ใช้ในการแยกแยะวัตถุ (Object Detection) ในภาพถ่าย โดย HOG จะดึงคุณสมบัติของภาพโดยการหา Gradient ของภาพ แล้วแบ่งทิศทางของ Gradient เป็นกลุ่ม Bin และสร้าง Histogram ของ Gradient โดยใช้ Magnitude ของ Gradient ในการกำหนดน้ำหนัก จากนั้นนำ Histogram นี้มาเป็นฟีเจอร์ในการแยกแยะวัตถุในการทำงานของ HOG

ขั้นตอนที่ 1 นำภาพที่ input เข้ามาทำการย่อขนาดหรือลดขนาดลง ยกตัวอย่าง 8x8 พิกเซล และจะถูกแบ่งเป็น Cell ขนาด 3x3 พิกเซล ในการทำ Sliding Window จำนวน 64 ช่อง ดังภาพประกอบที่ 3.4

64	129	183	22	41	217	24	83
199	41	247	110	242	109	48	192
218	76	20	235	15	54	227	131
144	98	56	90	89	69	35	141
139	179	218	91	160	116	19	168
123	221	96	130	161	129	71	232
81	65	100	108	249	201	202	76
60	126	45	185	52	25	184	44

ภาพประกอบที่ 3.4 ขั้นตอนการแบ่ง cell

Computer Science Department
Faculty of Informatics Mahasarakham University

ขั้นตอนที่ 2 นำเซลล์ที่มีขนาด 3x3 มาคำนวณในสมการ เพื่อหา Gx Gy ตามสมการที่ 1 และ 2 ตามลำดับ

$$Gx = I(x+1, y) - I(x-1, y) \quad (1)$$

$$Gy = I(x, y+1) - I(x, y-1) \quad (2)$$

ขั้นตอนการหา Gx จะทำการนำค่าทั้งหมดในเซลล์ในตำแหน่งแนวแกน X มาคำนวณในสมการที่ 1 ดังภาพประกอบที่ 3.5

64	129	183
199	41	247
218	76	20

ภาพประกอบที่ 3.5 ตำแหน่งในแนวแกนนอนของ Gx

จะเริ่มจากการคำนวณ Gx ของ Pixel ที่ตำแหน่งจุดกลาง

$Gx = I(247) - I(199)$

$Gx = 247 - 199$

$Gx = 48$

เมื่อคำนวณช่องนี้เสร็จแล้วจะทำการเลื่อนไป 1 ช่อง Block ขนาด 3x3 เหมือนเดิม โดยจะใช้เพื่อไปคำนวณในช่องต่อไป เมื่อทำการคำนวณตามสมการ Gx เรียบร้อยแล้วจะได้ค่าทั้งหมด โดยขอบเราจะปรับให้เป็น 0 (สีเทา) เพราะข้อมูลตรงช่องที่เป็น 0 ในการคำนวณค่าจะออกนอกอาเรย์ ซึ่งจะทำให้หาค่าในช่องนี้ไม่ได้และเราจะไม่นำค่าที่เป็น 0 ไปคำนวณในขั้นตอนต่อไป ดังภาพประกอบที่ 3.6

0	0	0	0	0	0	0	0
0	48	62	180	-71	119	73	0
0	-113	213	-198	252	-25	156	0
0	-88	178	-89	158	-123	168	0
0	79	12	148	-32	51	117	0
0	-27	157	4	125	-54	232	0
0	19	89	160	41	161	-85	0
0	0	0	0	0	0	0	0

ภาพประกอบที่ 3.6 ค่าทั้งหมดของ Gx

Computer Science Department
Faculty of Informatics, Mahasarakham University

ขั้นตอนการหา Gy จะทำการนำค่าทั้งหมดในเซลล์ในตำแหน่งแนวแกน Y มาคำนวณในสมการที่ 2 ดังภาพประกอบที่ 3.7

64	129	183
199	41	247
218	76	20

ภาพประกอบที่ 3.7 ค่าในเซลล์ที่ให้คำนวณหา Gy

จะเริ่มจากการคำนวณ Gy ของ Pixel ของ Pixel ที่ตำแหน่งจุดกลาง

$$Gy = I(76) - I(129)$$

$$Gy = 76 - 129$$

$$Gy = -35$$

เมื่อคำนวณช่องนี้เสร็จแล้วจะทำการเลื่อนไป 1 ช่อง Block ขนาด 3x3 เหมือนเดิม โดยจะใช้เพื่อไปคำนวณในช่องต่อไป เมื่อทำการคำนวณตามสมการ Gy เรียบร้อยแล้วจะได้ค่าทั้งหมด โดยขอบเราจะปรับให้เป็น 0 (สีเทา) เพราะข้อมูลตรงช่องที่เป็น 0 ในการคำนวณค่าจะออกนอกอาเรย์ซึ่งจะทำให้หาค่าในช่องนี้ไม่ได้และเราจะไม่นำค่าที่เป็น 0 ไปคำนวณในขั้นตอนต่อไปดังภาพประกอบที่ 3.8

0	0	0	0	0	0	0	0
0	-53	-163	213	-26	-163	203	0
0	151	219	-123	115	232	-168	0
0	28	-1	214	45	-116	184	0
0	193	97	-87	116	245	-113	0
0	-128	3	195	133	-44	158	0
0	254	42	-10	-81	69	26	0
0	0	0	0	0	0	0	0

ภาพประกอบที่ 3.8 ค่าทั้งหมดของ Gy

Computer Science Department
Faculty of Informatics Mahasarakham University

ขั้นตอนที่ 3 เป็นขั้นตอนของการหาค่า Magnitude คือ ค่าที่ใช้วัดความสว่างหรือความเข้มของแสง โดยที่เราจะนำเอาค่าของ Gx และ Gy มาคำนวณเสร็จสิ้นในขั้นที่ 2 มาเพื่อหาค่า Magnitude ตามสมการที่ 3

$$m = \sqrt{Gx^2 + Gy^2} \tag{3}$$

0	0	0	0	0	0	0	0
0	48	62	180	-71	119	73	0
0	-113	213	-198	252	-25	156	0
0	-88	178	-89	158	-123	168	0
0	79	12	148	-32	51	117	0
0	-27	157	4	125	-54	232	0
0	19	89	160	41	161	-85	0
0	0	0	0	0	0	0	0

0	0	0	0	0	0	0	0
0	-53	-163	213	-26	-163	203	0
0	151	219	-123	115	232	-168	0
0	28	-1	214	45	-116	184	0
0	193	97	-87	116	245	-113	0
0	-128	3	195	133	-44	158	0
0	254	42	-10	-81	69	26	0
0	0	0	0	0	0	0	0

