

การพัฒนาหุ่นยนต์คล้ายมนุษย์ที่มีการเคลื่อนไหวเลียนแบบตา ปาก และใบหน้า Development of Humanoid Robots with Imitative Eye, Mouth, and Facial Movements

กิงกาญจน์ กิตติสุนทรโรภาส¹ อรรถพล พลหา² กิตติกานต์ พูลสุขโข² และวิมลภ อรุณธรรมนา^{3*}

¹สาขาวิชาวิศวกรรมโลจิสติกส์และกระบวนการ คณะเทคโนโลยีอุตสาหกรรม มหาวิทยาลัยราชภัฏราชนครินทร์ kingkam@techno.ru.ac.th

²สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยราชภัฏวชิรเวศน์ 1651010541128@rmuc.ac.th 1651010541150@rmuc.ac.th

³สาขาวิชาเทคโนโลยีทางธุรกิจดิจิทัล คณะบริหารธุรกิจ มหาวิทยาลัยราชภัฏวชิรเวศน์ wanlop.aru@rmu.ac.th

บทคัดย่อ

โครงการนี้นำเสนอการพัฒนาหุ่นยนต์ที่สามารถเลียนแบบการเคลื่อนไหวของตาและปากของมนุษย์ โดยมีวัตถุประสงค์เพื่อให้หุ่นยนต์สามารถแสดงสีหน้าและการเคลื่อนไหวได้ใกล้เคียงกับมนุษย์มากขึ้น ด้วยการนำภาพจากเว็บแคมมาวิเคราะห์ผ่านระบบปัญญาประดิษฐ์ และควบคุมเซอร์โวมอเตอร์ในการเคลื่อนไหวดวงตาและริมฝีปาก โดยใช้ภาษาไพทอน ร่วมกับ โอเพนซีวี และ มาร์คอาร์ชีเอ็นเอ็น ในการตรวจจับใบหน้า งานนี้ยังครอบคลุมถึงการออกแบบฮาร์ดแวร์ที่ใช้บอร์ด Arduino ในการควบคุมการเคลื่อนไหว ผลลัพธ์จากโครงการสามารถนำไปต่อยอดในอุตสาหกรรมบันเทิง หุ่นยนต์ผู้ช่วย หรือหุ่นยนต์เพื่อการศึกษาได้ในอนาคต

คำสำคัญ: หุ่นยนต์, การลอกเลียนแบบใบหน้า, ปัญญาประดิษฐ์

Abstract

This project presents the development of a robot capable of mimicking human eye and mouth movements. The objective is to enable the robot to express facial gestures and movements in a way that closely resembles human behavior. This is achieved by analyzing webcam images through an artificial intelligence (AI) system, which then controls servo motors to move the eyes and lips. The system is implemented using Python in conjunction with OpenCV and Mask R-CNN for facial landmark detection. The project also includes hardware design, utilizing an Arduino board to manage motor control. The outcome of this project can be further applied in entertainment, assistive robotics, or educational robotics in the future.

Keywords: robot, facial mimicry, artificial intelligence

1. บทนำ

ในปัจจุบันเทคโนโลยีหุ่นยนต์ได้เข้ามามีบทบาทสำคัญในหลายด้านของชีวิตประจำวัน ตั้งแต่การทำงานในภาคอุตสาหกรรม การบริการ ไปจนถึงการศึกษา หุ่นยนต์ไม่เพียงแต่ทำงานตามคำสั่งเท่านั้น แต่ยังสามารถตอบสนองต่อสิ่งแวดล้อมและปฏิสัมพันธ์กับมนุษย์ได้อย่างมีประสิทธิภาพ โดยเฉพาะหุ่นยนต์ที่สามารถเคลื่อนไหวได้เหมือนมนุษย์ ซึ่งถูกนำไปใช้อย่างต่อเนื่องในอุตสาหกรรมด้านความบันเทิง เช่น การนำหุ่นยนต์มาเป็นตัวละครในภาพยนตร์ หรือใช้ในสวนสนุก

จากที่กล่าวมาแสดงให้เห็นว่าหุ่นยนต์เริ่มมีบทบาทมากขึ้นในชีวิตประจำวันของมนุษย์ แต่ประเทศไทยยังมีการประยุกต์ใช้หุ่นยนต์

ค่อนข้างน้อย เนื่องมาจากการพัฒนาหุ่นยนต์ในประเทศยังไม่ก้าวหน้า โดยเฉพาะการเลียนแบบสีหน้าท่าทางของมนุษย์ ซึ่งมีความละเอียดอ่อน เช่น การหมุนของศีรษะ และการขยับริมฝีปาก เพื่อให้หุ่นยนต์สามารถลอกเลียนแบบการแสดงออกของมนุษย์ได้อย่างเป็นธรรมชาติ

