รายงานวิจัยฉบับสมบูรณ์

โครงการการรู้จำเสียงร้องที่ไม่ขึ้นกับเสียงรบกวนพื้นหลังโดยใช้แบบรูปสเปกโทรแกรมเชิงภาพ

โดย

ดร.พีระพล ขุนอาสา มหาวิทยาลัยราชภัฏอุตรดิตถ์

พฤษภาคม 2559

สัญญาเลขที่ TRG5780202

รายงานวิจัยฉบับสมบูรณ์

โครงการการรู้จำเสียงร้องที่ไม่ขึ้นกับเสียงรบกวนพื้นหลังโดยใช้แบบรูปสเปกโทรแกรมเชิงภาพ

ดร.พีระพล ขุนอาสา มหาวิทยาลัยราชภัฏอุตรดิตถ์

สนับสนุนโดยสำนักงานกองทุนสนับสนุนการวิจัย
และ มหาวิทยาลัยราชภัฏอุตรดิตถ์

กิตติกรรมประกาศ

โครงการวิจัยนี้จะสำเร็จลุล่วงไม่ได้ ถ้าไม่ได้รับความความอนุเคราะห์จาก ผู้ช่วยศาสตราจารย์ ศ.ดร.ชิดชนก เหลือสินทรัพย์ อาจารย์ที่ปรึกษาในระหว่างที่ผู้แต่งกำลังศึกษาในระดับปริญญาเอก ที่ถ่ายทอดวิชาความรู้ทางด้านการรู้จำแบบ และ หัวข้อวิจัยทางด้านชั้นข้อมูลอสมดุล นอกจากนี้ ข้าพเจ้ายังใครขอขอบพระคุณ สำนักงานกองทุนสนับสนุนการวิจัย (สกว.) ที่ได้โปรดมอบทุนทุนพัฒนาศักยภาพในการทำงานวิจัยของอาจารย์รุ่นใหม่ ประจำปีงบประมาณ 2557 เหนือสิ่งอื่นใด ผู้แต่งขอขอบพระคุณบุคคลทุกท่าน ซึ่งไม่อาจจะนำมากล่าวได้ทั้งหมด ที่ได้ให้ความช่วยเหลือสนับสนุนและให้กำลังใจด้วยดีตลอดมา

พีระพล ขุนอาสา

บทคัดย่อ

รหัสโครงการ: TRG5780202

ชื่อโครงการ: การรู้จำเสียงร้องที่ไม่ขึ้นกับเสียงรบกวนพื้นหลังโดยใช้แบบรูปสเปกโทรแกรมเชิงภาพ

ชื่อนักวิจัย: ดร.พีระพล ขุนอาสา

E-mail Address: peerapol_utt@hotmail.com, peerapol@uru.ac.th

ระยะเวลาโครงการ : 2 ปี

การรู้จำเสียงพูด (Speech Recognition) คือ การที่คอมพิวเตอร์สามารถรับรู้ เสียงของมนุษย์ได้โดยอัตโนมัติ โดยทั่วไปแล้วจะอาศัยระบบโปรแกรมคอมพิวเตอร์ที่สามารถแปลงเสียงพูด (Audio File) เป็นข้อความตัวอักษร (Text) โดยสามารถแจกแจงคำพูดต่างๆ ที่มนุษย์สามารถพูดใส่ไมโครโฟน โทรศัพท์หรืออุปกรณ์อื่นๆ และเข้าใจคำศัพท์ทุกคำอย่างถูกต้องเกือบ 100% โดยเป็นอิสระจากขนาดของกลุ่มคำศัพท์ ความดังของเสียงและลักษณะการออกเสียงของผู้พูด โดยระบบจะรับฟังเสียงพูดและตัดสินใจว่าเสียงที่ได้ยินนั้นเป็นคำๆใด โดยทั่วไปนั้นการรับรู้ของมนุษย์มีประสิทธิภาพสูงสามารถที่จะจำแนกประเภทเสียงต่างๆได้ แต่ปัญหาการรู้จำเสียง(Audio recognition) โดยใช้คอมพิวเตอร์นั้น คือการให้คอมพิวเตอร์สามารถจำแนกเสียงประเภทต่างๆ ไม่ว่าจะเป็น เสียงเพลง เสียงพูด หรือเสียงร้องได้เหมือนโสตประสาทของมนุษย์นั้นมีความซับซ้อนสูงและยากที่จะเลียนแบบ

ในงานวิจัยชิ้นนี้ เราสนใจในกลุ่มหัวข้อการสืบค้นข้อมูลจากเพลง(Music Information Retrieval) โดยเฉพาะปัญหาในการรู้จำคำร้อง(Singing voice recognition) จากเพลงนั้นมีความยุ่งยากและซับซ้อนมาก เนื่องจากคำที่ออกเสียงจากการร้องเพลงนั้นมีคุณสมบัติที่แตกต่างจากการพูดโดยทั่วๆ ไปหลายประการ เช่น รูปแบบการออกเสียงเฉพาะตัวบุคคลที่ขึ้นกับประเภทของดนตรี ระยะเวลาในการออกเสียงคำที่แตกต่างกันจากปัจจัยการเอื้อนหรือการลากเสียง ลูกคอในการร้องเพลง และจังหวะของเพลงต่างๆ จึงทำให้อัลกอลิทึมเดิมที่ใช้กับปัญหาการรู้จำเสียงพูดนั้นไม่ประสบผลสำเร็จเท่าที่ควร