ภาพประกอบที่ 3.9 ค่าของ Gx Gy ที่ใช้หา magnitude

จากภาพประกอบที่ 3.9 ภาพด้านซ้ายคือ ค่า Gradient ของ Gx และภาพด้านขวาคือ ค่า gradient ของ Gy นำทั้ง 2 ค่ามาหาค่า Magnitude ตามสมการที่ 3 ขั้นตอนการคำนวณ Magnitude

$$m = \sqrt{(48^2 + (-53)^2)(62^2 + (-163)^2)(180^2 + 213^2)(-71^2 + (-26)^2)(119^2 + (-163)^2)(73^2 + 203^2)}$$

$$m = \sqrt{(5113)(30413)(77769)(5717)(40730)(46538)}$$

$$m = [71.5052, 174.3932, 278.8709, 75.6108, 201.8167, 215.7266]$$

ดังนั้นค่า Magnitude ในแถวที่ 0 จะมีค่า ดังนี้

$$m = [71.5052, 174.3932, 278.8709, 75.6108, 201.8167, 215.7266]$$

จะคำนวณตามสมการที่ 3 ไปจนถึงตำแหน่งสุดท้ายของเซลล์ ค่าที่ได้ทั้งหมดในเซลล์จะได้ ดังภาพประกอบที่ 3.10

0	0	0	0	0	0	0	0
0	71.5052	174.3932	278.8709	75.6108	201.8167	215.7266	0
0	188.6001	305.4995	233.0944	277.00	233.3430	229.2596	0
0	92.3471	178.0028	231.7692	164.2832	169.0709	249.1585	0
0	208.5425	97.7394	171.6770	120.3328	250.2518	162.6591	0
0	130.8166	157.0286	195.0410	182.5212	69.6562	280.6920	0
0	254.7096	98.4123	160.0031	90.7854	175.1627	88.8875	0
0	0	0	0	0	0	0	0

ภาพประกอบที่ 3.10 ค่า Magnitude

Computer Science Department
Faculty of Informatics, Mahasarakham University

ขั้นตอนที่ 4 ขั้นตอนการคำนวณ Gradients Direction เป็นขั้นตอนที่สำคัญในการสกัดคุณลักษณะของภาพด้วย HOG โดย Gradients Direction จะช่วยให้เราหาความเอียงของเส้นแต่ละเส้นภายใน cell โดยใช้ Gradient ของภาพในแต่ละจุด สามารถคำนวณได้ตามสมการที่ 4

$$D = \arctan\left(\frac{Gy}{Gx}\right) * \left(\frac{360}{\pi}\right) \quad (4)$$

โดยการคำนวณนั้นจะนำค่า Gx และ Gy มาใช้ในการคำนวณ เพื่อหา Gradients Direction โดย 360 คือค่าอัตราส่วน Bin ที่จะเอามาเพื่อแยกแยะวัตถุและเก็บข้อมูลทิศทางโดยละเอียดแม่นยำ ส่วนค่า $\boldsymbol{\pi}$ คือ ค่าคงที่ที่จะใช้ในการคำนวณก่อนที่จะคูณกับค่า Radian เพื่อแปลงเป็น Degree

ขั้นตอนการคำนวณ Gradients Direction

$$D = \arctan\left(\frac{-53}{48}\right) * \left(\frac{360}{3.14159265359}\right)$$

$$D = -96$$

จะทำการคำนวณตามสมการที่ 4 ให้ครบทุกช่องของตารางและค่าทั้งหมดเมื่อถึงช่องสุดท้ายที่ได้ ดังภาพประกอบที่ 3.11

0	0	0	0	0	0	0	0
0	-96	138	100	-320	-108	140	0
0	254	92	-296	49	192	-94	0
0	325	-1	225	32	-273	95	0
0	135	166	-61	211	156	-88	0
0	-204	2	178	94	-282	69	0
0	171	51	-1	-126	46		0
0	0	0	0	0	0	0	0

ภาพประกอบที่ 3.11 ค่าทั้งหมดของ Gradients Direction

ขั้นตอนที่ 5 เก็บทิศทางลง Bin (Orientation Binning) ในขั้นตอนนี้จะทำให้ทราบน้ำหนักและทิศทางของแต่ละเซลล์ โดยนำค่าที่ได้จากการหาทิศทางของเกรเดียนต์มาลงในช่องภายใน Bin ซึ่ง

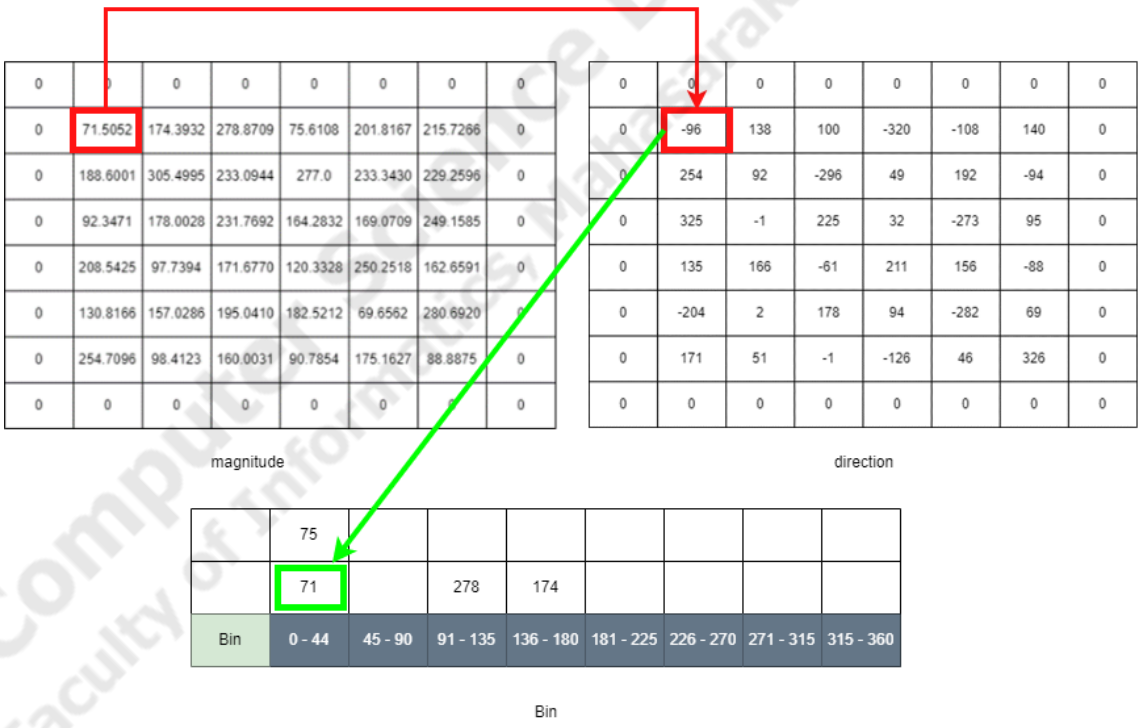
Computer Science Department
Faculty of Informatics, Mahasarakham University