งานวิจัยนี้จึงมุ่งเน้นการพัฒนาหุ่นยนต์ที่สามารถขยับใบหน้าได้ตรงตามการขยับของมนุษย์ โดยอาศัยการวิเคราะห์ภาพจากกล้องเว็บแคม และนำระบบปัญญาประดิษฐ์เข้ามาช่วยตรวจสอบการเคลื่อนไหวของดวงตาและปาก เพื่อนำไปควบคุมการเคลื่อนไหวของหุ่นยนต์ให้ใกล้เคียงกับมนุษย์มากที่สุด

2. งานวิจัยที่เกี่ยวข้อง

สำหรับงานวิจัยที่เกี่ยวข้อง ประกอบด้วยงานวิจัยที่เกี่ยวข้องกับทางด้านซอฟต์แวร์ และฮาร์ดแวร์ซึ่งจะได้ถูกอธิบายในส่วนตัวไป

งานวิจัยแรกจะใช้อัลกอริทึม Viola-Jones [1] นำเสนอระบบจดจำใบหน้าแบบเรียลไทม์ที่มีประสิทธิภาพแม้ใช้ภาพความละเอียดต่ำเพียง 15 พิกเซล โดยระบบสามารถจดจำใบหน้าได้หลายท่าทางและหลายอารมณ์ ด้วยการพัฒนาและใช้ชุดข้อมูล LRD200 และ LRD100 สำหรับการฝึกสอนและทดสอบระบบประกอบด้วยการตรวจจับใบหน้าด้วยอัลกอริทึม Viola-Jones และการจดจำใบหน้าด้วยอัลกอริทึม Local Binary Pattern Histogram (LBPH) [2] โดยมีการปรับปรุงภาพด้วย CLAHE [3] และการจัดตำแหน่งใบหน้า งานวิจัยต่อมา [4] บทความนี้มุ่งเน้นการศึกษารูปแบบของอารมณ์และพฤติกรรมในการแสดงออกในการควบคุมปฏิสัมพันธ์ทางสังคมระหว่างมนุษย์กับหุ่นยนต์ฮิวแมนนอยด์ โดยเฉพาะในบริบทของการสื่อสารและการเรียนการสอน ผู้วิจัยได้นำเสนอพื้นฐานทางวิทยาศาสตร์ที่ใช้พัฒนาแบบจำลองอารมณ์และพฤติกรรมในการแสดงออกของหุ่นยนต์ พร้อมทั้งแสดงให้เห็นถึงการประยุกต์ใช้แนวคิดเหล่านี้ในการใช้งานจริง หุ่นยนต์สามารถรับรู้อารมณ์ของมนุษย์ผ่านน้ำเสียง โดยอ้างอิงจากงานวิจัยในสาขาจิตวิทยาภาษาศาสตร์เพื่อพัฒนาให้เกิดการรับรู้ที่แม่นยำ ในขั้นแรกมีการประเมินความสามารถของหุ่นยนต์ในการแสดงอารมณ์ในลักษณะแยกส่วน ก่อนจะประเมินการแสดงออกทางอารมณ์แบบองค์รวม ซึ่งรวมถึงการทำงานร่วมกันของระบบการรับรู้อารมณ์ ระบบแรงจูงใจ และระบบการแสดงออก ในขณะที่หุ่นยนต์มีปฏิสัมพันธ์กับมนุษย์โดยตรง บทความนี้จึงเป็นการเน้นถึงความสำคัญของการออกแบบพฤติกรรมทางอารมณ์ในหุ่นยนต์เพื่อส่งเสริมปฏิสัมพันธ์ทางสังคมอย่างมีประสิทธิภาพ งานวิจัยที่สาม [5] ได้กล่าวถึงแพลตฟอร์มหุ่นยนต์ฮิวแมนนอยด์ ซึ่งได้รับการออกแบบมาเพื่อสนับสนุนการวิจัยด้านการเรียนรู้ผ่านการสำรวจอัตโนมัติและการปฏิสัมพันธ์ทางสังคม โดยมีแนวคิดหลักคือการพัฒนาระบบเปิด ทั้งในด้านซอฟต์แวร์และฮาร์ดแวร์ เพื่อส่งเสริมการทำงานร่วมกันและการแบ่งปันทรัพยากรในวงการวิจัย หุ่นยนต์ฮิวแมนนอยด์ถูกออกแบบให้มีคุณสมบัติที่เหมาะสมต่อการวิจัย ได้แก่ ความสามารถในการ