อีกปัจจัยหนึ่งที่ลดทอนคุณภาพของการรู้จำคือเพลงจะมีเสียงดนตรีประกอบจะมีผลเทียบได้กับเสียงรบกวน(Noise) ซึ่งจะทำให้ประสิทธิภาพในการรู้ลดลง โดยทั่วไปแล้วการแก้ปัญหาคือการใช้ตัวรองเสียงรบกวน (Noise filter) ชนิดต่างๆ เข้ามากรองเสียงดนตรีออกไป แต่การใช้ตัวกรองเสียงรบกวนนั้นจะเป็นการเพิ่มขั้นตอนเข้าไปทำให้เวลาในการประมวลผลเพิ่มขึ้นตาม อีกทั้งการใช้ตัวรองเสียงรบกวนอาจไปทำลายเสียงที่ต้องการใช้จริงไปด้วย

ดังนั้นในงานวิจัยชิ้นนี้จึงมีจุดประสงค์หลักคือ รู้จำคำร้องต่างๆ ในเพลงโดยที่ไม่ใช้ตัวรองเสียงรบกวน (Noise filter) มากรองหรือขจัดเสียงดนตรี เสียงรบกวนออกไป งานวิจัยนี้อาศัยหลักการของการรู้จำรูปภาพเข้ามาประยุกต์ใช้ในการแก้ปัญหา อย่างแรกนำเอาสัญาณเสียงที่ได้ไปแปลงให้อยู่ในรูปแบบของภาพที่เราเรียกว่า สเปกโตรแกรม(Spectrogram) ที่มีลักษณะข้อมูลเป็นแบบ MxN มิติ แต่เนื่องจาก สเปกโตรแกรม(Spectrogram) มีขนาดของมิติข้อมูลที่สูงเกินไปการนำเอามาใช้งานโดยตรงนั้นเป็นไปได้ยากและต้องใช้เครื่องประมวลผลที่ประสิทธิภาพสูงมาก อีกทั้งคำร้องแต่ละคำเมื่อนำมาแปลงเป็น สเปกโตรแกรม(Spectrogram) ขนาดจะไม่เท่ากันไม่สามารถนำไปทำการรู้จำได้ ดังนั้นงานวิจัยนี้จึงนำเอา สเปกโตรแกรม(Spectrogram) มาลดขนาดของมิติข้อมูลลงก่อนนำไปทำการรู้จำด้วยเทคนิคการปรับขนาดภาพ(Image resizing algorithm) สำหรับขั้นตอนในการรู้จำนั้นงานวิจัยนี้เลือกใช้ Feed-Forward neural Network

จากการทดลองพบว่างงานวิจัยนี้สามารถแก้ปัญหาการรู้จำเสียงร้องหรือคำร้องในเพลงที่มีเสียงดนตรีพื้นหลังได้เป็นอย่างดี โดยประสิทธิภาพในการรู้จำอยู่ที่ 90.0% ขึ้นไปในทุก ๆ ชุดทดลอง อีกทั้งยังสามารถรู้จำได้หลายๆ ภาษาพร้อมๆ กันในชุดข้อมูลเดียวกันอีกด้วย

คำหลัก: การรู้จำแบบ, นิวรอนเน็ตเวิร์ค, ประมวลผลสัญญาณดิจิตอล, การสกัดคุณลักษณะเด่น

Abstract

Project Code: TRG5780202

Project Title: Single Signal Entity Approach for Thai Singing Word Recognition Using Images of Power Spectrogram and Image Processing Techniques

Investigator: Dr. Peerapol Khunarsa

E-mail Address: peerapol_utt@hotmail.com, peerapol@uru.ac.th

Project Period: 2 Years

Singing word recognition is one of the interesting research topics in the area of Music Information Retrieval (MIR). The first approach to solve this problem used successful techniques in Automatic Speech Recognition (ASR). Moving from monophonic to polyphonic audio signal, the problem has become more complex. The background instrumental accompaniment is regarded as the noise source degrading the performance of the recognition system. The papers proposed a statistical learning method for recognition of the word in a singing signal with background music and for classification of singing voice region in a polyphonic audio signal.