จะทำให้การแบ่งเซลล์แต่ละพิกเซลที่อยู่ในเซลล์จะเป็นทิศทางของเกรเดียนต์ โดยแต่ละเซลล์จะสร้างช่องฮิสโตแกรมสำหรับเก็บทิศทาง ซึ่งจะกำหนดทิศทางเป็น 0 - 360 องศา โดยมีจำนวนช่องของฮิสโตแกรมจำนวน 8 ช่อง (Bin) ดังภาพประกอบที่ 3.12

Bin	0 - 44	45 - 90	91 - 135	136 - 180	181 - 225	226 - 270	271 - 315	316 - 360

ภาพประกอบที่ 3.12 Bin (Orientation Binning)

การเก็บค่าลงใน Bin (Orientation Binning) สามารถทำได้โดยนำค่าของ Gradients Direction และ Magnitude มาเทียบกันในแต่ละช่อง (m[0] , d[0]) โดยจะให้ค่า Gradients Direction จะเป็นค่าที่บอกว่าข้อมูลจะอยู่ในช่วงไหนของ Bin Orientation Binning เมื่อเอาค่า จาก Magnitude มาเทียบ ดังภาพประกอบที่ 3.13



ภาพประกอบที่ 3.13 ตัวอย่างการเก็บค่าลงใน Bin

จะทำขั้นตอนในการเก็บค่าลงใน Bin (Orientation Binning) ไปให้ครบทุกช่อง เมื่อครบทุกช่องแล้วจะได้ดังภาพประกอบที่ 3.14

Computer Science Department
Faculty of Informatics, Mahasarakham University

	90							
	160							
	69							
	157							
	130							
	162							
	171							
	169							
	164							
	178			254				
	229		182	195				
	233	175	208	250				
	201	98	249	97	120			
	75	280	305	215	231			88
	71	277	278	174	233	188		92
Bin	0 - 44	45 - 90	91 - 135	136 - 180	181 - 225	226 - 270	271 - 315	315 - 360

ภาพประกอบที่ 3.14 ค่าทั้งหมดใน Bin

ขั้นตอนการหาค่า Feature Vector จะเอาค่าในแนวตั้งของค่าที่เก็บลงใน Bin มาบวกกันทั้งหมดในขั้นตอนการทำ Feature Vector และจะหาค่า K การหาค่า K เราจะนับจำนวน Index ในแถวแนวตั้งของ Bin เมื่อทำทั้งสองขั้นตอนแล้วจะได้ดังภาพประกอบที่ 3.15

Computer Science Department
Faculty of Informatics, Mahasarakham University

	90							
	160							
	69							
	157							
	130							
	162							
	171							
	169							
	164							
	178			254				
	229		182	195				
	233	175	208	250				
	201	98	249	97	120			
	75	280	305	215	231			88
	71	277	278	174	233	188		92
Bin	0 - 44	45 - 90	91 - 135	136 - 180	181 - 225	226 - 270	271 - 315	315 - 360
K	15	4	5	6	3	1	0	2
FV	2259	830	1222	1185	584	188	0	

ภาพประกอบที่ 3.15 จำนวน Index ในแถวแนวตั้งของ Bin

ขั้นตอนที่ 5 การทำ Normalization

Normalization คือ การปรับสเกลข้อมูลให้อยู่ในช่วงที่ต้องการ เพื่อลดผลข้างเคียงจากข้อมูลที่มีค่ามากหรือน้อยจนเกินไปและเพื่อเปรียบเทียบข้อมูลได้ง่ายขึ้น ในการทำ Normalization นั้นมีหลายวิธี โดยวิธีการทำ Normalization จะแตกต่างกันไปตามเป้าหมายและลักษณะของข้อมูลที่ต้องการ Normalize นำค่า K ที่ได้จากขั้นตอนที่แล้วมาคำนวณในสมการที่ 5

$$K = \sqrt{b_1^2 + b_2^2 + b_3^2} \tag{5}$$

เมื่อนำค่ามาคำนวณในสมการจะได้

$$K = \sqrt{15^2 + 4^2 + 5^2 + 6^2 + 3^2 + 1^2 + 2^2}$$

$$K = \sqrt{225 + 16 + 25 + 36 + 9 + 1 + 4}$$

Computer Science Department
Faculty of Informatics, Mahasarakham University

$$K = \sqrt{316}$$

$$K = 17.7763$$

ต่อมาจะนำค่า K ที่ได้ กับค่า FV (Feature Vector) มาคำนวณในสมการที่ 6

$$F = \left[\frac{b_1}{k}, \frac{b_2}{k}, \frac{b_3}{k}\right] \quad (6)$$

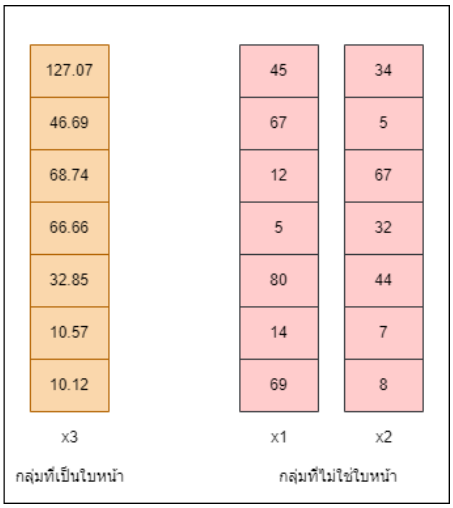
$$F = \left[\frac{2259}{17.7763}, \frac{830}{17.7763}, \frac{1222}{17.7763}, \frac{1185}{17.7763}, \frac{584}{17.7763}, \frac{188}{17.7763}, \frac{180}{17.7763}\right]$$

$$F = [127.0793\ , 46.6913\ , 68.7432\ , 66.6617\ , 32.8527\ , 10.5758, 10.1258]$$

ดังนั้นค่า Feature ที่เราจะนำเข้าไปสู่ในขั้นตอนการหา SVM คือ

$$F = \begin{bmatrix} 127.0793 \\ 46.6913 \\ 68.7432 \\ 66.6617 \\ 32.8527 \\ 10.5758 \\ 10.1258 \end{bmatrix}$$

3.2.3 Support Vector Machine (SVM) เป็นอัลกอริทึมที่สามารถนำมาช่วยแก้ปัญหาการจำแนกข้อมูล ซึ่งจะนำมาใช้ในการวิเคราะห์ข้อมูลและจำแนกข้อมูล ข้อมูลนำเข้า คือ ข้อมูลลักษณะเด่นของรูปภาพที่ได้หลังจากการทำ HOG โดยแบ่งกลุ่มออกเป็น 2 กลุ่มหลัก ๆ คือ กลุ่มที่ใช้ใบหน้าและไม่ใช้ใบหน้าดังภาพประกอบตัวอย่างที่ 3.16



