

ระเบียบวิธีทางคอมพิวเตอร์สำหรับการมองเห็นสีของการคั่วเมล็ดกาแฟโดยใช้ออเพนซีวี  
**Computational method for color vision of coffee bean roasting**  
**using openCV**

ชัยณรงค์ วิเศษศักดิ์วิชัย<sup>1\*</sup>, ประเสริฐ แผ่ழู<sup>2</sup>,  
เอกพล อันสุสเรนทร์<sup>1</sup>, ชูตักษิร์ กลมลขันติธิร<sup>1</sup>, วินัย เมธาวิทิต<sup>1</sup>  
Chainarong Wisassakwichai<sup>1\*</sup>, Prasert Phaochoo<sup>2</sup>,  
Ekkapol Anusurain<sup>1</sup>, Choosak Kamonkhantithorn<sup>1</sup>, Winai Methavithit<sup>1</sup>

<sup>1</sup>สาขาวิชาชีวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีราชมงคลกรุงเทพ

<sup>2</sup>สาขาวิชาคณิตศาสตร์ คณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยเทคโนโลยีราชมงคลกรุงเทพ

<sup>1</sup>Division of Electrical Engineering, Faculty of Engineering, Rajamangala University of Technology Krungthep

<sup>2</sup>Division of Mathematics, Faculty of Science and Technology, Rajamangala University of Technology Krungthep

\*Corresponding author. Tel.: 08 1343 7693 E-mail: chainarong.w@mail.rmutk.ac.th

## บทคัดย่อ

การพัฒนาซอฟต์แวร์สำหรับระบบคอมพิวเตอร์วิทัค์น์ ให้มีความอิสระต่อคอมพิวเตอร์แพลตฟอร์ม สามารถทำได้สำเร็จโดยการใช้ภาษาการโปรแกรมที่เหมาะสม บทความนี้จึงได้นำเสนอ ระเบียบวิธีทางคอมพิวเตอร์สำหรับการมองเห็นสีของการคั่วเมล็ดกาแฟ ใช้ทรัพยากรเปิดของภาษาการโปรแกรมชีพลัสพลัส ภาษาพิจิตอลจะถูกปฏิบัติการประมาณผลบนไฟล์ข้อมูลภาพบันทึกอยู่ในหน่วยความจำสำรองของระบบคอมพิวเตอร์ วัตถุเชิงโปรแกรมของไฟล์ข้อมูลภาพถูกสร้างขึ้นด้วยไลบรารีของออเพนซีวี (OpenCV, open source computer vision library) โดยใช้ปริภูมิสีแบบเชอเอสวี (HSV, hue-saturation-value color space) โดยที่ซองสีต่าง ๆ จะแยกออกจากกันในหน่วยวัตถุของการโปรแกรมเชิงวัตถุ การปรับเปลี่ยนสโตแกรมจะดำเนินการบนหน่วยวัตถุที่เลือกสำหรับการແ劈ขยายค่าความเข้ม ก่อนที่ทั้งหมดจะถูกจำลองด้วยการแจกแจงความน่าจะเป็นไม่ต่อเนื่องของความเข้มสีน้ำเงิน สีเขียว และสีแดง ตัวแทนสีเชิงตัวเลขของเมล็ดกาแฟคั่วจะถูกกำหนดด้วยค่าคาดหมายและค่าส่วนเบี่ยงเบนมาตรฐานของตัวแปรสุ่ม โปรแกรมที่พัฒนาขึ้นสามารถปฏิบัติการได้ทั้งเครื่องคอมพิวเตอร์ส่วนบุคคลและคอมพิวเตอร์บอร์ดเดียวแบบraspberrypi ที่ใช้ระบบปฏิบัติการwinдовส์และใช้ระบบปฏิบัติการลีนุกซ์ ตามลำดับ

**คำสำคัญ:** คอมพิวเตอร์วิทัค์น์ การประมาณผลภาพพิจิตอล วัตถุเชิงโปรแกรม ปริภูมิสีแบบเชอเอสวี การปรับเปลี่ยนสโตแกรม การแจกแจงความน่าจะเป็น ค่าคาดหมาย ตัวแปรสุ่ม

Received 24-04-2019  
Revised 17-06-2019  
Accepted 24-06-2019

## Abstract

Development of software for computer vision system which does not depend upon computer platforms can be accomplished by appropriate programming language. This paper presents the computational method for color vision of coffee bean roasting using the open source of C++ programing language. Digital image processing is performed on the image file recording in secondary memory unit of computer system. The programmatic object of image file is created by the library of OpenCV (open source computer vision) using HSV (hue-saturation-value) color space where the color channels separate into the instances of object-oriented programming. The histogram equalization operates on selected instance for intensity stretching before all of them are modeled with discrete probability distribution of color intensity in red green and blue. The numerical color representations of roast coffee beans are determined by expected values and standard deviation of random variables. The developed program can execute in both personal computer and Raspberry Pi computer single board with Widows and Linux operating system respectively.

**Keywords:** Computer vision, Digital image processing, Programmatic object, HSV color space, Histogram equalization, Probability distribution, Expected value, Random variables

## 1. บทนำ

การประมวลผลภาพดิจิตอลให้เอาร์พุต เป็นข้อมูลเข้าสู่กระบวนการแปลความหมาย เพื่อคอมพิวเตอร์จะมีความสามารถเห็นความแตกต่างของวัตถุ จำแนกประเภท ตลอดจนสามารถจัดเรียงวัตถุได้ตามขนาด ทั้งหมดนับได้เป็นกระบวนการทำงานของระบบคอมพิวเตอร์วิทัศน์ ดังจะพูดได้ในการควบคุมหุ่นยนต์งานอุตสาหกรรมแบบเลือกและวาง (pick and place) ได้ใช้คอมพิวเตอร์วิทัศน์ในการกำหนดเส้นทางการเคลื่อนที่ [1] ในที่ซึ่งกล้องจะถูกติดตั้งไว้หน้าพื้นที่ทำงาน เพื่อเชื่อมโยงภาพเข้ากับโปรแกรมประมวลผลภาพ การรู้จำวัตถุ เพื่อถ่ายโอนวัตถุจากระบบพิกัดของจุดภาพไปเป็นระบบพิกัดทำงานจริง แต่สำหรับงานทางด้านประมินคุณภาพอาหารได้มีการแบ่งเป็นลำดับขั้นการทำงานตั้งแต่ [2-3]

การจัดหาภาพ การประมวลผลเบื้องต้น การแยกบริเวณภาพ การเลือกหรือการสกัดลักษณะสมบัติเฉพาะ และการจัดประเภท ภาพดิจิตอลถูกจำลองในลักษณะแวดล้อม 2 มิติ ของความเข้มสีกำหนดค่าเชิงตัวเลข  $f(x, y)$  เมื่อคู่อันดับของจำนวนเต็ม  $(x, y)$  คือพิกัดเจิงพื้นที่ (Spatial coordinate) ของจุดภาพ [4] การวิเคราะห์สีของอาหารเพื่อกำหนดคุณภาพ จะมีการแยกภาพดิจิตอลออกเป็นช่องสีในปริภูมิเอกอัครสี [5-6] ทำให้สามารถกำหนดเส้นแบ่งเขต (Threshold) ในช่องสีส่วนประกอบของปริภูมิสีเพื่อให้เกิดการแยกบริเวณสี (Color segmentation) สำหรับกำจัดส่วนพื้นหลังออกจากส่วนบริเวณภาพที่ต้องการวิเคราะห์ภาพดิจิตอลซึ่งมีจำนวนจุดภาพ  $N$  จุดและความเข้มสีกำหนดเชิงตัวเลขอยู่ในช่วงปิด  $f_k(x_i, y_j) \in [0, L]$  เมื่อ  $k = 0, 1, 2, \dots, L$