The goal of this paper is to solve singing word recognition without using any method to separated instrumental from background music. The papers also applied the concept of image recognition by using a สเปกโตรแกรม(Spectrogram) as an image to solve the problem. An audio signal that accompanies music was analyzed and transformed into a สเปกโตรแกรม(Spectrogram). A dimension of สเปกโตรแกรม (Spectrogram) is very high and time interval of each singing word is not equal. Then we apply image resizing algorithm to solve both problem. To recognize it, the whole สเปกโตรแกรม(Spectrogram) was sliced and formed as a feature vector for a neural classifier. Several classification functions are compared, such as Fisher classifier, K-nearest

neighbor and Feed-Forward can effectively recognize the word in music with the accuracy rate more than 90.0% Especially, we can recognize Cross-Language Music Data.

Keyword Spectrogram, Singing voice recognition, Automatic speech recognition (ASR), Feed-Forward Neural Network.

1. บทนำ

โดยทั่วไปนั้นการรับรู้ของมนุษย์มีประสิทธิภาพสูงสามารถที่จะจำแนกประเภทเสียงต่างๆได้ แต่ปัญหาการรู้จำเสียง(Audio recognition) โดยใช้คอมพิวเตอร์นั้น คือการให้คอมพิวเตอร์สามารถจำแนกเสียงประเภทต่างๆ ไม่ว่าจะเป็น เสียงเพลง เสียงพูด หรือเสียงร้องได้เหมือนโสตประสาทของมนุษย์นั้นมีความซับซ้อนสูงและยากที่จะเลียนแบบ

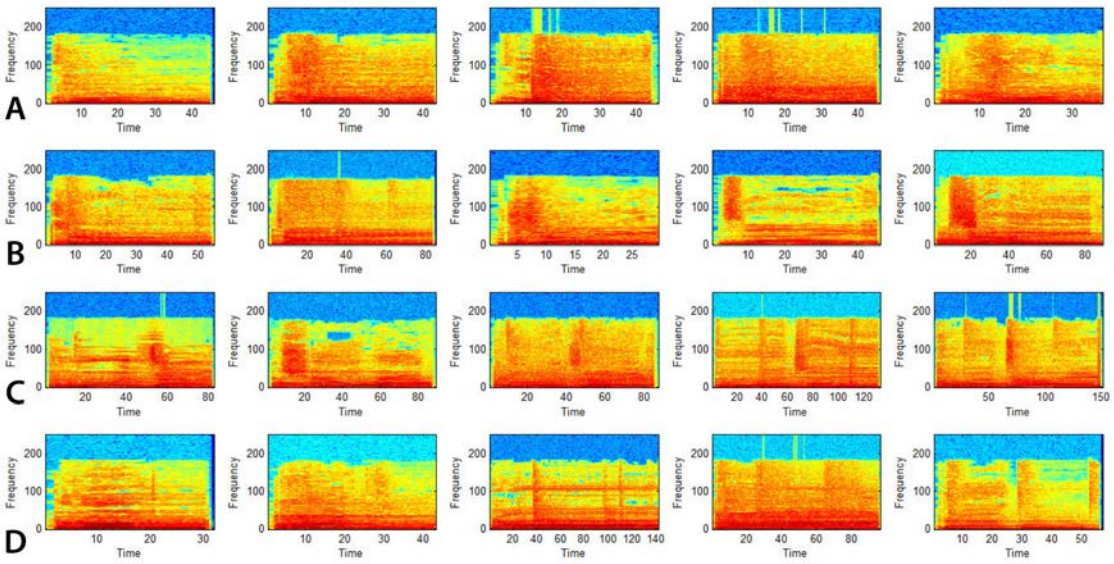
ในงานวิจัยชิ้นนี้ เราสนใจในกลุ่มหัวข้อการสืบค้นข้อมูลจากเพลง(Music Information Retrieval) โดยเฉพาะ ปัญหาในการรู้จำคำร้อง(Singing voice recognition) จากเพลงนั้นมีความยุ่งยากและซับซ้อนมาก เนื่องจากคำที่ออกเสียงจากการร้องเพลงนั้นมีคุณสมบัติที่แตกต่างจากการพูดโดยทั่วๆ ไปหลายประการ เช่น รูปแบบการออกเสียงเฉพาะตัวบุคคลที่ขึ้นกับประเภทของดนตรี ระยะเวลาในการออกเสียงคำที่แตกต่างกันจากปัจจัยการเอื้อนหรือการลากเสียง ลูกคอในการร้องเพลง และจังหวะของเพลงต่างๆ จึงทำให้อัลกอลิทึมเดิมที่ใช้กับปัญหาการรู้จำเสียงพูดนั้นไม่ประสบผลสำเร็จเท่าที่ควร

อีกปัจจัยหนึ่งที่ลดทอนคุณภาพของการรู้จำคือเพลงจะมีเสียงดนตรีประกอบจะมีผลเทียบได้กับเสียงรบกวน(Noise) ซึ่งจะทำให้ประสิทธิภาพในการรู้ลดลง โดยทั่วไปแล้วการแก้ปัญหาคือการใช้ตัวรองเสียงรบกวน (Noise filter) ชนิดต่างๆ เข้ามากรองเสียงดนตรีออกไป แต่การใช้ตัวกรองเสียงรบกวนนั้นจะเป็นการเพิ่มขั้นตอนเข้าไปทำให้เวลาในการประมวลผลเพิ่มขึ้นตาม อีกทั้งการใช้ตัวรองเสียงรบกวนอาจไปทำลายเสียงที่ต้องการใช้จริงไปด้วย