*ผู้ประพันธ์บรรณกิจ

รับรู้และเคลื่อนไหวที่หลากหลาย องค์ประกอบจำนวนมาก สถาปัตยกรรมซอฟต์แวร์ที่ยืดหยุ่นและสามารถนำกลับมาใช้ซ้ำได้ รวมถึงโครงสร้างพื้นฐานที่รองรับการรวมระบบและการทำงานร่วมกันได้ง่าย งานวิจัยสุดท้าย [6] บทความนี้นำเสนอการผสมผสานเทคนิคการเรียนรู้ของเครื่อง (Machine Learning) เข้ากับแอนิเมชันแบบอิงกฎเกณฑ์อย่างง่าย เพื่อพัฒนาหุ่นยนต์แอนดรอยด์ให้สามารถทำหน้าที่เป็นตัวแทนในการสนทนา โดยมี ChatGPT เป็นองค์ประกอบหลักในการสื่อสาร งานวิจัยได้อธิบายรายละเอียดเกี่ยวกับโครงสร้างของหุ่นยนต์ เทคนิคการสร้างแอนิเมชันลิปซิงค์ (lip-sync) และแนวทางการออกแบบซอฟต์แวร์ที่ใช้ นอกจากนี้ยังได้นำเสนอผลจากการเผยแพร่ระบบต่อสาธารณะ ซึ่งทำให้เห็นถึงจุดที่สามารถพัฒนาเพิ่มเติม และนำไปสู่การปรับปรุงระบบในลักษณะวนซ้ำ (iterative design) เพื่อเพิ่มประสิทธิภาพในการใช้งานจริง.

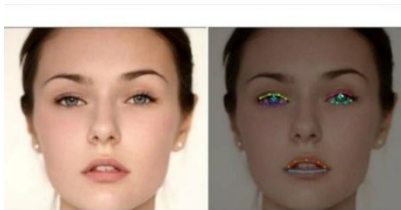
สำหรับงานวิจัยที่นำเสนอในเอกสารนี้ เป็นการพัฒนาระบบปัญญาประดิษฐ์ให้ทำงานร่วมกับหุ่นยนต์ โดยการตรวจจับใบหน้าของมนุษย์เพื่อจะควบคุมการให้หุ่นยนต์ ลอกเลียนแบบการขยับของดวงตา ริมฝีปาก และการหันใบหน้า โดยการขยับของมนุษย์จะตรวจจับได้จากกล้องเว็บแคม

3. วิธีการนำเสนอ

วิธีการที่นำเสนอในงานวิจัยนี้จะประกอบไปด้วย 2 ส่วน คือ ส่วนของซอฟต์แวร์ และส่วนของฮาร์ดแวร์

3.1 ส่วนของซอฟต์แวร์

ส่วนของซอฟต์แวร์ประกอบไปด้วย 3 ส่วนย่อยคือส่วนของการควบคุมดวงตา ปาก และใบหน้าของหุ่นยนต์ ซึ่งจะใช้ปัญญาประดิษฐ์ที่ทำการฝึกฝนสำหรับการตรวจจับโดยการมาร์คจุดใบหน้าดังรูปที่ 3.1 แล้วจึงนำไปฝึกฝนผ่าน MaskRCNN [7]



รูปที่ 3.1 การมาร์คตำแหน่งดวงตา และริมฝีปาก

ส่วนของการควบคุมดวงตาจะทำการตรวจสอบตำแหน่งของตาดำที่ได้จากกล้องโดยใช้ไลบรารีสำหรับจับตำแหน่งตาดำของ OpenCV [8] ในการตรวจจับ โดยเทียบกับตำแหน่งของขอบตาซ้ายกับขอบตาขวา และขอบตาบนกับขอบตาล่าง เพื่อหาระยะห่างโดยจะกำหนดระหว่างตำแหน่งขอบตากับตำแหน่งของตาดำ จะแทนค่าด้วยตัวแปร dis

โดยในแนวนอนจะใช้ตำแหน่งแลนด์มาร์คของขอบตาด้านซ้ายและขอบตาด้านขวามาหาระยะห่างระหว่างทั้งคู่ซึ่งจะเป็นค่าพิทเชล จากนั้นจะนำค่าที่ได้ไปทำการนอร์มัลไลเซชัน (Normalization) ให้มีค่าระหว่าง 0.00 ถึง 1.00 ดังรูปที่ 3.2 แล้วจึงใช้ตำแหน่งของตาดำมาวัดระยะการหันของดวงตาในแนวนอนแล้วเก็บค่าไว้ตั้งเงื่อนไข (1)

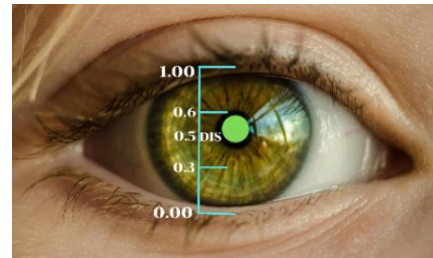
$$G_{lx}, G_{rx} = \begin{cases} 1, & \text{if : } dis < 0.43 \\ 2, & \text{if : } 0.43 \leq dis \leq 0.57 \\ 3, & \text{if : } dis > 0.57 \end{cases} \quad (1)$$