ขณะ ที่  $i, j \in [0, N-1]$  จะสามารถคำนวณค่าสีสโตแกรมได้ในลักษณะ

$$h(f_k) = \sum_{n=1}^{n_k} 1$$

เมื่อ  $n_k$  คือจำนวนจุดภาพที่มีความเข้มสีกำหนดค่าเชิงตัวเลขเท่ากับ  $f_k \Delta f_k(x_i, y_j)$  Laurent และคณ [7] แยกภาพดิจิตอลออกเป็นช่องสีเพื่อคำนวณสีสโตแกรม ซึ่งพบว่าการประมวลผลภาพโดยใช้สีสโตแกรมสี มีแบบรูปการเปลี่ยนแปลงอย่างมีนัยสำคัญสัมพันธ์กับความยากในการปรุงสุกของเม็ดถั่วแขก

เป้าหมายของงานในที่นี้คือ พัฒนาระเบียบวิธีทางคอมพิวเตอร์กำหนดค่าสีตัวแทนของภาพเมล็ดกาแฟจากการค่าวัด โดยใช้ค่าคาดหมายและค่าส่วนเบี่ยงเบนมาตรฐานของตัวแปรสุ่ม บ่งชี้ภาพสีตัวแทนที่สามารถนำไปใช้ในการเปรียบเทียบกับค่าสีระดับของการค่าวัดได้ก่อน สำหรับการพัฒนาระบบคอมพิวเตอร์วิทัศน์ของการค่าวัดเมล็ดกาแฟได้ทั้งบนเครื่องคอมพิวเตอร์ส่วนบุคคลที่ใช้ระบบปฏิบัติการวินโดว์ส และคอมพิวเตอร์บอร์ดเดียวแบบraspberrypiที่ใช้ระบบปฏิบัติการลีนุกซ์

### 1.1 หลักการที่เกี่ยวข้อง

การวางแผนมาตรฐานทางสีของ CIE (Commission internationale de l'Eclairage) ได้กำหนดสามสีปฐมภูมิโดยใช้ค่าความยาวคลื่นได้แก่ 435.8 , 546.1 และ 700 nm สำหรับ สีน้ำเงิน สีเขียว และสีแดง ตามลำดับ

#### 1.1.1 ปริภูมิสี

ปริภูมิสีอาร์จีบี (RGB color space) เป็นลูกบาศก์หนึ่งหน่วยที่แต่ละแกนพิกัดคือสีปฐมภูมิ เรียงลำดับสีแดง สีเขียว และ สีน้ำเงิน จุดสีดำคือ ตำแหน่งของจุดกำเนิด จุดภายในลูกบาศก์คือสี

ต่าง ๆ กำหนดจากพิกัดค่าสีปฐมภูมิที่จุดนั้น กรณีที่ใช้เลขฐาน 2 จำนวน 8 บิต ในแต่ละสีปฐมภูมิ สีแดงสว่าง (Bright red) มีพิกัด (255, 0, 0) สีเขียวสว่าง (Bright green) มีพิกัด (0, 255, 0) และสีน้ำเงินสว่าง (Bright blue) มีพิกัด (0, 0, 255) แนวเส้นทแยงจากจุด (0, 0, 0) ไปยัง (255, 255, 255) คือโทนสีเทา (Grey scale) สำหรับภาพขนาด  $m \times n$  ที่จุดใด ๆ ในปริภูมิอาร์จีบีสามารถส่งไปยังแนวเส้นทแยงโทนสีเทาด้วยภาพฉายของเวกเตอร์สามมิติ

$$\text{gray} = [r \ g \ b][1/\sqrt{3} \ 1/\sqrt{3} \ 1/\sqrt{3}]^T$$

เมื่อจำนวนเต็ม  $r, g, b \in [0, 255]$  คือพิกัดสีในปริภูมิ ทำการปรับขนาดโทนสีเทา gray ให้อยู่ในช่วงปิด  $[0, 255]$  ดังนี้

$$\text{Gray} = \text{gray} \left( \frac{255}{255\sqrt{3}} \right) = \frac{1}{\sqrt{3}} \text{gray}$$

ดังนั้นจึงสามารถคำนวณโทนสีเทา  $\text{Gray} \in [0, 255]$  ให้กับพิกัดสีในปริภูมิสีอาร์จีบีได้ด้วยสูตร

$$\text{Gray} = \frac{R + G + B}{3}$$

เมื่อ  $\text{Gray}, R, G$  และ  $B$  คือเมทริกซ์ขนาด  $m \times n$  โดยให้สัญลักษณ์อักษร mat แสดงถึงเมทริกซ์ทั้ง 4 นี้ และ  $\langle \text{mat} \rangle_{ij}$  คือสมาชิกตำแหน่งใด ๆ ที่นิยามค่าจำนวนเต็มในช่วง  $\langle \text{mat} \rangle_{ij} \in [0, 255]$

ปริภูมิสีเอชเอสวี (HSV color space) เป็นปริภูมิที่มี 3 ช่องสี (Channels) เช่นเดียวกับ ปริภูมิสีอาร์จีบีที่ประกอบด้วยช่องสีแดง ช่องสีเขียว และช่องสีน้ำเงิน แต่ปริภูมิเอชเอสวีจะประกอบด้วย ช่องเนื้อสี (hue; H) ช่องความอิมตัวของสี (Saturation; S) ช่องค่าความสว่างของแสง (Value; V) ซึ่งจัดเป็นช่องการมองเห็นสีสดคล้องกับมนุษย์มากกว่าปริภูมิอาร์จีบี ด้วยเพรำมนุษย์ใช้ความสว่างของแสงในการจำแนกโทนสี ดังแสดงในภาพที่ 1



ภาพที่ 1 ปริภูมิสีเอชแอลวี

สมการแสดงความสัมพันธ์ระหว่างปริภูมิสี  
อาร์จีบีกับปริภูมิสีเอชแอลวีได้แก่ [8]

$$v = \max(r, g, b)$$

$$s = \begin{cases} \frac{v - \min(r, g, b)}{v} & \text{if } v \neq 0 \\ 0 & \text{otherwise} \end{cases}$$

$$h = \begin{cases} \frac{60(g-b)}{v - \min(r, g, b)} & \text{if } v=r \\ 120 + \frac{60(b-r)}{v - \min(r, g, b)} & \text{if } v=g \\ 240 + \frac{60(r-g)}{v - \min(r, g, b)} & \text{if } v=b \end{cases}$$

$$h \leftarrow h + 360 \quad \text{if } h < 0$$

เมื่อ  $0 \leq r, g, b \leq 1$  คือพิกัดสีในปริภูมิสี  
อาร์จีบี(ปรับขนาด) และ  $h, s$  และ  $v$  คือพิกัดสีใน  
ปริภูมิสีเอชแอลวี

#### 1.1.2 อีสโต้แกรมของภาพดิจิตอล

ภาพดิจิตอลโทนสีเทาที่ใช้เล็กฐาน 2 จำนวน 8 บิต  
จะแสดงค่าระดับความเข้มโทนสีเทาได้ 256 ระดับ มี  
ผลรวมค่าอีสโต้แกรมทุกระดับคือ

$$\sum_{k=0}^{255} h(f_k) = \sum_{k=0}^{255} \sum_{n=1}^{n_k} 1 = \sum_{k=0}^{255} n_k$$

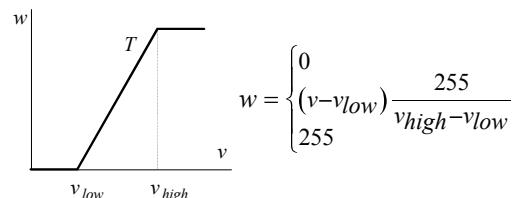
ในกรณีภาพมีขนาด  $m \times n$  ผลรวมค่าอีสโต้แกรมทุก  
ระดับจึงเท่ากับจำนวนจุดภาพทั้งหมดหรือ

$$\sum_{k=0}^{255} n_k = mn$$

โดยที่การกระจายตัวของอีสโต้แกรมในรูปแบบ  $xy$   
( $x=k, y=h$ ) จะส่งผลต่อภาพประกาย การแผ่ขยาย  
กราฟอีสโต้แกรมที่รวมกลุ่มอยู่ในช่วงระดับความเข้ม  
หนึ่งออกไป จะส่งผลให้การเปรียบต่าง (Contrast)  
ของภาพประกายดีขึ้น การแผ่ขยายกราฟอีสโต้แกรม  
สามารถทำได้โดยการแปลงค่าระดับความเข้ม

$$w = T(v)$$

เมื่อ  $w$  และ  $v$  คือระดับความเข้มโทนสีเทา  
ของภาพใหม่และเดิมตามลำดับ การแปลงค่าระดับ  
ความเข้มดังภาพที่ 2 คือการปรับเบรียบต่างแบบ  
อัตโนมัติ (Automatic contrast adjustment) [9]