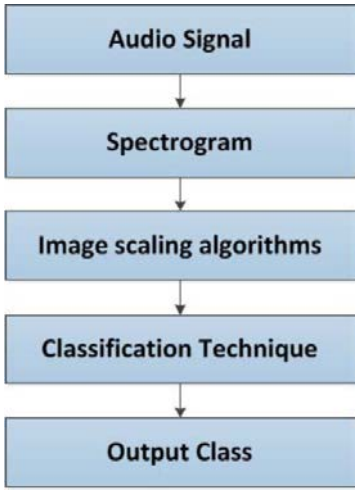
ดังนั้นในงานวิจัยชิ้นนี้จึงมีจุดประสงค์หลักคือ รู้จำคำร้องต่างๆ ในเพลงโดยที่ไม่ใช้ตัวรองเสียงรบกวน (Noise filter) มากรองหรือขจัดเสียงดนตรี เสียงรบกวนออกไป

2. ระเบียบวิธีวิจัย

ในงานวิจัยนี้ได้ทำการออกแบบวิธีการรู้จำคำร้องของนักร้องที่อยู่ในเพลงจะใช้อยู่บนพื้นฐานเดิมของงาน “Singing voice recognition based on matching of spectrogram pattern” ที่ได้ตีพิมพ์ในปี 2009 คือการรวบรวมและคัดเลือกข้อมูลคำจากเพลง จากนั้นใช้สมุติฐานที่ว่าการออกเสียงคำที่มีความเหมือนกันเมื่อนำมาการสกัดลักษณะ (Feature Extraction) ให้อยู่ในรูปแบบของสเปกโตรแกรม(Spectrogram) ซึ่งจะให้ภาพสเปกโตรแกรมที่ได้ที่มีลักษณะที่เหมือนกัน จากนั้นเราจึงนำเอาหลักการของการรู้จำงานประเภทลายนิ้วมือมาประยุกย์ใช้ตามภาพที่ 2



รูปที่ 1 รูปของสเปกโตรแกรม(Spectrogram) ของคำร้องตัวอย่าง 4 คำ



ภาพที่ 2: ขั้นตอนดำเนินงานวิจัย

2.1. ขั้นตอนการคำนวณรูปแบบของสเปกโตรแกรม(Spectrogram)

รูปแบบของสเปกโตรแกรม(Spectrogram) เป็นลักษณะการนำเสนอการกระจายตัวของกำลังงานเสียงระหว่างมิติความถี่ของสัญญาณ(Frequency) และเวลา(Times)ซึ่งมีลักษณะเป็นเหมือนรูปภาพแบบที่มีแกน X และแกน Y โดยแกน X แทนค่ากำลังงานเสียงที่กระจายตามเวลา

และ แกน Y แทนค่ากำลังงานเสียงที่กระจายตามความถี่ของสัญญาณกระจายซึ่งมีลักษณะตามภาพที่ 1

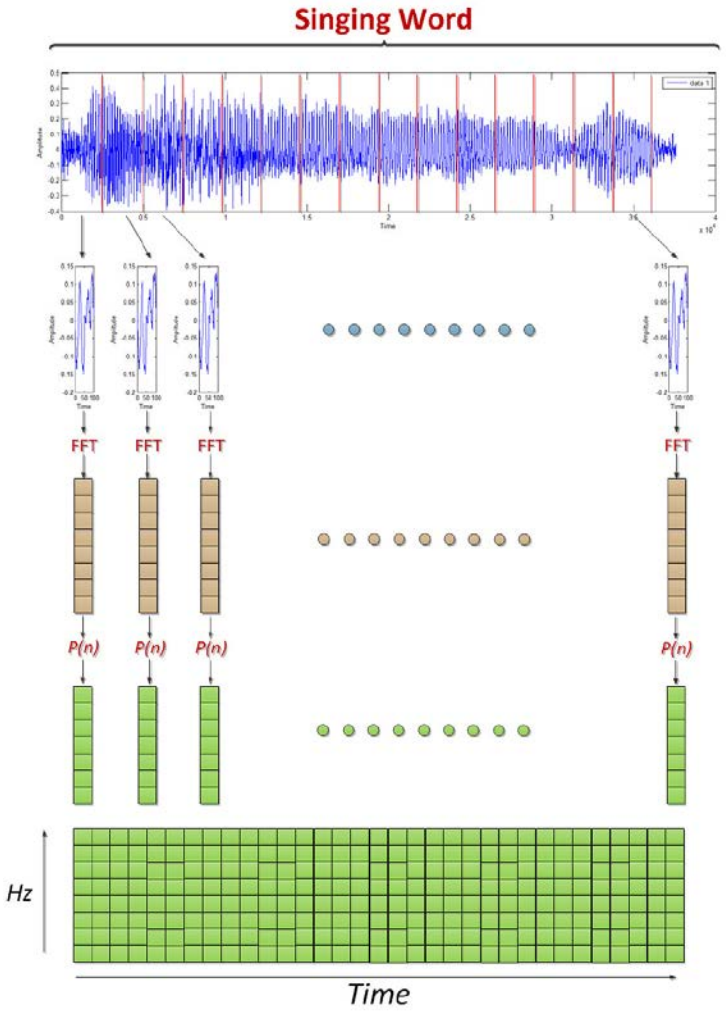
ขั้นแรกให้ $x(n), n = 0, 1, 2, ..., N-1$ เป็นลำดับของตัวเลขจำนวนจริงที่แทนสัญญาณข้อมูลเสียงของแต่ละคำร้องที่เข้ามา และ $X(m), m = 0, 1, 2, ..., N-1$ เป็นการคำนวณ DFT ของ $x(n)$ ตามสมการต่อไปนี้