ภาพประกอบที่ 3.16 ข้อมูลนำเข้า SVM

Computer Science Department
Faculty of Informatics, Mahasarakham University

3.2.3.1 กระบวนการทำงานของ SVM

เราจะนำค่าที่ได้จากการทำ HOG เสร็จแล้วที่เป็น Feature Model มาทำในขั้นตอนการทำ SVM เพื่อจำแนกข้อมูล

ขั้นตอนที่ 1 เพิ่มค่า Bias ให้กับ X1 , X2 และ X3 โดยมีค่า Bias เท่ากับ 1 ดังนั้นจะได้ค่าใหม่ดังนี้

$$X_1 = \begin{bmatrix} 45 \\ 67 \\ 12 \\ 5 \\ 80 \\ 14 \\ 69 \\ 1 \end{bmatrix} X_2 = \begin{bmatrix} 34 \\ 5 \\ 67 \\ 32 \\ 44 \\ 7 \\ 8 \\ 1 \end{bmatrix} X_3 = \begin{bmatrix} 127 \\ 46 \\ 68 \\ 66 \\ 32 \\ 10 \\ 10 \\ 1 \end{bmatrix}$$

ขั้นตอนที่ 2 นำคุณลักษณะที่ได้ไปคำนวนหาสมการเส้นตรงที่ 7, 8, 9 ตามลำดับ

$$a_1x_1 * x_1 + a_2x_2 * x_1 + a_3x_3 * x_1 = -1(-ve\ class) \quad (7)$$

$$a_1x_1 * x_2 + a_2x_2 * x_2 + a_3x_3 * x_2 = -1(-ve\ class) \quad (8)$$

$$a_1x_1 * x_3 + a_2x_2 * x_3 + a_3x_3 * x_3 = +1(-ve\ class) \quad (9)$$

จากสมการจะทำการหาค่า a_1 , a_2 และ a_3

$$a1\begin{bmatrix} 45 \\ 67 \\ 12 \\ 5 \\ 80 \\ 14 \\ 69 \\ 1 \end{bmatrix} * \begin{bmatrix} 45 \\ 67 \\ 12 \\ 5 \\ 80 \\ 14 \\ 69 \\ 1 \end{bmatrix} + a2\begin{bmatrix} 34 \\ 5 \\ 67 \\ 32 \\ 44 \\ 7 \\ 8 \\ 1 \end{bmatrix} * \begin{bmatrix} 45 \\ 67 \\ 12 \\ 5 \\ 80 \\ 14 \\ 69 \\ 1 \end{bmatrix} + a3\begin{bmatrix} 127 \\ 46 \\ 68 \\ 66 \\ 32 \\ 10 \\ 10 \\ 1 \end{bmatrix} * \begin{bmatrix} 45 \\ 67 \\ 12 \\ 5 \\ 80 \\ 14 \\ 69 \\ 1 \end{bmatrix} = -1(-ve\ class)$$

$$a1\begin{bmatrix} 45 \\ 67 \\ 12 \\ 5 \\ 80 \\ 14 \\ 69 \\ 1 \end{bmatrix} * \begin{bmatrix} 34 \\ 5 \\ 67 \\ 32 \\ 44 \\ 7 \\ 8 \\ 1 \end{bmatrix} * +a2\begin{bmatrix} 34 \\ 5 \\ 67 \\ 32 \\ 44 \\ 7 \\ 8 \\ 1 \end{bmatrix} * \begin{bmatrix} 34 \\ 5 \\ 67 \\ 32 \\ 44 \\ 7 \\ 8 \\ 1 \end{bmatrix} * +a3\begin{bmatrix} 127 \\ 46 \\ 68 \\ 66 \\ 32 \\ 10 \\ 10 \\ 1 \end{bmatrix} * \begin{bmatrix} 34 \\ 5 \\ 67 \\ 32 \\ 44 \\ 7 \\ 8 \\ 1 \end{bmatrix} *= -1(-ve\ class)$$

$$a1\begin{bmatrix} 45 \\ 67 \\ 12 \\ 5 \\ 80 \\ 14 \\ 69 \\ 1 \end{bmatrix} * \begin{bmatrix} 127 \\ 46 \\ 68 \\ 66 \\ 32 \\ 10 \\ 10 \\ 1 \end{bmatrix} + a2\begin{bmatrix} 34 \\ 5 \\ 67 \\ 32 \\ 44 \\ 7 \\ 8 \\ 1 \end{bmatrix} * \begin{bmatrix} 127 \\ 46 \\ 68 \\ 66 \\ 32 \\ 10 \\ 10 \\ 1 \end{bmatrix} + a3\begin{bmatrix} 127 \\ 46 \\ 68 \\ 66 \\ 32 \\ 10 \\ 10 \\ 1 \end{bmatrix} * \begin{bmatrix} 127 \\ 46 \\ 68 \\ 66 \\ 32 \\ 10 \\ 10 \\ 1 \end{bmatrix} = +1(-ve\ class)$$

Computer Science Department
Faculty of Informatics, Mahasarakham University

จากการแทนค่าในสมการด้านบนจะได้สมการใหม่ ดังนี้

$$18041a_1 + 7000a_2 + 13334a_3 = -1$$

$$7000a_1 + 8744a_2 + 12775a_3 = -1$$

$$13334a_1 + 12775a_2 + 28450a_3 = +1$$

ขั้นตอนที่ 3 การหาค่าน้ำหนัก จากสมการนี้ $a1 = 0.045247$, $a_2 = -0.0074447$ และ $a_3 = -0.0018814$

$$w = \sum_i a_i \, x_i \tag{10}$$

แทนค่า $\boldsymbol{a_i}$ และ $\boldsymbol{x_i}$ ลงในสมการ

$$w = a_1 \begin{bmatrix} 45 \\ 67 \\ 12 \\ 5 \\ 80 \\ 14 \\ 69 \\ 1 \end{bmatrix} + a_2 \begin{bmatrix} 34 \\ 5 \\ 67 \\ 32 \\ 44 \\ 7 \\ 8 \\ 1 \end{bmatrix} + a_3 \begin{bmatrix} 127 \\ 46 \\ 68 \\ 66 \\ 32 \\ 10 \\ 10 \\ 1 \end{bmatrix}$$

$$W = (0.045247) \begin{bmatrix} 45 \\ 67 \\ 12 \\ 5 \\ 80 \\ 14 \\ 69 \\ 1 \end{bmatrix} + (-0.0074447) \begin{bmatrix} 34 \\ 5 \\ 67 \\ 32 \\ 44 \\ 7 \\ 8 \\ 1 \end{bmatrix} + (0.0018814) \begin{bmatrix} 127 \\ 46 \\ 68 \\ 66 \\ 32 \\ 10 \\ 10 \\ 1 \end{bmatrix}$$

$$W = \begin{bmatrix} 2.036115 \\ 3.031549 \\ 0.542964 \\ 0.226235 \\ 3.61976 \\ 0.633458 \\ 3.122043 \\ 0.045247 \end{bmatrix} + \begin{bmatrix} -0.2531198 \\ -0.0372235 \\ -0.4987949 \\ -0.2382304 \\ -0.3275668 \\ -0.0521129 \\ -0.0595576 \\ -0.0074447 \end{bmatrix} + \begin{bmatrix} -0.2389378 \\ -0.0865444 \\ -0.1279352 \\ -0.1241724 \\ -0.0602048 \\ -0.018814 \\ -0.018814 \\ -0.0018814 \end{bmatrix}$$