ภาพที่ 2 การแปลงค่าระดับความเข้มแบบการปรับ  
เบรียบต่างแบบอัตโนมัติ

## 2. วิธีดำเนินการวิจัย

ในอุตสาหกรรมเกี่ยวข้องกับกาแฟ ยังขาด  
การกำหนดคำศัพท์เฉพาะเป็นข้อตกลงร่วมกันในการ  
กำหนดระดับของเมล็ดกาแฟคั่ว ทำให้เกิดความสับสน  
ในกลุ่มของนักค้ากาแฟและผู้บริโภค อย่างไรก็ตามมี  
การแบ่งระดับการคั่วออกเป็น 6 ระดับได้แก่ [10]  
Cinnamon, City, Full city, Vienna, French  
และ Italian การให้ความร้อนเมล็ดกาแฟโดยการคั่ว

จะทำให้สีของเมล็ดกาแฟเปลี่ยนแปลง นำไปสู่การจัดระดับของการคั่วเมล็ดกาแฟดังภาพที่ 3



ภาพที่ 3 ระดับสีการคั่วเมล็ดกาแฟ (ที่มา : Dan Bollinger - Own work, CC BY-SA 3.0, <https://commons.wikimedia.org>)

## 2.1 ตัวแปรหน่วยความจำภาพเมล็ดกาแฟคั่ว

ปริภูมิสีอาร์จีบีถูกใช้ในการแสดงผลภาพสีบนจอภาพคอมพิวเตอร์ ดังนั้นภาพเมล็ดกาแฟคั่วขนาด  $m \times n$  จะดูภาพจะต้องใช้เมทริกซ์  $mat$  ตามขนาดภาพ จำนวน 3 เมทริกซ์สำหรับช่องสี  $R$ ,  $G$  และ  $B$  ทำให้ภาพเมล็ดกาแฟคั่วเขียนได้ในลักษณะเมทริกซ์แต่งเติม (Augmented matrix) ดังนี้

$${}^{RGB} Mat = [R : G : B]$$

ค่าระดับความเข้มสีในช่องสีใด ๆ เช่นช่องสี  $R$  จึงจะเขียนแทนด้วย

$$\left( {}^{RGB} Mat.R \right)_{ij} = r$$

เมื่อ  $r$  คือระดับความเข้มสีในช่องสี  $R$  การสร้างตัวแปรหน่วยความจำสำหรับเมทริกซ์  $Mat$  โดยภาษาการโปรแกรมซีพลัสพลัส ประกอบร่วมกับไลบรารี

ของโอลูฟอนซีวี [11] ที่เป็นแบบทรัพยากรเปิดจะมีลักษณะข้อความสั้งต่อไปนี้

```
Mat CofRstImg =
    Imread(RstBean.jpg);
```

เมื่อ  $Mat$  คือวัตถุเชิงโปรแกรมของโอลูฟอนซีวี  $CofRstImg$  คือชื่อตัวแปรหน่วยความจำตั้งขึ้นสำหรับหน่วยวัตถุข้อมูลภาพในปริภูมิสีอาร์จีบี และ  $RstBean.jpg$  คือไฟล์ข้อมูลภาพเมล็ดกาแฟคั่วถ้าให้  $r = 127$  แล้ว  $\langle Mat.R \rangle_{ij}$  ดังสมการข้างต้นจะเขียนเป็นข้อความสั้งได้ดังนี้

```
CofRstImg.at<Vec3b>(i, j)[3]=127;
```

## 2.2 การบันทึกอัลเมล็ดกาแฟคั่วเดียวกัน ถ่ายในสภาพแสงที่แตกต่างกัน สามารถส่งผลต่อระบบคอมพิวเตอร์วิทัศน์ในการตัดสินใจระบุค่าระดับสีที่ไม่เหมือนกันซึ่งนับเป็นข้อด้อย ดังนั้นเพื่อลดความไวแสงของเหตุการณ์นี้จึงทำการปรับเท่าอีสโตแกรมให้กับช่องค่าความสว่างของแสง ของภาพเมล็ดกาแฟคั่วในปริภูมิเชอส์วี ในกรณีจะประยุกต์ใช้ฟังก์ชันการแจกแจงสะสม (Cumulative distribution function) 並將ค่าระดับความเข้ม [12]

$$w = T(v) = \int_0^v p_v(s) ds \quad 0 \leq v \leq 1$$

เมื่อ  $s$  คือตัวแปรหุ่น (Dummy variable) ของการหาปริพันธ์  $p_v(v)$  คือฟังก์ชันความหนาแน่นของความน่าจะเป็น (Probability density function) ทำการแปลงมีสมบัติ  $0 \leq T(v) \leq 1$  เป็นฟังก์ชันค่าเดียวและเพิ่มขึ้นทางเดียว ทำให้

ความหนาแน่นของความน่าจะเป็นของความเข้มใหม่  
เป็นแบบเอกสารปุ๊บ

$$p_w(w) = \left[ p_v(v) \frac{dv}{dw} \right]_{v=T^{-1}(w)} = 1$$

สำหรับการปรับเท่าฮีสโตแกรม ซึ่งในงานวิจัยนี้  
 $p_v(v)$  ไม่ต่อเนื่องและคำนวณค่าจาก

$$p_v(v_k) = \frac{n_k}{mn}$$

เมื่อ  $p_v(v_k)$  คือการแจกแจงความน่าจะเป็น<sup>(Probability distribution)</sup> ของตัวแปรสุ่ม  $v_k$  ทำให้  
ต้องกำหนดสมการแปลงค่าระดับความเข้มในรูป<sup>ปริพันธ์</sup>ข้างต้นใหม่เป็น

$$w_{v_k} = T(v_k) = \sum_{j=0}^{v_k} \frac{n_j}{mn} = \sum_{j=0}^{v_k} p_v(v_k)$$

ทำการแปลงปริภูมิสีอาร์จีบี เป็นปริภูมิสีเอชเอสวี  
ที่เขียนแทนด้วย

$$HSV\ Mat \leftarrow RGB\ Mat$$

เมื่อ  $HSV\ Mat$  คือเมทริกซ์แต่งเติมสำหรับช่องสี  
 $H, S$  และ  $V$

$$HSV\ Mat = [H : S : V]$$

ทำการเขียนฮีสโตแกรมของช่องค่าความสว่างของ  
แสงในรูปเมทริกซ์ແວตั้งนี้

$$H(\text{HSV}\ Mat.V) = [n_0 \ n_1 \ n_2 \dots n_k \dots n_{255}]$$

เมื่อ  $n_k$  คือค่าฮีสโตแกรม  $h(f_k)$  ทำให้  
สามารถคำนวณเมทริกซ์การแจกแจงความน่าจะเป็น

$$P = \frac{H(\text{HSV}\ Mat.V)}{mn} = [p_0 \ p_1 \ p_2 \ p_3 \dots p_k \dots p_{255}]$$

จึงคำนวณค่าจำนวนเต็มระดับความเข้มใหม่จาก

$$\langle V' \rangle_{ij} = w_{\langle v \rangle_{ij}} = 255 \times \sum_{k=0}^{\langle v \rangle_{ij}} \langle P \rangle_k$$