$$X(k) = \sum_{n=0}^{N-1} w(n)x(n)exp(-\frac{2\pi kn}{N})$$

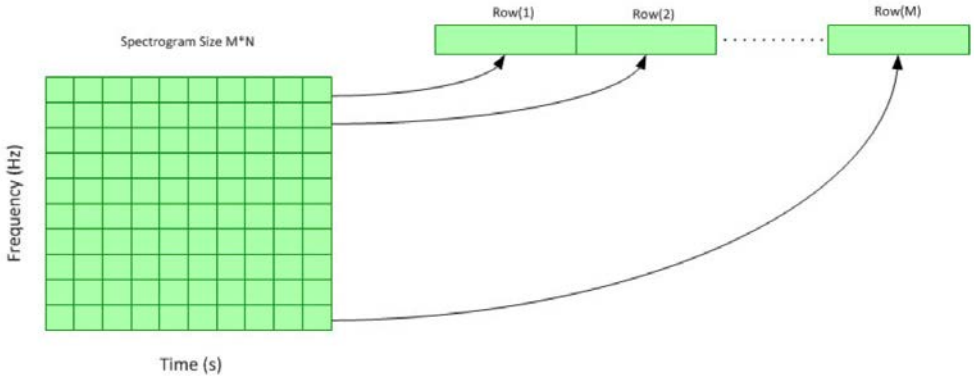
จากนั้นคำนวณ $P(k)$ ซึ่งแทนกำลังงานของแต่ละ $X(m)$ ที่คำนวณได้ตามสมการต่อไปนี้

$$P(k) = 10log_{10}(X(k))$$

จากนั้นนำเอากำลังงาน $P(k)$ ของแต่ละ $X(m)$ มาทำการสร้างเป็นแบบรูปสเปกโตรแกรม(Spectrogram) ของแต่ละคำร้องตามลำดับภาพที่ 3 ในงานวิจัยนี้เราได้นำเสนอการใช้ประบวนการรู้จำแบบนิวรอนเน็ตเวิร์ค ดังนั้นข้อมูลที่ใช้จำเป็นต้องอยู่ในรูปแบบของเวกเตอร์ที่เป็นลักษณะ 1 มิติแต่เนื่องจากแบบรูปสเปกโตรแกรม(Spectrogram) เป้นข้อมูลที่อยู่ในลักษณะของข้อมูล 2 มิติดังนั้นก่อนนำไปใช้งานจึงจำเป็นต้องแปลงให่อยู่ในรูปแบบ 1 มิติของ ขั้นตอนตามภาพที่ 4