รูปที่ 3.2 ระยะห่างระหว่างขอบตาดำถึงตาดำในแนวนอน

ในแนวนอนตั้งจะใช้ระยะห่างที่ได้จากตำแหน่งแลนด์มาร์คของขอบตาด้านล่างกับตำแหน่งแลนด์มาร์คของขอบตาบนที่ผ่านการนอร์มัลไลเซชัน (Normalization) ให้มีค่าตั้งแต่ 0.00 ถึง 1.00 ดังรูปที่ 3.3 แต่จะใช้การวัดตำแหน่งตาดำในแนวนอนตั้ง และใช้เงื่อนไขที่แตกต่างจากแนวนอนตั้งเงื่อนไขต่อไปนี้ (2)

$$G_{ry}, G_{ly} = \begin{cases} 1, & \text{if : } dis > 0.6 \\ 2, & \text{if : } 0.3 \leq dis \leq 0.6 \\ 3, & \text{if : } dis < 0.3 \end{cases} \quad (2)$$



รูปที่ 3.3 ระยะห่างระหว่างขอบตาดำถึงตาดำในแนวนอนตั้ง

โดยจะใช้ตำแหน่งของดวงตาในแนวนอนแนวนอนกับแนวนอนตั้งของดวงตาด้านซ้ายและขวาที่ตรวจจับได้จากกล้องเว็บแคม มาหารเพื่อหาค่าเฉลี่ยที่กำหนดค่าตำแหน่งของดวงตาทั้งสองข้าง จากสมการที่ (3) และ (4)

$$x = \left[\frac{G_{lx} + G_{rx}}{2} \right] \quad (3)$$

$$y = \left[\frac{G_{ly} + G_{ry}}{2} \right] \quad (4)$$

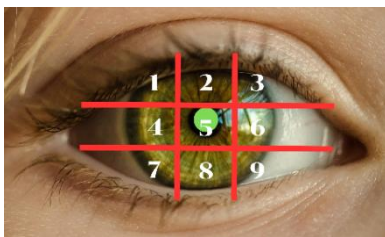
แล้วจึงกำหนดค่าตำแหน่งของดวงตาที่แสดงในแนวนอน และตำแหน่งของดวงตาที่แสดงในแนวนอนตั้งที่ได้ ไปคำนวณหาตำแหน่งของดวงตาที่จะแสดงออกมา จากสมการที่ (5)

$$pos = (y-1) \times 3 + x \quad (5)$$

โดยที่

G_{ix}, G_{rx} คือ ตำแหน่งที่ตรวจจับได้ในแนวนอนของตาซ้าย/ขวา
 G_{iy}, G_{ry} คือ ตำแหน่งที่ตรวจจับได้ในแนวตั้งของตาซ้าย/ขวา
 x คือ ตำแหน่งของลูกตาที่แสดงในแนวนอน
 y คือ ตำแหน่งของลูกตาที่แสดงในแนวตั้ง
 pos คือ ตำแหน่งของลูกตาที่จะแสดงออกมา
 dis คือ ค่าระยะห่างระหว่างขอบตาถึงตา

ตำแหน่งของลูกตาที่คำนวณออกมาได้ตั้งแต่ 1 ถึง 9 จะถูกใช้ในการโปรแกรมผ่านบอร์ดอาดูโน (Arduino) เพื่อควบคุมการขยับดวงตาของหุ่นยนต์ ซึ่งมีทั้งหมด 9 รูปแบบ ดังรูปที่ 3.4



รูปที่ 3.4 ตำแหน่งที่ดวงตาจะแสดงออกมา

ตารางที่ 3.1 ตำแหน่งของดวงตา

$\begin{matrix} \backslash & X \\ y \end{matrix}$	1 (ซ้าย)	2 (กลาง)	3 (ขวา)
1 (บน)	1	2	3
2 (กลาง)	4	5	6
3 (ล่าง)	7	8	9

ในส่วนของการควบคุมการขยับปาก จะใช้ระบบการเรียนรู้เชิงลึกโมเดล MaskRCNN ที่ทำการฝึกฝนไว้สำหรับตรวจจับตำแหน่งของริมฝีปากบน และตำแหน่งของริมฝีปากล่างที่ได้จากกล้องเว็บแคม นำมาหาระยะห่างระหว่างตำแหน่งของริมฝีปากบนกับตำแหน่งของริมฝีปากล่าง นำมาคำนวณโดยใช้สมการที่ (6) เพื่อหาอัตราขยายสำหรับหาค่าระยะห่างจริงของปากบนและปากล่าง

$$s = \frac{w_r}{w_p} \quad (6)$$

เมื่อได้อัตราขยายสำหรับคำนวณระยะห่างจริงระหว่างปากบนและปากล่างแล้วจะนำค่าอัตราขยายที่ได้ไปคำนวณกับค่าระยะห่างระหว่างริมฝีปากที่ตรวจจับได้จากการเรียนรู้เชิงลึกที่ได้ฝึกฝนไว้ โดยใช้สมการที่ (7)