โดยที่จำนวนเต็ม  $\langle V' \rangle_{ij} \in [0, 255]$  สำหรับ  
การปรับปรุงภาพดิจิตอลใหม่

$$HSV\ Mat' = [H : S : V'] \leftarrow HSV\ Mat = [H : S : V]$$

ซึ่งสามารถดำเนินงานด้วยไลบรารีของโอเพนซีวีด้วย<sup>ข้อความสั้น</sup>

`equalizeHist(src, dst);`

เมื่อ  $src$  คือ  $HSV\ Mat.V$  และ  $dst$  คือ <sup>$HSV\ Mat'.V'$</sup>

### 2.3 คำนวณสีเมล็ดกาแฟคั่วตัวยค่าทาง

สถิติ

ด้วยมุมมองการกระจายตัวค่าระดับ  
ความเข้มสีในภาพดิจิตอลคือตัวแปรสุ่ม ที่มีการแจก  
แจงความน่าจะเป็นแบบไม่ต่อเนื่อง คำนวณค่าได้ดังนี้

$$\begin{aligned} {}^{RGB} Mat' &= [R' : G' : B'] \leftarrow {}^{HSV} Mat' = [H : S : V] \\ P_r &= \frac{H({}^{RGB} Mat'. R')}{mn} \\ P_g &= \frac{H({}^{RGB} Mat'. G')}{mn} \\ P_b &= \frac{H({}^{RGB} Mat'. B')}{mn} \end{aligned}$$

เมื่อ  $P_r$ ,  $P_g$  และ  $P_b$  คือการแจกแจงความน่าจะเป็นไม่ต่อเนื่องของระดับความเข้มสีแดง สีเขียว และสีน้ำเงินตามลำดับ นำไปสู่ค่าคาดหมายของแต่ละสีปฐมภูมิ โดยการกำหนดให้  $L$  คือเมตริกซ์แล้วของระดับความเข้มสี

$$L = [0 \ 1 \ 2 \ \dots \ k \ \dots \ 255]$$

ค่าคาดหมายจึงคำนวณด้วยสูตร

$$E_r = P_r L^T, E_g = P_g L^T \text{ และ } E_b = P_b L^T$$

ภาพสีตัวแทนของภาพเมล็ดกาแฟคั่ว  ${}^{RGB} Mat^*$  จะคำนวณได้โดย

$$\begin{aligned} \left\langle {}^{RGB} Mat^*. R \right\rangle_{ij} &= E_r \quad \forall i, j \\ \left\langle {}^{RGB} Mat^*. G \right\rangle_{ij} &= E_g \quad \forall i, j \\ \left\langle {}^{RGB} Mat^*. B \right\rangle_{ij} &= E_b \quad \forall i, j \end{aligned}$$

$${}^{RGB} Mat^* = [R : G : B]$$

ซึ่งมีค่าส่วนเบี่ยงเบนมาตรฐานแต่ละช่องสีดังนี้

$$S_r = \sqrt{P_r D_r (L - E_r I_R)^T}$$

$$\begin{aligned} S_g &= \sqrt{P_g D_g (L - E_g I_R)^T} \\ S_b &= \sqrt{P_b D_b (L - E_b I_R)^T} \end{aligned}$$

เมื่อเมตริกซ์ที่ແຍ່ງນຸ່ມ  $D_k$  คือ

$$D_k \Big|_{k=r,g,b} = \begin{bmatrix} 0-E_k & 0 & 0 & 0 & \dots & 0 \\ 0 & 1-E_k & 0 & 0 & \dots & 0 \\ 0 & 0 & 2-E_k & 0 & \dots & 0 \\ 0 & 0 & 0 & 3-E_k & \dots & 0 \\ \vdots & & & & & \\ 0 & 0 & 0 & 0 & \dots & 255-E_k \end{bmatrix}$$

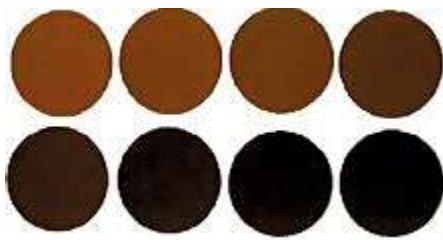
และ  $I_R$  คือเมตริกซ์ແລວ  $I_R = [1 \ 1 \ 1 \ \dots \ 1 \ \dots \ 1]_{1 \times 256}$   
ທັງค่าคาดหมายและค่าส่วนເບີຍເບີນມາຕຽບ  
ສາມາດຄຳນວນໄດ້ດ້ວຍຂໍ້ອົກສ່ວນສິ່ງທີ່ໃຊ້ພຶກໜັນຂອງໄລບຣາຣີຂອງໂອເປັນຊີຣີດັ່ງນີ້

`meanStdDev(srcRGB, Ergb, Srgb);`

เมื่อ `srcRGB` คือหน่วยວັດຖຸເຂົ້າໂປຣແກຣມ  
ຂອນລາຍສໍາຫຼັບ  ${}^{RGB} Mat' = [R' : G' : B']$  โดยທີ່  
`Ergb` และ `Srgb` คือຕັ້ງແປຮນ່ວຍຄວາມຈຳແກວ  
ລຳດັບຂອງค่าคาดหมาย  $E_r$ ,  $E_g$ ,  $E_b$  และສ່ວນ  
ເບີຍເບີນມາຕຽບ  $S_r$ ,  $S_g$ ,  $S_b$

#### 2.4 ขั้นตอนວິธີคำนวณตัวแทนสีภาพเมล็ดกาแฟคั่ว

การວัดຮະດັບມີເລື່ອດັບມີກາແພື່ນວ່າອູ້ຮະດັບໄດ້  
ໃນທາງປົງປັບຕິໄດ້ມີກາໃຊ້ແຜ່ນວັດຖຸເທິຍກາຮັກ  
(Roast color tiles) ມີຮະດັບສີແສດງດັ່ງການທີ່ 4 [13]



ภาพที่ 4 แผ่นรัสดุสีเทียบการค่าวเมล็ดกาแฟ

จากภาพจะสังเกตได้ว่ามีทั้งหมด 8 ระดับ โดยระดับที่เข้มที่สุดเป็นระดับ Italian/Spanish หมายความว่า แผ่นรัสดุเทียบสีการคั่ว มีสีระดับการคั่วขึ้นมากกว่า ก่อนหน้าอีก 2 ระดับ

---

*input(<sup>RGB</sup> Mat = [R : G : B], m, n, L, D<sub>r</sub>, D<sub>g</sub>, D<sub>b</sub>, I<sub>R</sub>)*  
<sup>HSV</sup> Mat = [H : S : V]  $\leftarrow$  <sup>RGB</sup> Mat = [R : G : B]  
*equalizeHist(<sup>HSV</sup> Mat)*  
<sup>RGB</sup> Mat' = [R' : G' : B']  $\leftarrow$  <sup>HSV</sup> Mat' = [H : S : V']  
*meanStdDev(<sup>RGB</sup> Mat')*

---

ภาพที่ 5 ขั้นตอนวิธีคำนวณตัวแทนสีภาพเมล็ดกาแฟคั่ว

ขั้นตอนวิธีในภาพที่ 5 ถูกสร้างขึ้นเพื่อใช้ คอมพิวเตอร์ทำการคำนวณค่าตัวแทนสี ของภาพ เมล็ดกาแฟคั่ว  $E_r$ ,  $E_g$  และ  $E_b$  พร้อมทั้งค่า ส่วนเบี่ยงมาตรฐาน ซึ่งค่าตัวแทนสีนี้นำไปสร้างภาพ ดิจิตอล  $^{RGB} Mat^* = [R : G : B]$  ของภาพเมล็ดกาแฟคั่วได้โดยมีขั้นตอนวิธีสำหรับฟังก์ชัน *equalizeHist* และ *meanStdDev* แสดงต่อไป

---

*input(<sup>HSV</sup> Mat)*  
<sup>HSV</sup> Mat' = [H : S : V]  $\leftarrow$  <sup>HSV</sup> Mat = [H : S : V]  
 $P = \frac{H(^{HSV} Mat.V)}{mn}$   
*for i = 0 to m step 1*  
*for j = 0 to n step 1*  

$$\langle^{HSV} Mat.V \rangle_{ij} = 255 \times \sum_{k=0}^{\langle^{HSV} Mat.V \rangle_{ij}} \langle P \rangle_k$$
  