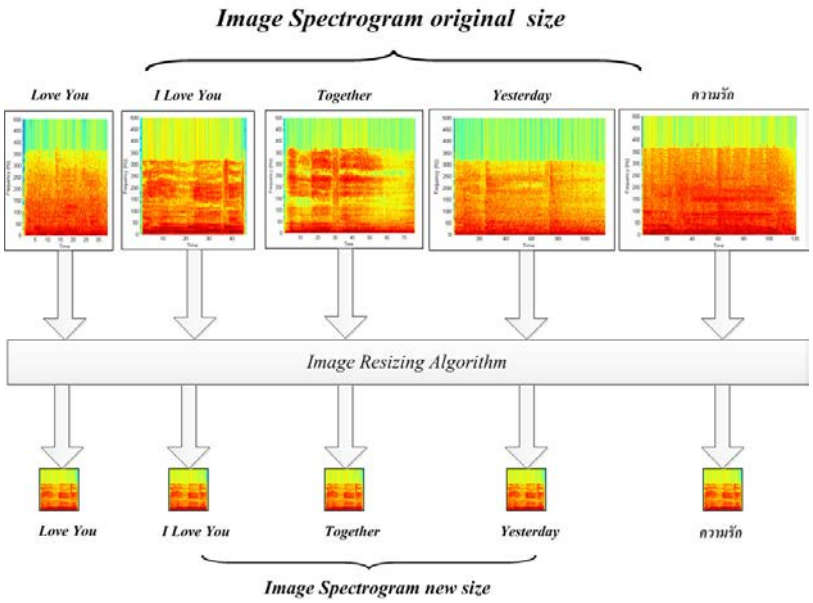


ภาพที่ 3: ขั้นตอนการคำนวณและสร้างสเปกโตรแกรม(Spectrogram) ของแต่ละคำร้อง



รูปที่ 4 การจัดเรียงสเปกโตรแกรม(Spectrogram) ให้อยู่ในรูปแบบ input vector ที่เหมาะสม สำหรับเตรียมเข้าสู่กระบวนการรู้จำ

2.2. การลดมิติของข้อมูลด้วย Image Scaling Algorithm หรือเทคนิคการย่อหรือขยายภาพ



ภาพที่ 5 การลดมิติของข้อมูลด้วย Image Scaling Algorithm

เมื่อได้คำที่ต้องการแล้วจึงนำมาสกัดลักษณะ (Feature Extraction) โดยให้อยู่ในรูปแบบของสเปกโตรแกรม(Spectrogram) แต่การใช้สเปกโตรแกรม(Spectrogram) ในการสกัดลักษณะนั้นมีปัญหาในการรู้จำอยู่คือ มิติของข้อมูลที่ได้นั้นสูงมากเมื่อเทียบกับการสกัดลักษณะตัวอื่นๆ ดังนั้นในงานวิจัยนี้จึงเลือกที่จะใช้เทคนิคการย่อหรือขยายภาพ (Scaling) มาเพื่อลดขนาดของภาพสเปกโตรแกรมลงซึ่งจะลดขนาดของข้อมูลลงด้วยตามภาพที่ 5 โดยเทคนิคที่เลือกใช้คือ nearest neighbor sampling, bilinear interpolation, bicubic interpolation และ discrete cosine transform-based compression เป็นต้น

2.3 การรู้จำข้อมูลเสียงด้วยโครงข่ายประสาทเทียม (Artificial Neural Networks)

โครงข่ายประสาทเทียม (Artificial Neural Networks) หรือที่มักจะเรียกสั้น ๆ ว่า โครงข่ายประสาท (Neural Networks หรือ Neural Net) เป็นหนึ่งในเทคนิคของการทำเหมืองข้อมูล (Data Mining) คือโมเดลทางคณิตศาสตร์ สำหรับประมวลผลสารสนเทศด้วยการคำนวณแบบคอนเนคชันนิสต์ (Connectionist) เพื่อจำลองการทำงานของเครือข่ายประสาทในสมองมนุษย์ ด้วยวัตถุประสงค์ที่จะสร้างเครื่องมือซึ่งมีความสามารถในการเรียนรู้การจดจำรูปแบบ(Pattern Recognition) และการสร้างความรู้ใหม่ (Knowledge Extraction) เช่นเดียวกับความสามารถที่มีใน

สมองมนุษย์ แนวคิดเริ่มต้นของเทคนิคนี้ได้มาจากการศึกษาโครงข่ายไฟฟ้าชีวภาพ (Bioelectric Network) ในสมอง ซึ่งประกอบด้วย เซลล์ประสาท หรือ "นิวรอน" (Neurons) และ "จุดประสานประสาท" (Synapses) แต่ละเซลล์ประสาทประกอบด้วยปลายในการรับกระแสประสาท เรียกว่า "เดนไดรท์" (Dendrite) ซึ่งเป็น input และปลายในการส่งกระแสประสาทเรียกว่า "แอคซอน" (Axon) ซึ่งเป็นเหมือน output ของเซลล์ เซลล์เหล่านี้ทำงานด้วยปฏิกิริยาไฟฟ้าเคมี เมื่อมีการกระตุ้นด้วยสิ่งเร้าภายนอกหรือกระตุ้นด้วยเซลล์ด้วยกัน กระแสประสาทจะวิ่งผ่านเดนไดรท์เข้าสู่นิวเคลียสซึ่งจะเป็นตัวตัดสินว่าต้องกระตุ้นเซลล์อื่น ๆ ต่อหรือไม่ ถ้ากระแสประสาทแรงพอนิวเคลียสก็จะกระตุ้นเซลล์อื่น ๆ ต่อไปผ่านทางแอคซอนของมัน

นักวิจัยส่วนใหญ่ในปัจจุบันเห็นตรงกันว่าโครงข่ายประสาทเทียมมีโครงสร้างแตกต่างจากโครงข่ายในสมอง แต่ก็ยังเหมือนสมอง ในแง่ที่ว่าโครงข่ายประสาทเทียม คือการรวมกลุ่มแบบขนานของหน่วยประมวลผลย่อย ๆ และการเชื่อมต่อนี้เป็นส่วนสำคัญที่ทำให้เกิดสติปัญญาของโครงข่าย เมื่อพิจารณาขนาดแล้วสมองมีขนาดใหญ่กว่าโครงข่ายประสาทเทียมอย่างมาก รวมทั้งเซลล์ประสาทยังมีความซับซ้อนกว่าหน่วยย่อยของโครงข่าย อย่างไรก็ดีหน้าที่สำคัญของสมอง เช่น การเรียนรู้ยังคงสามารถถูกจำลองขึ้นอย่างง่ายด้วยโครงข่ายประสาทนี้