$$d_r = d_p \cdot s \quad (7)$$

โดยที่

w_p คือ ความกว้างของริมฝีปากล่างในภาพ (พิกเซล)

w_r คือ ความกว้างจริงของริมฝีปากล่าง (เซนติเมตร)

s คือ อัตราขยาย

d_p คือ ระยะทางระหว่างริมฝีปากบนและล่าง (พิกเซล)

d_r คือ ระยะห่างจริงของริมฝีปากบน-ล่าง (เซนติเมตร)

นำค่าระยะห่างจริงของริมฝีปากบน-ล่าง ดังรูปที่ 3.5 ที่ได้มาเทียบเงื่อนไข เพื่อใช้คำนวณค่าที่จะส่งไปยังบอร์ดอาดูโน (Arduino) สำหรับควบคุมเซอร์โวมอเตอร์ให้ขยับส่วนของปากหุ่นยนต์ ถ้าค่าระยะห่างจริงของริมฝีปากที่คำนวณมีค่าน้อยกว่าหรือเท่ากับ 0.37 เซนติเมตร จะกำหนดให้เซอร์โวมอเตอร์ไม่มีการเคลื่อนไหว แต่ค่าที่ได้มีค่ามากกว่า 0.37 เซนติเมตร จะนำค่าระยะห่างจริงของริมฝีปากบนและริมฝีปากล่างที่ได้ไปคำนวณตามสมการที่ (8)



รูปที่ 3.5 ตัวอย่างการแสดงผลค่าระยะห่างจริง

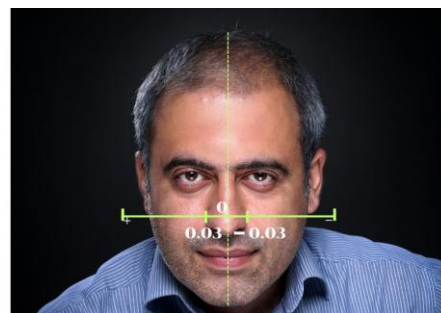
$$mouse = \begin{cases} 0, & \text{if } d_r \leq 0.37 \\ (100 \cdot d_r) - 20, & \text{if } d_r > 0.37 \end{cases} \quad (8)$$

ค่าที่ได้จะนำไปใช้กำหนดองศาที่จะขยับของเซอร์โวมอเตอร์เพื่อควบคุมการอ้าปากของหุ่นยนต์ตั้ง

ส่วนสุดท้าย ส่วนของการขยับใบหน้าหุ่นยนต์ จะใช้ไลบรารีสำหรับกำหนดตำแหน่งแลนดมาร์คใบหน้าของ OpenCV เพื่อกำหนดตำแหน่งของโหนดแก้มซ้าย และโหนดขวาที่ได้จากกล้องสำหรับหาค่ากึ่งกลางระหว่างทั้งสอง โดยใช้สมการที่ (9)

$$x_{center} = \frac{x_{left} + x_{right}}{2} \quad (9)$$

กำหนดให้ปลายจมูกของใบหน้าที่ตรวจจับได้จากกล้อง เป็นจุดศูนย์กลางสำหรับหาว่าจุดของปลายจมูกมีการเคลื่อนที่ไปทิศทางใดมากกว่ากันระหว่างซีกซ้าย กับซีกขวาดังรูปที่ 3.6 เพื่อใช้ในการควบคุมการหันหน้าของหุ่นยนต์ว่าจะหันไปยังทิศทางไหน ระหว่างทิศทางซ้าย ขวา หรือตรงกลาง จากสมการที่ (10)



รูปที่ 3.6 ตัวอย่างการแสดงผลค่าระยะห่างจริง

$$\Delta x = x_{nose} - x_{center} \quad (10)$$

โดยที่

x_{center} คือ ตำแหน่งกึ่งกลางของใบหน้า

x_{nose} คือ ตำแหน่งของปลายจมูก

x_{left} คือ ตำแหน่งแกนนอนของแก้มซ้าย

x_{right} คือ ตำแหน่งแกนนอนของแก้มขวา

Δx คือ ตำแหน่งในการหันใบหน้า

การกำหนดทิศทางในการหันของใบหน้าจากค่าตำแหน่งในการหัน
ใบหน้าที่ได้คำนวณได้ โดยใช้เงื่อนไขในขั้นต่อไป