*end for*  
*end for*

---

ภาพที่ 6 ขั้นตอนวิธีคำนวณ *equalizeHist*

---

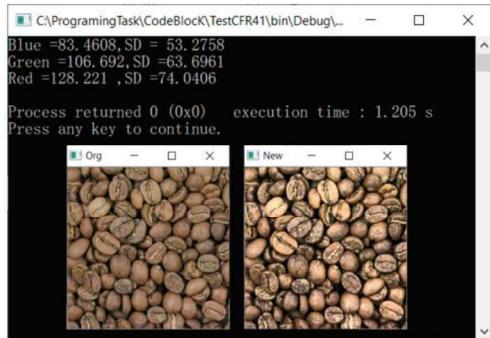
*input(<sup>RGB</sup> Mat')*  
*for i = r, g, b according to I = R, G, B*  
 $P_i = \frac{H(^{RGB} Mat'.I)}{mn}$   
 $E_i = P_i^T$   
 $S_i = \sqrt{P_i D_i (L - E_i I_R)^T}$   
*end for*

---

ภาพที่ 7 ขั้นตอนวิธีคำนวณ *meanStdDev*

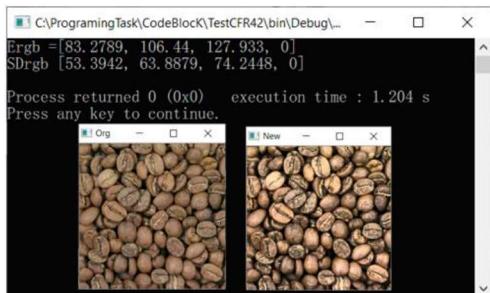
### 3. ผลการวิจัยและอภิปรายผล

เมื่อนำขั้นตอนวิธีที่สร้างขึ้นในภาพที่ 5 ไป พัฒนาโปรแกรม ด้วยภาษาการโปรแกรมซีพลัสพลัส จะมีลักษณะดังภาพที่ ก.1 ในภาคผนวก ก. และใช้ ภาพดิจิตอลของเมล็ดกาแฟคั่วในภาพที่ 3 ที่ระดับคั่ว Cinnamon บันทึกอยู่ในหน่วยความจำสำรอง ของ ระบบคอมพิวเตอร์ใช้ชื่อไฟล์ “Cinnamon.png” สำหรับเป็นข้อมูลเข้าให้กับขั้นตอนวิธีคำนวณตัวแทน สีภาพเมล็ดกาแฟ ผลการทำงานของโปรแกรมแสดง ดังภาพที่ 8 ได้ค่าตัวแทนสีแดง 128.2210 ค่าตัวแทน สีเขียว 106.6920 และ ค่าตัวแทนสีน้ำเงิน 83.4608



ภาพที่ 8 ผลการทำงานของโปรแกรมคำนวณค่าตัวแทนสีภาพเมล็ดกาแฟคั่วจากขั้นตอนวิธีในภาพที่ 5 - 7

ภาพเมล็ดกาแฟคั่วระดับ Cinnamon ทางซ้ายของภาพที่ 8 เป็นภาพตั้งเดิมที่ยังไม่ปรับเท่าอีสโตแกรมส่วนทางขวาปรับเท่าอีสโตแกรมแล้วจึงนำเข้าสู่การคำนวณค่าตัวแทนสี อย่างไรก็ตามผลการทำงานแบบเดียวกัน แต่ใช้ฟังก์ชันในไลบรารีของโอลเอนชีรี ปรับเท่าอีสโตแกรม ตลอดจนคำนวณค่าคาดหมาย และส่วนเบี่ยงมาตรฐาน แทนที่จะใช้ขั้นตอนวิธีที่สร้างขึ้นในภาพที่ 6 - 7 ของบทความ ส่งผลให้การพัฒนาโปรแกรมทำได้จ่ายขั้นตั้งโปรแกรม ในภาพที่ ก.3 (ภาพผนวก ก.) นำไปสู่ผลการทำงานที่แตกต่างออกไปเล็กน้อยดังภาพที่ 9



ภาพที่ 9 ผลการทำงานของโปรแกรมคำนวณค่าตัวแทนสีภาพเมล็ดกาแฟคั่วโดยใช้ไลบรารีของโอลเอนชีรี

การใช้ฟังก์ชันในไลบรารีของโอลเอนชีรี ทำให้ผลลัพธ์แทนสีของเมล็ดกาแฟคั่วระดับ Cinnamon ในภาพที่ 9 แตกต่างจากภาพที่ 8 ไปเล็กน้อย โดยค่าตัวแทนสีแดง 127.9330 ค่าตัวแทนสีเขียว 106.4400 และค่าตัวแทนสีน้ำเงิน 83.2789 ทั้งนี้ในการที่จะทำให้ผลการทำงานของโปรแกรม เทียบเคียงได้กับแผ่นรั่ดสีเทียบการคั่วเมล็ดกาแฟ ดังภาพที่ 4 ค่าตัวแทนสีจึงถูกแปลงเป็นภาพตัวแทนสีด้วยขั้นตอนวิธีในภาพที่ 10 โดยจะมีผลการทำงานดังภาพที่ 11

---

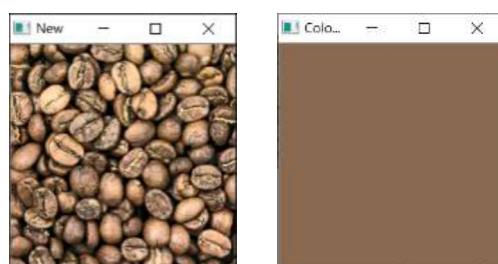
```

RGB Mat = [R : G : B] m, n, Er, Eg, Eb)
for i = 0 to m step 1
    for j = 0 to n step 1
        for k = r, g, b according to
            K = R, G, B
             $\left\langle \begin{smallmatrix} \text{RGB} \\ \text{Mat.K} \end{smallmatrix} \right\rangle_{ij} = E_k$ 
        end for
    end for
end for

```

---

ภาพที่ 10 ขั้นตอนวิธีสร้างภาพสีตัวแทนเมล็ดกาแฟคั่ว



ภาพที่ 11 ผลการสร้างภาพสีตัวแทนเมล็ดกาแฟคั่ว

ผลการทำงานของโปรแกรมยังสามารถเปรียบเทียบกับการเฉลี่ยสีของภาพดิจิตอลแบบออนไลน์บนเว็บไซต์ <http://matkl.github.io/average-color/> ซึ่งมีผลการทำงานดังภาพที่ 12



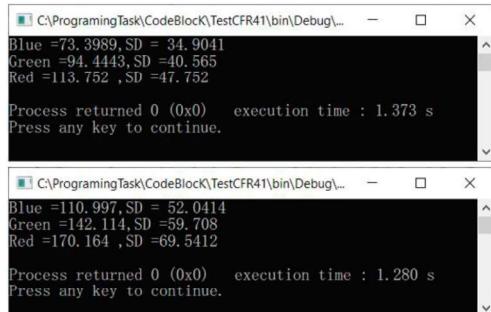
ภาพที่ 12 ผลการเฉลี่ยสีของภาพเมล็ดกาแฟคั่วโดยใช้เว็บไซต์แบบออนไลน์

ผลการเฉลี่ยสีของภาพเมล็ดกาแฟคั่วดังภาพที่ 12 ให้ผลลัพธ์ค่าสีแดง 127 ค่าสีเขียว 106 และค่าสีน้ำเงิน 83 ซึ่งสอดคล้องและใกล้เคียงกันมากกับโปรแกรมที่พัฒนาขึ้น ผลการปรับเท่าฮีสโตแกรมจะส่งผลให้ค่าตัวแทนสีของภาพเดียวกันแต่ให้แสงสว่างไม่เท่ากัน 2 ภาพ มีสภาพคงทันต่อความสว่างของภาพโดยจะสังเกตได้จากภาพเมล็ดกาแฟคั่วเดียวกัน 2 ภาพ แต่อีกภาพปรับสภาพความสว่างเพิ่มขึ้น 2 เท่า