สำหรับหลักการในคอมพิวเตอร์ Neurons ประกอบด้วย input และ output เหมือนกัน โดยจำลองให้ input แต่ละอันมี weight เป็นตัวกำหนดน้ำหนักของ input โดย neuron แต่ละหน่วยจะมีค่า threshold เป็นตัวกำหนดว่าน้ำหนักรวมของ input ต้องมากขนาดไหนจึงจะสามารถส่ง output ไปยัง neurons ตัวอื่นได้ เมื่อนำ neuron แต่ละหน่วยมาต่อกันให้ทำงานร่วมกันการทำงานนี้ในทางตรรกแล้วก็จะเหมือนกับปฏิกิริยาเคมีที่เกิดในสมอง เพียงแต่ในคอมพิวเตอร์ทุกอย่างเป็นตัวเลขเท่านั้นเอง

การทำงานของ Neural Networks คือเมื่อมี input เข้ามายัง network ก็เอา input มาคูณกับ weight ของแต่ละขา ผลที่ได้จาก input ทุก ๆ ขาของ neuron จะเอามารวมกันแล้วก็เอามาเทียบกับ threshold ที่กำหนดไว้ ถ้าผลรวมมีค่ามากกว่า threshold แล้ว neuron ก็จะส่ง output ออกไป output นี้ก็จะถูกส่งไปยัง input ของ neuron อื่น ๆ ที่เชื่อมกันใน network ถ้าค่าน้อยกว่า threshold ก็จะไม่เกิด output สิ่งสำคัญคือเราต้องทราบค่า weight และ threshold สำหรับสิ่งที่เราต้องการเพื่อให้คอมพิวเตอร์รู้จำ ซึ่งเป็นค่าที่ไม่แน่นอน แต่สามารถกำหนดให้คอมพิวเตอร์ปรับค่าเหล่านั้นได้โดยการสอนให้มันรู้จัก pattern ของสิ่งที่เราต้องการให้มันรู้จำ เรียกว่า "back propagation" ซึ่งเป็นกระบวนการย้อนกลับของการรู้จำ ในการฝึก feed-forward Neural Networks จะมีการใช้อัลกอริทึมแบบ back-propagation เพื่อใช้ในการปรับปรุงน้ำหนักคะแนนของเครือข่าย (Network Weight) หลังจากใส่รูปแบบข้อมูลสำหรับฝึกให้แก่เครือข่ายในแต่ละครั้งแล้ว ค่าที่ได้รับ (output) จากเครือข่ายจะถูกนำไปเปรียบเทียบกับผลที่คาดหวัง แล้วทำการคำนวณหาค่าความผิดพลาด ซึ่งค่าความผิดพลาดนี้จะถูกส่งกลับเข้าสู่เครือข่ายเพื่อใช้แก้ไขค่าน้ำหนักคะแนนต่อไป

การเรียนรู้สำหรับ Neural Networks มีทั้งหมด 2 แบบประกอบไปด้วย

1. Supervised Learning การเรียนแบบมีการสอน

เป็นการเรียนแบบที่มีการตรวจคำตอบเพื่อให้โครงข่ายประสาทเทียมปรับตัว ชุดข้อมูลที่ใช้สอนโครงข่ายประสาทเทียมจะมีคำตอบไว้คอยตรวจดูว่าโครงข่ายประสาทเทียมให้คำตอบที่ถูกหรือไม่ ถ้าตอบไม่ถูก โครงข่ายประสาทเทียมก็จะปรับตัวเองเพื่อให้ได้คำตอบที่ดีขึ้น (เปรียบเทียบกับคน เหมือนกับการสอนนักเรียนโดยมีครูผู้สอนคอยแนะนำ)

2. Unsupervised Learning การเรียนแบบไม่มีการสอน

เป็นการเรียนแบบไม่มีผู้แนะนำ ไม่มีการตรวจคำตอบว่าถูกหรือผิด โครงข่ายประสาทเทียมจะจัดเรียงโครงสร้างด้วยตัวเองตามลักษณะของข้อมูล ผลลัพธ์ที่ได้ โครงข่ายประสาทเทียมจะสามารถจัดหมวดหมู่ของข้อมูลได้ (เปรียบเทียบกับคน เช่น การที่เราสามารถแยกแยะพันธุ์พืช พันธุ์สัตว์ตามลักษณะรูปร่างของมันได้เองโดยไม่มีใครสอน)

3. การจัดเตรียมข้อมูลที่ใช้ในการทดลอง (DATA COLLECTION)