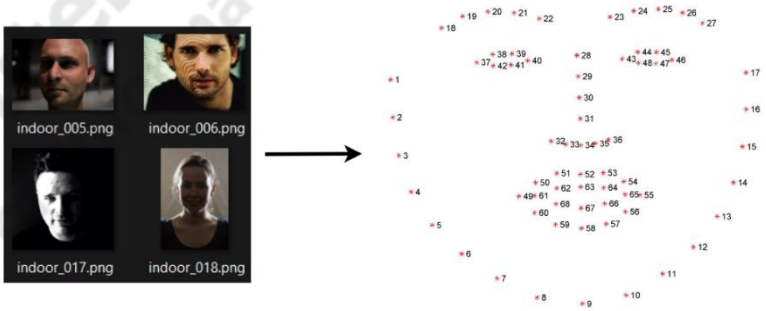
Computer Science Department
Faculty of Information Technology Mahasarakham University

$$W = \begin{bmatrix} 1.5440574 \\ 2.9077811 \\ -0.0837661 \\ -0.1361678 \\ 3.2319884 \\ 0.5625311 \\ 3.0436714 \\ 0.0359209 \end{bmatrix}$$

ดังนั้น ค่าโมเดลที่จะนำไปใช้งาน ต้องนำส่วนของการตรวจจับใบหน้าและไม่ใช้ใบหน้า

$$W = \begin{bmatrix} 1.5440574 \\ 2.9077811 \\ -0.0837661 \\ -0.1361678 \\ 3.2319884 \\ 0.5625311 \\ 3.0436714 \\ 0.0359209 \end{bmatrix}$$

3.2.4 Facial Landmark เมื่อเราได้ค่าใบหน้าแล้วต่อมาเราจะนำใบหน้าไปเทียบจุดและมาร์กจุด Landmark บนใบหน้าเพื่อหาว่าส่วนไหนคือดวงตาของผู้ขับขี่ Facial Landmark คือ จุดบนใบหน้าที่ถูกกำหนดไว้เพื่อใช้ในการติดตามและวิเคราะห์คุณลักษณะของใบหน้า เช่น คิ้ว จมูก ตา และปาก โดยใช้โมเดล shape_predictor_68_face_landmarks เพื่อคำนวณหาจุดบนใบหน้า โดยโมเดลนี้ได้มาจากการทำ Detecting Facial Keypoints ดังภาพประกอบที่ 3.17

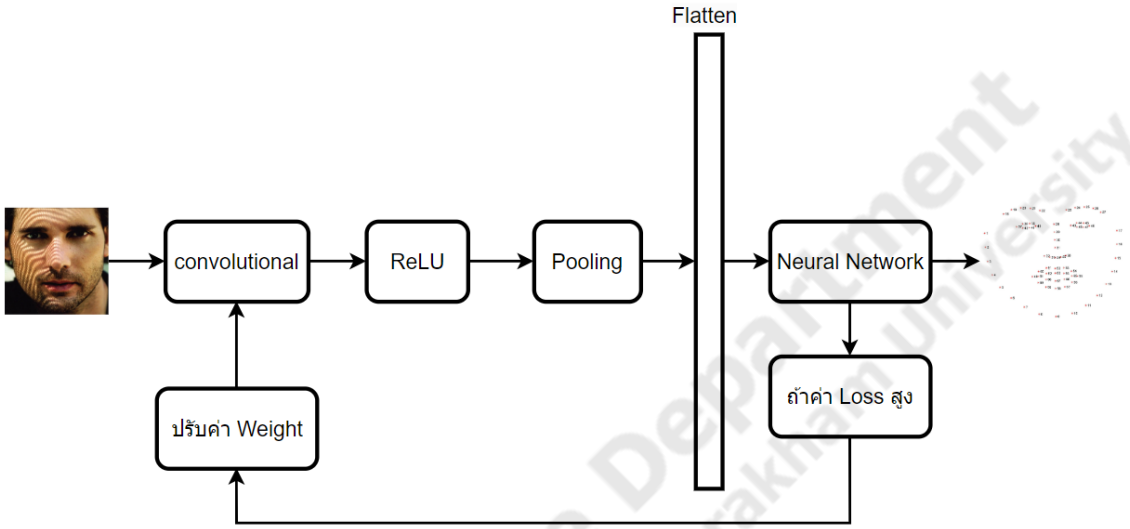


ภาพประกอบที่ 3.17 shape_predictor_68_face_landmarks

ในกระบวนการ Detecting Facial Keypoints โมเดล CNN จะรับภาพใบหน้าเป็นข้อมูลเข้าไปแล้วทำการสกัดลักษณะของภาพออกมาด้วย Filter ซึ่งจะสร้าง Feature Map ขึ้นมา และนำ Feature Map นั้นไปผ่าน Pooling Layer เพื่อลดขนาดลง จากนั้นจึงทำการ Train โมเดล CNN ด้วย Dataset ที่มีการตั้งค่าจุดสำคัญบนใบหน้าแล้ว โดยใช้ค่าจุดสำคัญบนใบหน้าเหล่านี้เป็น Output ที่ต้องการจากโมเดล

Computer Science Department
Faculty of Applied Science and Technology Mahasarakham University

ในการ Train โมเดล CNN นั้นจะใช้ Loss Function เพื่อหาค่าความต่างระหว่าง Output จริง กับ Output ที่โมเดลทำนายได้ และนำค่า Loss นี้ไปใช้ในการปรับค่า Weight ของโมเดล CNN ดังภาพประกอบที่ 3.18



ภาพประกอบที่ 3.18 ขั้นตอนในการ trianmodel

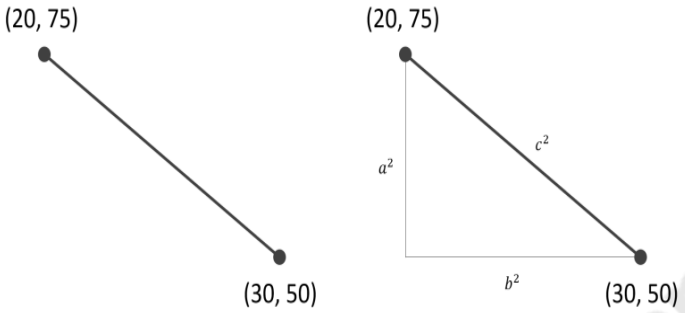
เมื่อโมเดล CNN ได้ถูก Train แล้ว จะสามารถนำไปใช้ในการ Detecting Facial Keypoints บนภาพใหม่ได้ โดยการใช้โมเดล CNN นี้ใช้กับภาพใหม่ จะทำการทำนายค่าจุดสำคัญบนใบหน้าและสามารถนำผลลัพธ์ที่ได้มาวางไว้บนภาพหรือนำไปใช้ในงานต่าง ๆ ตามที่เราต้องการ