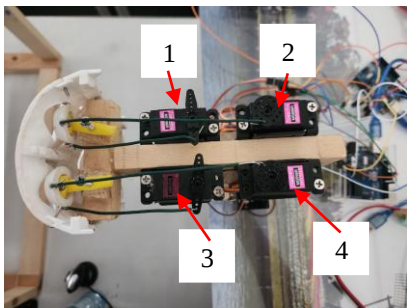
$$face = \begin{cases} right, & \text{if } \Delta x > 0.03 \\ center, & \text{if } -0.03 \leq \Delta x \leq 0.03 \\ left, & \text{if } \Delta x < -0.03 \end{cases} \quad (11)$$

หากค่าตำแหน่งในการหันใบหน้าที่ได้มีค่ามากกว่า 0.03 จะควบคุมเซอร์โวมอเตอร์ให้หันหน้าหุ่นยนต์ไปทางขวา และหากค่าที่ได้มีค่าน้อยกว่า -0.03 เซอร์โวมอเตอร์จะควบคุมหุ่นยนต์ให้หันไปทางซ้าย กรณีที่ไม่อยู่ใน 2 เงื่อนไขก่อนหน้าจะควบคุมให้เซอร์โวมอเตอร์กลับมาอยู่ที่ตำแหน่งกึ่งกลาง

3.2 ส่วนของฮาร์ดแวร์

ส่วนของฮาร์ดแวร์ประกอบไปด้วย 4 ส่วนย่อย คือ การขับเคลื่อนของหุ่นยนต์ การอาร์มฟลิปของหุ่นยนต์ การหันใบหน้าของหุ่นยนต์ และระบบวงจรของหุ่นยนต์ โดยแต่ละส่วนจะถูกอธิบายในส่วนต่อไป

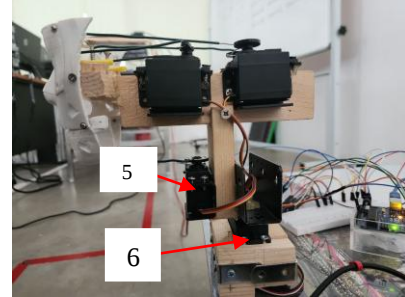
ในส่วนของการขับเคลื่อนจะมีการใช้เซอร์โวมอเตอร์ทั้งหมด 4 ตัว แบ่งเป็นข้างละ 2 ตัวเพื่อควบคุมการขับเคลื่อนของดวงตา ดังรูปที่ 3.7 โดยเซอร์โวมอเตอร์ตัวที่ 1 กับ 3 ใช้เพื่อควบคุมตำแหน่งของดวงตาในแนวแกนนอน และเซอร์โวมอเตอร์ตัวที่ 2 กับ 4 ใช้สำหรับควบคุมตำแหน่งของดวงตาในแนวแกนตั้ง ตามค่าที่กล้องเว็บแคมตรวจจับและผ่านการเช็คเงื่อนไขเพื่อให้ดวงตาสามารถเคลื่อนไหวตามค่าที่ส่งมาได้ทั้งหมด 9 ตำแหน่ง



รูปที่ 3.7 ตัวอย่างโครงสร้างหุ่นยนต์จากด้านบน

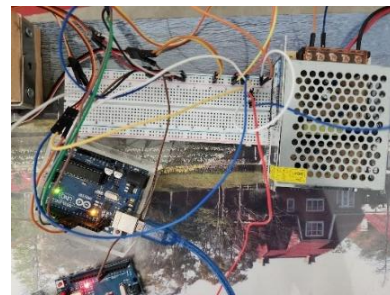
ต่อมาในส่วนของการปากจะใช้เซอร์โวมอเตอร์ตัวที่ 5 เพียงตัวเดียวในการปรับองศาความกว้างของปาก โดยจะใช้ค่าที่รับมา ควบคุมเซอร์โวมอเตอร์ให้ปรับความกว้างตั้งแต่ 0 องศา ไปจนถึง 60 องศา

ส่วนของการขยับใบหน้า จะใช้เซอร์โวมอเตอร์ตัวที่ 6 ซึ่งเป็นรุ่น DM4000 ที่มีขนาดใหญ่กว่าเพื่อให้มีแรงหมุนมากพอที่จะหันใบหน้าของหุ่นยนต์ ในแนวแกนนอน ได้ 3 ตำแหน่ง คือ ซ้าย กลาง และขวา ดังรูปที่ 3.8



รูปที่ 3.8 ตัวอย่างโครงสร้างหุ่นยนต์จากด้านข้าง

ในส่วนสุดท้ายส่วนวงจรของหุ่นยนต์ จะมีการใช้บอร์ดอา두โน (Arduino) สำหรับรับค่าที่คำนวณไว้ในส่วนของซอฟต์แวร์จากคอมพิวเตอร์ โดยจะต่อสายวงจรเข้ากับเบรตบอร์ด (breadboard) เพื่อควบคุมเซอร์โวมอเตอร์ทั้ง 6 ตัว จากค่าที่ได้คำนวณไว้และจ่ายไฟให้กับเซอร์โวมอเตอร์ทั้งหมดผ่านพาวเวอร์ซัพพลาย (Power Supply) ดังรูปที่ 3.9