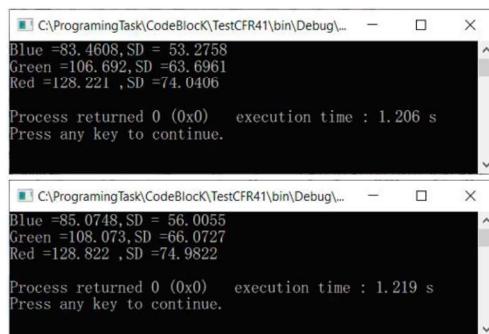
ภาพที่ 13 ภาพเมล็ดกาแฟคั่วเดียวกันปรับความสว่าง ต่างกัน 1 เท่า

จากเดิม ผลการใช้โปรแกรมคำนวณค่าสีตัวแทนเมล็ดกาแฟคั่วทั้ง 2 ภาพดังปรากฏในภาพที่ 13 คือ

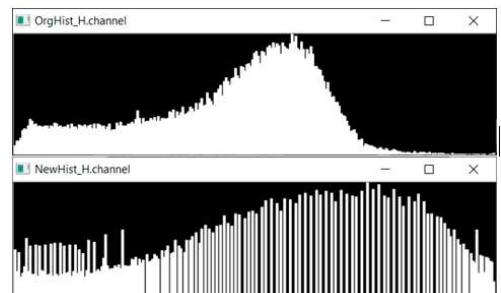


ภาพที่ 14 ผลคำนวณสีตัวแทนภาพเมล็ดกาแฟคั่วความสว่างต่างกัน 1 เท่าที่มีการปรับเท่าฮีสโตแกรม

เห็นได้จากการคำนวณในภาพข้างต้นว่า มีความแตกต่างค่าสีตัวแทนแต่ละค่าไม่เกิน 2 หน่วย แต่ในขณะที่ไม่มีการใช้การปรับเท่าฮีสโตแกรม ในขั้นตอนวิธีคำนวณค่าตัวแทนสีภาพเมล็ดกาแฟคั่ว จะได้ผลดังภาพที่ 15 มีความแตกต่างของค่าสีตัวแทนแต่ละสีเกินกว่า 37 หน่วย นอกจากนี้ ในการปรับเท่าฮีสโตแกรมซึ่งส่งผลให้ การกระจายตัวของฮีสโตแกรมของภาพขยายกว้างขึ้น จะสามารถสังเกตได้จากการนำค่าฮีสโตแกรมในช่องค่าความสว่างของแสง V ก่อนและหลังการปรับเท่าฮีสโตแกรมไปคาดกราฟ มีผลดังภาพที่ 16 โดยภาพด้านบนคือภาพฮีสโตแกรมของช่องค่าความสว่างของแสง V ของเมล็ดกาแฟคั่วเดิม ส่วนภาพล่างคือฮีสโตแกรมที่ได้หลังการปรับเท่าฮีสโตแกรมแล้ว

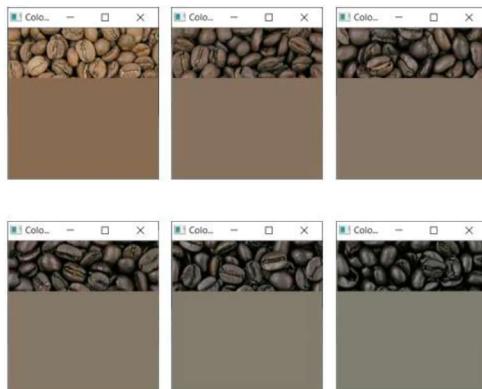


ภาพที่ 15 ผลคำนวณสีตัวแทนภาพเมล็ดกาแฟคั่วความสว่างต่างกัน 1 เท่าไม่มีการปรับเท่าฮีสโตแกรม



ภาพที่ 16 ภาพฮีสโตแกรมก่อนและหลังการปรับเท่าฮีสโตแกรม

เห็นได้ชัดว่าการปรับเท่าฮีสโตแกรมที่ทำให้ค่าฮีสโตแกรมมีกระจายตัวมากขึ้น ส่งผลให้ส่วนเบี่ยง บนมาตรฐานมีค่าสูงตั้งผลการทำงานในภาพที่ 14 เมื่อเปรียบเทียบกับผลการทำงานในภาพที่ 15 ที่มีค่าส่วนเบี่ยงบนมาตรฐานต่ำกว่าอย่างเด่นชัด การสร้างภาพสีตัวแทนสี ด้วยระเบียบวิธีทางคอมพิวเตอร์ของบทความนี้สำหรับภาพเมล็ดกาแฟคั่วตามภาพที่ 3 ทั้ง 6 ระดับได้ผลแสดงดังภาพที่ 17 และ 18



ภาพที่ 17 ผลภาพสีตัวแทนเมล็ดกาแฟคั่วโดยปรับเท่าฮีสโตแกรมช่องค่าความสว่างก่อนคำนวน

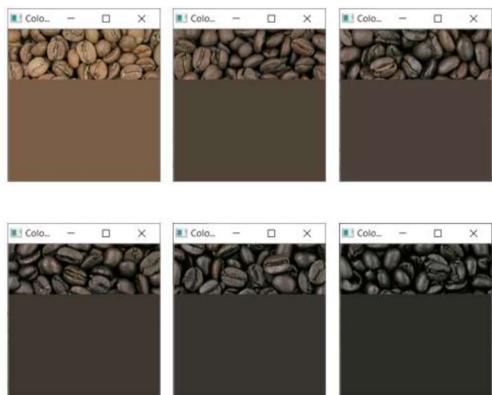
ตารางที่ 1 ผลคำนวนค่าตัวแทนสีภาพเมล็ดกาแฟคั่วปรับเท่าฮีสโตแกรมช่องค่าความสว่างก่อนคำนวน

ระดับ คั่ว	ขั้นตอนวิธี			ค่าเฉลี่ยสี			ความแตกต่าง ของภาพ		
	บทความวิจัย			(ออนไลน์)					
B	G	R	B	G	R	B	G	R	
cnm	83	107	128	83	106	127	-	1	1
cit	96	112	128	97	113	128	1	-	-
Fcit	103	116	128	103	115	128	-	1	-
ven	104	118	128	104	118	127	-	-	1
fre	110	122	127	109	122	127	1	-	-
ita	111	126	126	110	125	126	1	-	-

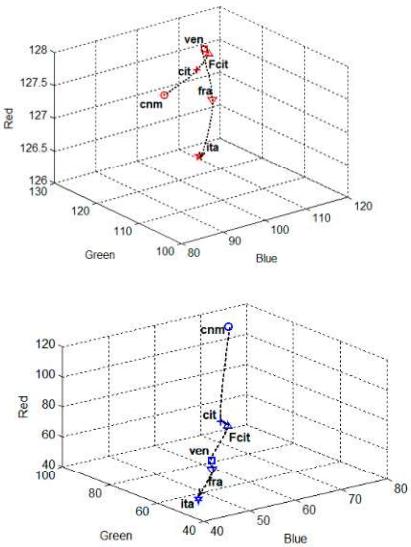
ตารางที่ 2 ผลคำนวนค่าตัวแทนสีภาพเมล็ดกาแฟคั่วไม่ปรับเท่าฮีสโตแกรมช่องค่าความสว่างก่อนคำนวน

ระดับ คั่ว	ขั้นตอนวิธี			ค่าเฉลี่ยสี			ความแตกต่าง		
	บทความวิจัย			ของภาพ			(ออนไลน์)		
B	G	R	B	G	R	B	G	R	
cnm	73	94	114	73	94	113	-	-	1
cit	58	68	77	57	67	77	1	1	-
Fcit	59	67	74	59	66	73	-	1	1
ven	51	58	62	51	57	62	-	1	-
fre	50	56	58	50	55	57	-	1	1
ita	43	48	48	42	48	48	1	-	-