3.2.5 Euclidean Distance

นำเอาลักษณะเด่นของภาพมาตรวจจับความใกล้เคียงของดวงตาเพื่อหาว่าดวงตาเปิดหรือปิด โดยใช้ Euclidean Distance สูตรทั่วไปของ Euclidean Distance เขียนได้ตามรูปด้านล่าง โดยที่ q คือ vector_1 และ p คือ vector_2 และความยาวของทั้งสองเวกเตอร์เท่ากับ n ตัวอย่างเช่น q = (30, 50, 60, 100), p = (20, 75, 50, 45)

$$d = \sqrt{\sum_{i=1}^{n}(q_i - p_i)^2} \tag{11}$$

ยกตัวอย่างว่ามี Data Point อยู่ 2 จุด คือ [20 , 75] และ [30 , 50] เราจะหาระยะห่างระหว่างสองจุดนี้โดยใช้หลัก พีทาโกรัส $a^2 + b^2 = C^2$ ดังภาพประกอบที่ 3.19

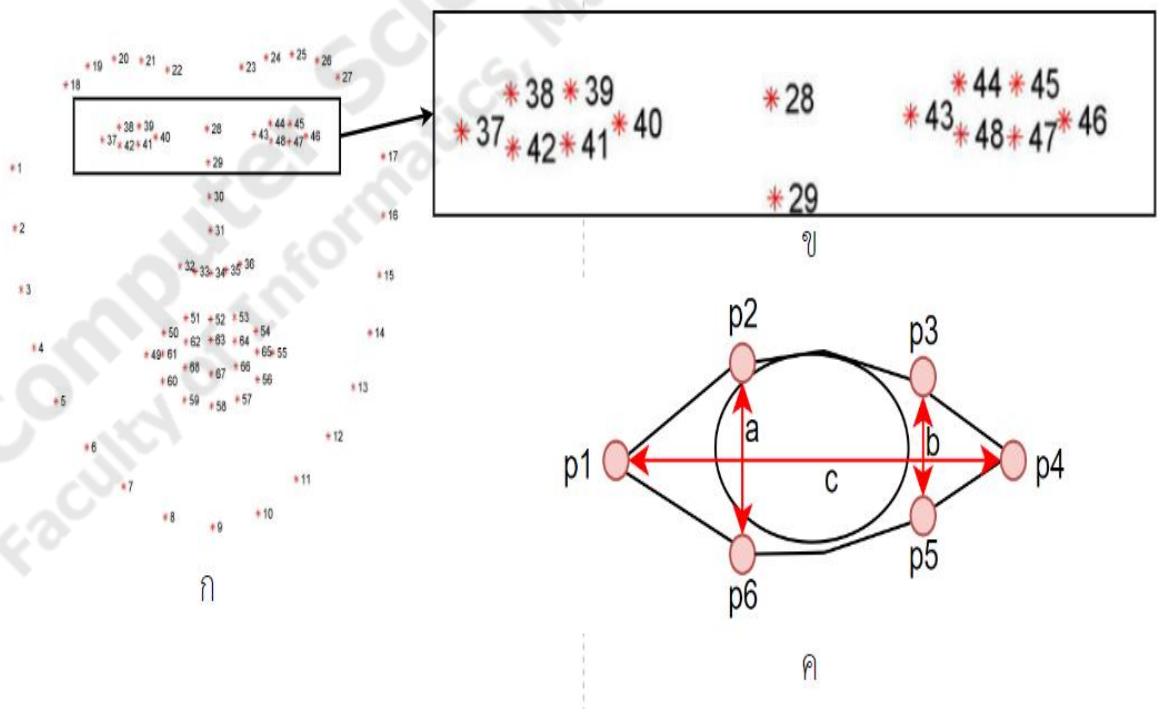


ภาพประกอบที่ 3.19 การหาระยะห่างด้วยพีทาโกรัส

การนำ Euclidean Distance มาใช้เพื่อหาระยะห่างระหว่างคุณลักษณะเด่นของดวงตาที่เราต้องการและคุณลักษณะเด่นของภาพดวงตาต้นแบบ เพื่อหาว่าภาพดวงตาที่เป็นเป้าหมายนั้นมีความใกล้เคียงกันกับภาพดวงตาต้นแบบใดมากที่สุด

3.2.5.1 การใช้ Euclidean Distance หา Eye Aspect Ratio (EAR)

เพื่อตรวจสอบสถานะดวงตาว่าเปิดหรือปิด โดยทำการคำนวณหาระยะห่างของแต่ละจุดของดวงตา ดังภาพประกอบที่ 3.20



ภาพประกอบที่ 3.20 เส้นตรงของแต่ละจุด

โดยจะใช้ตัวอย่างตำแหน่งจุดของดวงตาในการคำนวณ ดังภาพประกอบที่ 3.21

Computer Science Department
Faculty of Informatics, Mahasarakham University

ภาพประกอบที่ 3.21 ตัวอย่างตำแหน่งจุดของดวงตา

คำนวณหาระยะห่าง (p) ของแต่ละจุดด้วยสมการที่ 12

$$p = \sqrt{(x1 - x2)^2 + (y1 - y2)^2} \quad (12)$$

จากสมการที่ 12

p คือ ระยะห่างระหว่างจุดที่ 1 ถึงจุดที่ 2

x1 คือ ตำแหน่งแกน x ของจุดที่ 1

x2 คือ ตำแหน่งแกน x ของจุดที่ 2

y1 คือ ตำแหน่งแกน y ของจุดที่ 1

y2 คือ ตำแหน่งแกน y ของจุดที่ 2

ตัวอย่างการคำนวณเพื่อตรวจสอบระยะทางการปิดเปิดของดวงตาข้างซ้าย

ขั้นตอนที่ 1 ตรวจสอบระยะทางการปิดเปิดของดวงตาข้างซ้ายคำนวณได้จากสมการที่ 12 โดยพิจารณาจากเส้นตรง a b และ c โดยแสดง ดังภาพที่ 3.21 (ค) โดยที่