รูปที่ 3.9 ตัวอย่างวงจรของหุ่นยนต์

4. ผลการทดลอง

จะทำการทดลองโดยให้ผู้ทดลองใช้กล้องเว็บแคมเพื่อตรวจจับใบหน้าของผู้ทดลอง เพื่อทำการขยับส่วนต่างๆของหุ่นยนต์จากค่าที่ได้จากการตรวจจับดังรูปที่ 4.1 สำหรับการทดสอบผลในแต่ละส่วนให้มีความพร้อมสมบูรณ์ก่อนการทำงานร่วมกัน จะได้อธิบายในส่วนถัดไป

4.1 การทดลองการเคลื่อนไหวของดวงตา

การที่ดวงตาจะเคลื่อนไหวได้จะต้องถูกควบคุมด้วยเซอร์โวมอเตอร์ข้างละ 2 ตัว สำหรับควบคุมการขับเคลื่อนของดวงตาในแนวแกนนอน และแนวแกนตั้ง

จากการทดลองโดยให้ผู้ทดลองขยับดวงตาไปในองศาต่างๆเพื่อหาประสิทธิภาพการขับเคลื่อนของดวงตาได้ผลดังนี้ การขยับของดวงตาในแนวแกนนอนในรูปหมายเลข 4 กับ 6 และ ในแนวแกนตั้งในรูปหมายเลข 2 กับ 8 ดังรูปที่ 4.1 จากการทดสอบการขยับของผู้ทดสอบจะพบว่ามีความแม่นยำสูง และมีความคลาดเคลื่อนเพียงเล็กน้อย แต่กับการขยับดวงตาในแนวเฉียงจะมีความคลาดเคลื่อนมากกว่าโดยเฉพาะการขยับดวงตาในแนวเฉียงลงในรูปหมายเลข 7 กับ 9 ที่จะมีความคลาดเคลื่อนมากกว่าแนวเฉียงขึ้นในรูปที่ 1 กับ 3 เนื่องจากการหันลูกตามองลงจะทำ

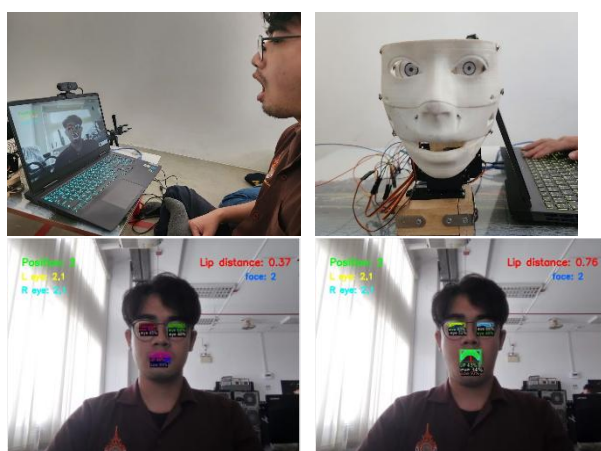
ให้หนังตาปิดลงมา ทำให้การตรวจจับดวงตาเป็นไปได้ยากกว่าจึงทำให้การขยับลูกตาลงในแนวเฉียงมีความคลาดเคลื่อนสูงกว่าการมองขึ้นที่จะทำให้สามารถตรวจจับดวงตาได้ชัดเจน



รูปที่ 4.1 การขยับดวงตา

4.2 การทดลองการเคลื่อนไหวของปาก

จากการทดสอบโดยการตรวจจับการอ้าปากของผู้ทดสอบโดยใช้ระบบการเรียนรู้เชิงลึกที่ทำการฝึกฝน ได้ผลการทดสอบว่าโมเดลสามารถตรวจจับการขยับของปากแบบเรียลไทม์ได้อย่างแม่นยำ จึงสามารถนำไปใช้กำหนดให้เซอร์โวมอเตอร์หมุนในส่วนกรามของหุ่นยนต์เพื่อขยับปากได้ระหว่าง 0 องศา ถึง 90 องศา โดยมีตัวอย่างการอ้าปากของหุ่นยนต์ดังรูปที่ 4.2



รูปที่ 4.2 ตัวอย่างการขยับปาก

4.3 การหันใบหน้า

จากการทดสอบโดยให้ผู้ทดสอบทำการหันหน้าไปทางซ้ายและทางขวาให้กล้องเว็บแคมตรวจจับเพื่อหาความแม่นยำในการตรวจจับของโมเดลโดยที่เมื่อหันไปทางซ้ายจะแสดงผลเป็นเลข 1 และการหันไปทางขวาจะแสดงผลเป็นหมายเลข 2 ได้ผลการทดสอบว่าสามารถตรวจจับใบหน้าของผู้ทดสอบแม่นยำ และไม่มีมีความคลาดเคลื่อนเลย ทำให้สามารถใช้ในการควบคุมการหันหน้าของหุ่นยนต์ได้อย่างแม่นยำ และมีระยะที่ใช้สำหรับควบคุมการหันใบหน้าของหุ่นยนต์ไปทางซ้ายและขวาได้ฝั่งละ 60 องศา ทำให้หุ่นยนต์สามารถขยับศีรษะในแนวแกนอนได้ 3 รูปแบบรูปแบบ คือ หน้า