ภาพที่เป็นข้อมูลเข้าให้กับขั้นตอนวิธีทางคอมพิวเตอร์ที่สร้างขึ้น ในแต่ละระดับการคั่วถูกปรับขนาดภาพลดลงจาก 800×800 เหลือ 200×200 จุดภาพ และมีการเปรียบเทียบผลการคำนวนกับการคำนวนค่าเฉลี่ยสีของภาพที่ได้จากการเว็บไซต์ออนไลน์ ดังแสดงในภาพที่ 12 พอบว่ามีระดับค่าตัวแทนใกล้เคียงกันมาก ดังแสดงในตารางที่ 1 - 2 ซึ่งนับเป็นผลการพัฒนาโปรแกรมบนไอเดีย (Integrated development environment; IDE) ที่เป็นแบบทรัพยากรีดูซึ่งกันในนาม “CODE::BLOCK” รุ่น 17.12 โดยใช้ตัวแปลงโปรแกรม (Compiler) “GNU GCC” บนระบบปฏิบัติการ “Windows 10” ตัวประมวลผล “Intel(R) Core (TM) i7- 8750H CPU@2.2 GHz” นอกจากนี้ยังได้ทดลองปฏิบัติโปรแกรมบนไอเดีย และตัวแปลงโปรแกรมเดียวกันบนคอมพิวเตอร์บอร์ดเดียวraspberrypi 3 รุ่นบี (Raspberry pi 3 model B) บนระบบปฏิบัติการ Linux raspberrypi 4.14.93-v7, armv71 GNU/Linux ซึ่งได้ผลการทำงานเหมือนกันทุกประการ



ภาพที่ 18 ผลภาพสีตัวแทนภาพเมล็ดกาแฟคั่วไม่ปรับเท่าฮีสโตแกรมซึ่งค่าความสว่างก่อนคำนวณ



ภาพที่ 19 กราฟตัวแทนค่าตัวแทนสีภาพเมล็ดกาแฟคั่วจากการคำนวณในตารางที่ 1 และตารางที่ 2

การเคลื่อนที่ของพิกัดสีในปริภูมิอาร์จีบี ของกราฟด้านบนในภาพที่ 19 จากค่าอ่อนที่สุดไปถึงค่าเข้มที่สุด 6 ระดับ เป็นผลจากการคำนวณในตารางที่ 1 โดย 4 ระดับแรกมีลักษณะเคลื่อนที่ในทิศทางการเพิ่มค่าของพิกัดสีเขียวและสีน้ำเงิน บนรูปแบบแกนพิกัดสีแดงคงที่ ขณะที่ 2 ระดับสุดท้ายค่าพิกัดสีแดงจะมี

ทิศทางลดลงเล็กน้อย แต่สำหรับกราฟด้านล่างเป็นผลจากการคำนวณในตารางที่ 2 การเคลื่อนที่ของพิกัดสีในปริภูมิอาร์จีบีจากค่าอ่อนที่สุดไปถึงค่าเข้มที่สุดมีทิศทางลดลงทั้ง 3 สี

#### 4. สรุปผลการวิจัย

ระเบียบวิธีทางคอมพิวเตอร์ที่สร้างขึ้น ทำให้ภาพดิจิตอลของการคั่วเมล็ดกาแฟ ถูกแสดงแทนด้วยข้อมูลเชิงตัวเลข 3 ค่าคือ ตัวแทนสีน้ำเงิน สีเขียวและสีแดงพร้อมทั้งสร้างภาพตัวแทนสี ที่สามารถนำไปพัฒนาระบบคอมพิวเตอร์วิทยศึกษา ในอุตสาหกรรมยุคที่ 4.0 ด้วยภาษาการโปรแกรมชีพลัสพลัสบนแหล่งทรัพยากรเปิดได้ทั้งหมด สำหรับระบบอัตโนมัติ การคั่วเมล็ดกาแฟเชิงอุตสาหกรรมหรือในครัวเรือน จากผลการทดลองพบว่าการปรับเท่าฮีสโตแกรมในช่องค่าความสว่างของปริภูมิอีซูวี ของภาพดิจิตอลเมล็ดกาแฟคั่วที่ระดับใด ๆ ทำให้ค่าตัวแทนสีมีความคงทนต่อสภาพแสงที่เปลี่ยนไป และข้อมูลตัวแทนสีมีแบบรูปการเคลื่อนที่ของค่าพิกัดสีที่แน่นอน ตามลำดับของภาพดิจิตอลเมล็ดกาแฟคั่ว จากระดับอ่อนไปจนถึงเข้มที่สุด ผลการทำงานของระเบียบวิธีทางคอมพิวเตอร์ ให้ผลการคำนวณต่างกันเพียงเล็กน้อย ในระดับเลข ทศนิยม เมื่อเทียบกับการใช้ฟังก์ชันสำเร็จรูปของໄล บรารีโอเพนซีวีบังฟังก์ชัน ที่ทำการคำนวณอย่างเดียวกัน จึงนับเป็นหนทางให้สามารถนำขั้นตอนวิธีของบทความนี้ไปพัฒนาโปรแกรมบนแพลตฟอร์มอื่นที่ไม่สามารถใช้ไลบรารีโอเพนซีวีได้ ดังเช่นในระดับมีโครคอนโทรลเลอร์ และเมื่อนำผลการคำนวณตัวแทนสี ไปเปรียบเทียบกับผลการคำนวณค่าเฉลี่ยสีของภาพดิจิตอลแบบออนไลน์ พบร่วมมีผลใกล้เคียงกัน

## 5. อ้างอิง

- [1] Andhare P, Rawat SP. Pick and Place Industrial Robot Controller with Computer Vision. 2016 International Conference on Computing Communication Control and automation (ICCUBEA); 2016 Aug 12-13; Pune, India: IEEE; 2016:1-4.
- [2] Mery D, Pedreschi F, Soto A. Automated Design of a Computer Vision System for Visual Food Quality Evaluation. Food and Bioprocess Technology, 2013; 6(8): 2093-2108.
- [3] Priyadarshini K, Akila R. A Survey on Computer Vision Technology for Food Quality Evaluation. IJIRCCE 2016; 4(8): 14860–65.
- [4] Habib MT, Majumder A, Jakaria AZM, et al. Machine Vision Based Papaya Disease Recognition. Journal of King Saud University-Computer and Information Sciences, 2018.
- [5] Trientin D, Hidayat B, Darana S. Beef Freshness Classification by Using Color Analysis, Multi-wavelet Transformation and Artificial Neural Network. 2015 International Conference on Automation, Cognitive Science, Optics, Micro Electro-Mechanical System, and Information Technology (ICACOMIT); 2015 Oct 29-30; Bandung, Indonesia: IEEE; 2015:181-85.
- [6] Adi K, Pujiyanto S, Nurhayati OD, et al. Beef Quality Identification Using Color Analysis and K-nearest Neighbor Classification. 2015 4<sup>th</sup> International Conference on Instrumentation, Communications, Information Technology, and Biomedical Engineering (ICICI-BME); 2015 Nov 2-3; Bandung, Indonesia: IEEE; 2015:180-4.
- [7] Laurent B, Ousman B, Dzudie T, et al. Digital Camera Images Processing of Hard-to-Cook Beans. J Eng Tech Res, 2010; 2(9): 177 – 88.
- [8] Garcia G, Suarez OD, Aranda JLE, et al. Learning Image Processing with Open CV. Birmingham: Packt Publishing Ltd; 2015
- [9] Burger W, Burge MJ. Principles of Digital Image Processing: Fundamental Techniques. London: Springer-Verlag London Limited; 2009.
- [10] Rao S. The Coffee Roaster's Companion. Canada: Scott Rao; 2014.
- [11] Kaehler A, Bradski G. Learning OpenCV 3: Computer Vision in C++ with The OpenCV library. Tokyo: O'Reilly Media, Inc.; 2017.
- [12] Gonzalez RC, Woods RE. Digital Image Processing. New Jersey: Prentice Hall; 2002.
- [13] Sinnott K. The Art and Craft of Coffee: An Enthusiast's Guide to Selecting, Roasting, and Brewing Exquisite Coffee. Massachusetts: Quarry Books; 2011.