เส้นตรง a ประกอบด้วยจุดที่ p2 (313, 232) และ p6 (313, 225) แล้วคำนวณหาระยะทางจากจุด p2 ไปยัง p6 (a) มีค่าดังนี้

$$a = \sqrt{(313 - 313)^2 + (225 - 232)^2} = 7$$

เส้นตรง b ประกอบด้วยจุดที่ p3 (322, 231) และ p5 (322, 224) แล้วคำนวณหาระยะทางจากจุด p3 ไปยัง p5 (b) มีค่าดังนี้

$$b = \sqrt{(322 - 322)^2 + (224 - 231)^2} = 7$$

เส้นตรง c ประกอบด้วยจุดที่ p1 (330, 229) และ p4 (306, 229) แล้วคำนวณหาระยะทางจากจุด p1 ไปยัง p4 (c) มีค่าดังนี้

Computer Science Department
Faculty of Informatics Mahasarakham University

$$c = \sqrt{(306 - 330)^2 + (229 - 229)^2} = 24$$

ขั้นตอนที่ 2 คำนวณหาระยะทางการปิดเปิดของดวงตาข้างซ้ายด้วยสมการที่ 13

$$p = \frac{a + b}{k * c} \tag{13}$$

ซึ่งกำหนดให้

p คือ ระยะทางการปิดเปิดของดวงตา

k คือ จำนวนเส้นแนวตั้งระหว่างเปลือกตาบน และเปลือกตาล่างซึ่งในโปรเจคนี้มีจำนวน 2 เส้น จึงกำหนดให้ k มีค่าเท่ากับ 2

a คือ ระยะทางระหว่าง p2 ถึง p6

b คือ ระยะทางระหว่าง p3 ถึง p5

c คือ ระยะทางระหว่าง p1 ถึง p4

จากนั้นแทนค่า a, b และ c จากขั้นตอนที่ 1 ลงในสมการที่ 13 จะได้ค่าค่าระยะทางดวงตาข้างซ้าย (p_{left}) ดังนี้

$$p_{left} = \frac{7 + 7}{2.0 * 24} = 0.29166666666$$

ตัวอย่างการคำนวณเพื่อตรวจสอบระยะทางการปิดเปิดของดวงตาข้างขวา

ขั้นตอนที่ 1 ตรวจสอบระยะทางการปิดเปิดของดวงตาข้างขวา คำนวณได้จากสมการที่ 12 โดยพิจารณาจากเส้นตรง a b และ c โดยแสดง ดังภาพที่ 3.21 (ค) โดยที่

เส้นตรง a ประกอบด้วยจุดที่ p2 (373, 227) และ p6 (372, 220) แล้วคำนวณหาระยะทางจากจุด p2 ไปยัง p6 (a) มีค่าดังนี้

$$a = \sqrt{(372 - 373)\text{^}2 + (220 - 227)\text{^}2} = 7.07106781187$$

เส้นตรง b ประกอบด้วยจุดที่ p3 (382, 226) และ p5 (381, 219) แล้วคำนวณหาระยะทางจากจุด p3 ไปยัง p5 (b) มีค่าดังนี้

$$b = \sqrt{(381 - 382)\text{^}2 + (219 - 226)\text{^}2} = 7.07106781187$$

เส้นตรง c ประกอบด้วยจุดที่ p1 (389, 223) และ p4 (364, 226) แล้วคำนวณหาระยะทางจากจุด p1 ไปยัง p4 (c) มีค่าดังนี้

$$c = \sqrt{(364 - 389)^2 + (226 - 223)^2} = 25.179356624$$

Computer Science Department
Faculty of Informatics, Mahasarakham University

ขั้นตอนที่ 2 คำนวณหาระยะทางการปิดเปิดของดวงตาข้างขวาด้วยสมการที่ 13

จากนั้นแทนค่า a, b และ c จากขั้นตอนที่ 1 ลงในสมการที่ 13 จะได้ค่าค่าระยะทางดวงตาข้างขวา (p_{right}) ดังนี้

$$p_{right} = \frac{7.07106781187 + 7.07106781187}{2.0 * 25.179356624} = 0.28082797815$$

เมื่อได้ระยะทางการปิดเปิดของดวงตาข้างซ้าย (p_{left}) และข้างขวา (p_{right}) ให้นำมาหาค่าเฉลี่ยตามสมการที่ 14

$$avg = \frac{p_{left} + p_{right}}{k} \quad (14)$$

ซึ่งกำหนดให้

avg คือ ค่าเฉลี่ยของระยะทางการปิดเปิดดวงตาข้างซ้าย (p_{left}) และข้างขวา (p_{right})

k คือ จำนวนของดวงตาที่นำมาหาค่าเฉลี่ย ซึ่งในโปรเจคนี้ใช้ดวงตา 2 ดวง ค่า k จึงมีค่าเท่ากับ 2

p_{left} คือ ระยะทางการปิดเปิดของดวงตาข้างซ้าย

p_{right} คือ ระยะทางการปิดเปิดของดวงตาข้างขวา

จากนั้นแทนค่าระยะทางการปิดเปิดของดวงตาข้างซ้าย (p_{left}) และข้างขวา (p_{right}) ลงในสมการที่ 14 จะได้ค่าเฉลี่ยระยะทางดวงตาข้างซ้าย (p_{left}) และข้างขวา (p_{right}) ดังนี้

$$avg = \frac{0.29166666666 + 0.28082797815}{2} = 0.2862473224$$

$$avg = 0.2862473224$$

เมื่อได้ค่าเฉลี่ยมาแล้วทำการปรับทศนิยมให้เป็น 2 ตำแหน่ง จะได้ 0.28 แล้วนำค่านี้ไปเปรียบเทียบกับค่า 0.25 หากมากกว่าจะถือว่าดวงตาเปิดอยู่ (โดยค่า 0.25 ได้มาจากภาพตัวอย่างการหลับตาและไม่หลับตาจำนวน 60 ภาพ แล้วนำมาหาค่าเฉลี่ยจึงได้ค่า 0.25

3.2.1 การวัดประสิทธิภาพของโปรแกรม

3.2.1.1 ใช้ความถูกต้อง (Accuracy) โดยมีสูตรการคำนวณตามสมการ ได้แก่ True positive rate , False positive rate และค่าความเที่ยงตรง (Precision) หาได้จาก

$$Accuracy = \frac{TPs+TNs}{TPs+TNs+FPs+FNs} \quad (15)$$

Computer Science Department
Faculty of Informatics Mahasarakham University

3.2.1.2 True positive Rate (TPR) คือ ค่าที่บอกว่าโปรแกรมทำนายได้ว่าจริง เป็นอัตราส่วนเท่าไรของจริงทั้งหมด หาได้จาก

$$TPR = \frac{TP}{TP + FN} \tag{16}$$

3.2.1.3 False Positive Rate (FPR) คือ ค่าที่บอกว่าโปรแกรมทำนายว่าจริง เป็นอันตราส่วนเท่าไรของไม่จริงทั้งหมด หาได้จาก

$$FPR = \frac{FP}{TN + FP} \tag{17}$$

3.2.1.4 Precision (ค่าความเที่ยงตรง) เป็นการเปรียบเทียบ การทำนายที่บอกว่า จริง และก็เกิดขึ้นจริง (TP) กับ การทำนายว่า จริง แต่สิ่งที่เกิดขึ้น คือ ไม่จริง หาได้จาก