ตรง ทางซ้าย และทางขวา ในองศาที่เท่ากันโดยใช้เซอร์โวมอเตอร์รุ่น DM4000 ในการหมุนส่วนคอของหุ่นยนต์ดังรูปที่ 4.3



รูปที่ 4.3 รูปตัวอย่างการหันหน้าไปทางด้านขวา

4.4 การเก็บผลการทดลอง

การเก็บผลการทดลองโดยให้ผู้ทดสอบจำนวน 30 คน มาทดสอบกับกล้องเว็บแคม เพื่อเช็คการตรวจจับใบหน้า ดวงตา และริมฝีปากของผู้ทดสอบในเวลาเดียวกัน และสามารถส่งค่าไปยังบอร์ดได้

ตารางที่ 4.1 ตารางเก็บผลการทดลอง

รูป	การตรวจจับใบหน้า	การตรวจจับดวงตา	การตรวจจับริมฝีปาก	การขยับหุ่นยนต์
	✓	✓	✓	✓
	✓	✓	✓	✓
	✓	✓	✓	✓
	✓	✓	✓	✓
	✓	✓	✓	✓
	✓	✓	✓	✓
	✓	✓	✓	✓
	✓	✓	✓	✓
	✓	✓	✓	✓

	✓	✓	✓	✓
	✓	✓	✓	✓
	✓	✓	✓	✓
	✓	✓	✓	✓
	✓	✓	✓	✓
	✓	✓	✓	✓
	✓	✓	✓	✓
	✓	✓	✓	✓
	✓	✓	✓	✓
	✓	✓	✓	✓
	✓	✓	✓	✓
	✓	✓	✓	✓
	✓	✓	✓	✓
	✓	✓	✓	✓
	✓	✓	✓	✓
	✓	✓	✓	✓
	✓	✓	✓	✓
	✓	✓	✓	✓
	✓	✓	✓	✓

5. สรุป

การพัฒนาหุ่นยนต์ลอกเลียนแบบการเคลื่อนไหวศีรษะของมนุษย์ เป็นงานวิจัยที่จัดทำขึ้นเพื่อส่งเสริมการพัฒนาหุ่นยนต์ในอนาคต และช่วยส่งเสริมการเรียนรู้และวิจัยในด้านหุ่นยนต์และปัญญาประดิษฐ์ จากการ

พัฒนาหุ่นยนต์ลอกเลียนแบบการเคลื่อนไหวศีรษะของมนุษย์ การพัฒนาโมเดลปัญญาประดิษฐ์สำหรับตรวจจับดวงตา ริมฝีปาก และใบหน้า เพื่อมาใช้ในการควบคุมการเคลื่อนไหวศีรษะของหุ่นยนต์ และได้ทำการทดลองโดยใช้กล้องเว็บแคมในการตรวจจับใบหน้าของผู้ทดลองจำนวน 30 คน เพื่อควบคุมศีรษะของหุ่นยนต์ ได้ผลการทดสอบ พบว่าหุ่นยนต์สามารถเคลื่อนไหวตามการขยับของผู้เข้าทดสอบทุกคนได้อย่างแม่นยำ

6. อ้างอิง

- [1] Viola, P., & Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition.
- [2] Yann Rodriguez, & Sebastien Marcel, (2006). Face Authentication Using Adapted Local Binary Pattern Histograms.
- [3] Zuiderveld, K. (1994). Contrast Limited Adaptive Histogram Equalization. In Hackbert, P. S. (Ed.). Graphics Gems IV (p. 474-485). San Diego, California: Academic Press.
- [4] Breazeal, C. (2003). Emotion and sociable humanoid robots. International Journal of Human-Computer Studies.
- [5] Metta, G. et al. (2010). The iCub humanoid robot: an open-systems platform for research in cognitive development. Neural Networks.
- [6] Marcel Heisler Christian Becker-Asano (2023). "An Android Robot Head as Embodied Conversational Agent" Thu, 18 May 2023
- [7] Kaiming He, Georgia Gkioxari, Piotr Dollar, Ross Girshick. (2017). Mask R-CNN Proceedings of the IEEE International Conference on Computer Vision (ICCV).
- [8] G. Bradski, (2000). 'The OpenCV Library', Dr. Dobb's Journal of Software Tools.