## 7. ภาคผนวก

### ภาคผนวก ก.

```

1 #include "opencv2/highgui/highgui.hpp"
2 #include "opencv2/imgproc/imgproc.hpp"
3 #include <iostream>
4 using namespace cv; using namespace std;
5 void histogramPlot(Mat, String);
6 void colorPlot(Mat, Mat* );
7 int main( int argc, char** argv )
8 { Mat MatRGB imread("cinnamon.png");
9  namedWindow("Org", 1);
10 imshow("Org", MatRGB);
11 Mat MatHSV;
12 cvtColor(MatRGB, MatHSV, CV_BGR2HSV);
13 vector<Mat>colorSpaceHSV;
14 split(MatHSV, colorSpaceHSV);
15 Mat Mat_Hue= colorSpaceHSV[2].clone();
16 int imgRow=MatRGB.rows, imgCol=MatRGB.cols;
17 Mat histData; int nbins=256;
18 int hsize[]={nbins};
19 float range[]={0, 256};
20 const float *ranges[]={range};
21 int chnls[]={0};
22 calcHist(&Mat_Hue, 1, chnls, Mat(),
23          histData,1,hsize,ranges);
24 histogramPlot(histData, "OrgHist_H.channel");
25 Mat Pr=histData/sum(histData)[0];
26 Mat Mat_nHue = Mat_Hue.clone();
27 float CPD[256]; Mat L=cv::Mat(1,256,CV_32F);
28 for(int i=0; i<256; i++)
29 {
    CPD[i]=0.0; L.at<float>(i)=(float)i;
30     for( int j=0; j<=i; j++)
31         CPD[i] += Pr.at<float>(j); } uchar k;
32 for( int i=0; i<imgRow; i++)
33     { for( int j=0; j<imgCol; j++)
34         { k=Mat_Hue.at<uchar>(i,j);
35             Mat_nHue.at<uchar>(i,j)=
36             (uchar)round((255*CPD[k]));
37             calcHist(&Mat_nHue, 1, chnls, Mat(),
38                      histData,1,hsize,ranges);
39             histogramPlot(histData, "NewHist_H.channel");
40             Mat nMatHSV, nMatRGB; colorSpaceHSV[2]=Mat_nHue;
41             merge(colorSpaceHSV, nMatHSV);
42             cvtColor(nMatHSV, nMatRGB, CV_HSV2BGR);
43             vector<Mat> colorSpaceRGB;
44             split(nMatRGB, colorSpaceRGB);
45             Mat histRGB[3], Prb[3], Expt[3];
46             Mat R1(1,256,CV_32FC1,Scalar(1.0)), D[3], Var[3];
47             Mat IL=Mat::eye(256,256, CV_32F);
48             for(int i=0; i<256; i++)
49                 IL.at<float>(i,i)=1;
50             calcHist(&colorSpaceRGB[0] , 1, chnls,
51                      Mat(), histRGB[0],1,hsize,ranges);
52             calcHist(&colorSpaceRGB[1] , 1, chnls,
53                      Mat(), histRGB[1],1,hsize,ranges);
54             calcHist(&colorSpaceRGB[2] , 1, chnls,
55                      Mat(), histRGB[2],1,hsize,ranges);
56             namedWindow("New", 1 );
57             imshow("New", nMatRGB );
58             for(int i=0; i<3; i++)
59                 { Prb[i]=histRGB[i]/sum(histRGB[i])[0];
60                   Expt[i]=Prb[i].t() * L.t();
61                   D[i]=IL-Mat::eye(256,256, CV_32F)*
62                     Expt[i].at<float>(0);
63                   Var[i]= Prb[i].t() * D[i] * (L.t()-
64                     Expt[i].at<float>(0) * R1.t()); }
65             cout<<"Blue ="<< round(Expt[0].at<float>(0))
66             << ", SD ="<<sqrt(Var[0].at<float>(0))<< "\n";
67             cout<<"Green ="<<round(Expt[1].at<float>(0))
68             << ", SD ="<<sqrt(Var[1].at<float>(0))<< "\n";
69             cout<<"Red ="<<round(Expt[2].at<float>(0))<< "\n";
70             << ", SD ="<<sqrt(Var[2].at<float>(0))<< "\n";
71             colorPlot(MatRGB, Expt);
72             waitKey();
73             return 0;
74 }
```

ภาพที่ ก1. โปรแกรมคำนวณตัวแทนสีภาพเมล็ด  
การแพ็คก์จากขั้นตอนวิธีในภาพที่ 5 – 7

```

76 void colorPlot(Mat MatRGB, Mat *Expt)
77 { Mat ColrRep = MatRGB.clone();
78     for ( int i=ColrRep.rows/3; i<ColrRep.rows; i++)
79     { for ( int j=0; j<ColrRep.cols; j++)
80         { ColrRep.at<Vec3b>(i,j)[0]=
81             round(Expt[0].at<float>(0));
82             ColrRep.at<Vec3b>(i,j)[1]=
83             round(Expt[1].at<float>(0));
84             ColrRep.at<Vec3b>(i,j)[2]=
85             round(Expt[2].at<float>(0));
86         }
87     }
88     namedWindow( "ColorRep", 1 );
89     imshow( "ColorRep", ColrRep );
90 }
91
92 void histogramPlot(Mat histData, String Name)
93 { Mat histBuff, histImg;
94     normalize(histData, histBuff,
95               0, 1, NORM_MINMAX, -1, Mat() );
96     histImg=imHist(histBuff,2,2);
97     namedWindow(Name, 1 );
98     imshow(Name,histImg);
99 }
```

ภาพที่ ก2. พังก์ชันสร้างภาพสีตัวแทนเมล็ดการแพ็คก์  
ตามขั้นตอนวิธีในภาพที่ 10 และวัดกราฟวิสโซตограм  
ในภาพที่ 16

```

1 #include "opencv2/highgui/highgui.hpp"
2 #include "opencv2/imgproc/imgproc.hpp"
3 #include <iostream>
4 using namespace cv; using namespace std;
5 void histogramPlot(Mat, String);
6 void colorPlot(Mat,Scalar);
7 int main( int argc, char** argv )
8 { Mat MatRGB imread("cinnamon.png");
9  namedWindow("Org", 1 );imshow("Org",MatRGB );
10 Mat MatHSV; cvtColor(MatRGB, MatHSV,CV_BGR2HSV);
11 vector<Mat>colorSpaceHSV;split(MatHSV,colorSpaceHSV);
12 Mat Mat_Hue=colorSpaceHSV[2].clone();
13 Mat histData;int nbins = 256;
14 int hsize[]={nbins};
15 float range[]={0,256};const float *ranges[]={range};
16 int chnls[]={0};
17 calcHist(&Mat_Hue,1,chnls,Mat(),histData,1,hsize,ranges);
18 histogramPlot(histData, "OrgHist_H.channel");
19 Mat src = Mat_Hue.clone(), dst;
20 equalizeHist( src, dst );
21 calcHist(&dst,1,chnls,Mat(),histData,1,hsize,ranges);
22 histogramPlot(histData, "NewHist_H.channel");
23 Mat nMatHSV, nMatRGB; colorSpaceHSV[2] = dst;
24 merge(&colorSpaceHSV, nMatHSV);
25 cvtColor(nMatHSV, nMatRGB, CV_HSV2BGR);
26 Scalar Ergb, SDrgb;meanStdDev(nMatRGB, Ergb, SDrgb);
27 cout<<"Ergebnis ="<<Ergebnis<<"\n"<<"SDrgb "<<SDrgb<<"\n";
28 colorPlot(MatRGB, Ergb);
29 waitKey();
30 return 0;
31 }
```

ภาพที่ ก3. โปรแกรมคำนวณตัวแทนสีภาพเมล็ด  
การแพ็คก์ โดยใช้พังก์ชันในไลบรารีของโอเพนซีวีใน  
การปรับเท่าอี้สโซตограм คำนวณค่าคาดหมาย และ  
ส่วนเบี่ยงมาตรฐาน