$$Precision = \frac{TP}{TP + FP} \tag{18}$$

3.2.1.5 Recall ค่าความถูกต้องของการทำนายว่าจะเป็น จริง เทียบกับจำนวนครั้งของเหตุการณ์ทั้งทำนายและเกิดขึ้นว่าเป็นจริง หาได้จาก

$$Recall = \frac{TPs}{TPs + FNs} \tag{19}$$

TP คือ True Positive คือสิ่งที่โปรแกรมทำนายว่าจริง และผลลัพธ์เป็นจริง

TN คือ True Negative คือสิ่งที่โปรแกรมทำนายว่าไม่จริง และผลลัพธ์ไม่เป็นจริง

FN คือ False Positive คือสิ่งที่โปรแกรมทำนายว่าจริง แต่ผลลัพธ์เป็นไม่จริง

FP คือ Fales Negative คือสิ่งที่โปรแกรมทำนายว่าไม่จริง แต่ผลลัพธ์เป็นจริง

Computer Science Department
Faculty of Informatics, Mahasarakham University

3.3 การพัฒนาระบบ

```
10    def calculate_EAR(eye):
11        A = distance.euclidean(eye[1], eye[5])
12        B = distance.euclidean(eye[2], eye[4])
13        C = distance.euclidean(eye[0], eye[3])
14        ear_aspect_ratio = (A+B)/(2.0*C)
15        return ear_aspect_ratio
```

ภาพประกอบที่ 3.23 ฟังก์ชันคำนวณระยะการปิดของดวงตา

3.3.1 บรรทัดที่ 10 ถึง 15

เป็นฟังก์ชันในการคำนวณหาระยะการปิดของดวงตาด้วย Euclidean Distance โดยรับค่าพารามิเตอร์มาเป็นพิกัดจุดตำแหน่ง x y ของดวงตามาเพื่อหาระยะห่างของ A B C แล้วนำไปคำนวณโดยเอา (A+B)/(2.0*C) แล้วทำการ Return ค่ากลับ ดังภาพประกอบที่ 3.23

```
17    cap = cv2.VideoCapture(0)
18    hog_face_detector = dlib.get_frontal_face_detector()
19    dlib_facelandmark = dlib.shape_predictor("shape_predictor_68_face_landmarks.dat")
```

ภาพประกอบที่ 3.24 การนำเข้าข้อมูลและโมเดล

3.3.2 บรรทัดที่ 17 ถึง 19

บรรทัดที่ 17 จะเป็นการนำเข้าข้อมูล หากเป็นจากกล้องเว็บแคมจะใส่ค่าเป็น 0 หากเป็นไฟล์วิดีโอใส่เป็น Path ของไฟล์ ในบรรทัดที่ 18 จะเป็นการเรียกใช้ HOG และบรรทัดที่ 19 เป็นการเรียกใช้โมเดลสำหรับการทำ Facial Landmark เพื่อมาร์กจุดบนใบหน้า ดังภาพประกอบที่ 3.24

```
24            _, frame = cap.read()
25            gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
26            faces = hog_face_detector(gray)
27            for face in faces:
28                face_landmarks = dlib_facelandmark(gray, face)
29                leftEye = []
30                rightEye = []
```

ภาพประกอบที่ 3.25 การแปลงภาพเป็น Gray Scale และมาร์กจุดใบหน้า

Computer Science Department Faculty of Information Technology Rangsit University

3.3.3 บรรทัดที่ 24 ถึง 30

ในบรรทัด 24 จะเป็นการตัดเฟรมของภาพ บรรทัดที่ 25 ทำการแปลงภาพจาก RGB เป็น Gray Scale แล้วนำไปเข้ากระบวนการ HOG ในบรรทัดที่ 28 จะทำการมาร์กจุดบนใบหน้าและสร้างอาร์เรย์ Left Eye, Light Eye สำหรับเก็บพิกัดจุดของดวงตา ดังภาพประกอบที่ 3.25

```
32                  for n in range(36,42):
33                      x = face_landmarks.part(n).x
34                      y = face_landmarks.part(n).y
35                      leftEye.append((x,y))
36                      next_point = n+1
37                      if n == 41:
38                          next_point = 36
39                      x2 = face_landmarks.part(next_point).x
40                      y2 = face_landmarks.part(next_point).y
41                      cv2.line(frame,(x,y),(x2,y2),(0,255,0),1)
42
43                  for n in range(42,48):
44                      x = face_landmarks.part(n).x
45                      y = face_landmarks.part(n).y
46                      rightEye.append((x,y))
47                      next_point = n+1
48                      if n == 47:
49                          next_point = 42
50                      x2 = face_landmarks.part(next_point).x
51                      y2 = face_landmarks.part(next_point).y
52                      cv2.line(frame,(x,y),(x2,y2),(0,255,0),1)
```

ภาพประกอบที่ 3.26 การหาแกน x y ของตาซ้ายและตาขวา

3.3.4 บรรทัดที่ 32 ถึง 52

ในส่วนนี้จะเป็นการหาตำแหน่ง x, y ของแต่ละจุดของดวงตาแล้วนำไปเก็บไว้ในอาร์เรย์ Left Eye, Right Eye โดยบรรทัดที่ 32 ถึง 41 เป็นตาซ้าย และบรรทัดที่ 43 ถึง 52 เป็นตาขวา ดังภาพประกอบที่ 3.26

```
53
54              left_ear = calculate_EAR(leftEye)
55              right_ear = calculate_EAR(rightEye)
56              EAR = (left_ear+right_ear)/2
57              EAR = round(EAR,2)
58              if EAR<0.20:
59                  time=time+1
60                  if time>10:
61                      cv2.putText(frame,"Are you Sleepy?",(20,400),cv2.FONT_HERSHEY_SIMPLEX,2,(0,0,255),4)
62              else:
63                  time=0
```

ภาพประกอบที่ 3.27 การส่งค่าไปฟังก์ชันและตรวจสอบค่าสถานะดวงตา

3.3.5 บรรทัดที่ 54 ถึง 63

ในบรรทัดที่ 54 และ 55 จะทำการส่งค่า Left Eye, Right Eye ไปที่ฟังก์ชัน Calculate_EAR ในบรรทัดที่ 10 ถึง 15 เพื่อคำนวณหาระยะการปิดของดวงตา เมื่อได้ค่าที่ Return กลับมาแล้วนำไปหาค่าเฉลี่ยแล้วตัดทศนิยมเป็น 2 ตำแหน่ง เอาที่ได้ไปเช็คเงื่อนไขหากต่ำกว่าที่กำหนดให้ทำการแจ้งเตือน ดังภาพประกอบที่ 3.27

Computer Science Department
Faculty of Informatics, Mahasarakham University