เริ่มต้นจากข้อมูลที่ใช้ในการทดลองงานวิจัยนี้จะเลือกคำจากเพลงที่วางขายในท้องตลาดทั่วๆ ไป เนื่องมาจากฐานข้อมูลที่เกี่ยวข้องกับงานของ Singing Word Recognition ยังไม่มี เพื่อสร้างความน่าเชื่อถือให้กับข้อมูลนั้นในงานวิจัยชิ้นนี้จะพิจารณาในหลายๆ ปัจจัยเพื่อให้ข้อมูลมีความหลากหลายประกอบไปด้วย เช่น

1. คำร้องที่ทำมารู้จำในงานวิจัยนี้ จะใช้ทั้งภาษาไทยและอังกฤษ โดยแบ่งออกเป็น 2 กลุ่มข้อมูลคือ คำเดียว(isolate word) และ คำประสม (Compound Word) ประกอบไปด้วย
    i. ข้อมูลชุดที่ 1 ประกอบไปด้วยคำไทยจำนวน 12 คำ { คน, ความ, เคย, ใคร, ใจ, ฉัน, ที่, เธอ, มี, รัก, รู้, เรา} สำหรับกลุ่มคำเดียว(isolate word) โดยทำการเลือกจากเพลงไทยจำนวนมากกว่า 1500 อัลบัม โดยในแต่ละคำจะมีไฟล์เสียงทั้งหมด 600 ไฟล์ของคำร้องนั้นๆ โดยรายละเอียดแสดงในตารางที่ 1 และ 2
    ii. ข้อมูลชุดที่ 2 ประกอบไปด้วยคำไทยและภาษาอังกฤษจำนวน 12 คำ { I love you, Love you, Together, Tomorrow, Yesterday, ความรัก, คิดถึง, ใครสักคน, ไม่เคย, ไม่มี, รักเธอ, หัวใจ } สำหรับกลุ่มคำประสม (Compound Word) โดยทำการเลือกจากเพลงไทยจำนวนมากกว่า 1500 อัลบัม โดยในแต่ละคำจะมีไฟล์เสียงทั้งหมด 600 ไฟล์ของคำร้องนั้นๆ โดยรายละเอียดแสดงในตารางที่ 3 และ 4

2. คำร้องต่างๆ ในเพลงที่ใช้ในงานวิจัยนี้ จะไม่ใช้ตัวรองเสียงรบกวน (Noise filter) มาทำการกรองหรือขจัดเสียงดนตรี เสียงรบกวนออกไป
3. ในงานวิจัยนี้จะใช้เพลงที่วางจำหน่ายอยู่บนท้องตลาดทั่วๆ ไป ไม่อัดเสียงด้วยตัวเอง
4. แนวเพลงที่ใช้ในงานวิจัยนี้จะใช้แนวเพลงที่หลายหลายประกอบไปด้วย Rock ,hard rock, Soft Rock , Dance , Hip-Pop, Soul, R&B ,folk และ Acoustic ที่มาจากนักร้องหลายคน
5. เพลงที่ใช้จะเลือกมาจากนักร้องทั้งชายและหญิง

ข้อมูลไฟล์เสียงที่ใช้ในทุกคำร้องนั้นจะถูกจัดการด้วยโปรแกรม Sony Sound Forge program. โดยคุณสมบัติของเสียงที่ใช้คือ Sampling Rate ที่ 44.2 kHz ด้วย 128/s bit rate.

ตารางที่ 1 ประเภทและจำนวนที่ใช้ในข้อมูลชุดที่ 1

Music Genres	Male Singer	Female Singer	Total
Pop Rock	1,768	1,545	3,313
hard rock	978	667	1,645
soft rock	2,284	2,100	4,384
dance	1,177	467	1,644
hip-pop	304	160	464
soul	250	108	358
R&B	1,135	652	1,787
folk	297	162	459
Acoustic	1,288	982	2,270
	Total		16324

ตารางที่ 2 คำร้องภาษาไทยและจำนวนที่ใช้ในข้อมูลชุดที่ 1

Class.	Singing word	Time duration(min-max)	Pronounce (in Thai)
1	”คน”	0.65s-2.95s	”kon”
2	”ความ”	0.26s-0.60s	”kwarm”
3	”เคย”	0.33s-0.62s	”koey”
4	”ใคร”	0.33s-0.70s	”krai”
5	”ใจ”	0.44s-1.38s	”jai”
6	”ฉัน”	0.26s-1.23s	”chan”
7	”ที่”	0.26s-0.54s	”tee”
8	”เธอ”	0.23s-0.78s	”ther”
9	”มี”	0.28s-0.86s	”mai”
10	”รัก”	0.18s-1.48s	”luck”
11	”รู้”	0.28s-0.47s	”roo”
12	”เรา”	0.26s-0.73s	”raw”

ตารางที่ 3 ประเภทและจำนวนที่ใช้ในข้อมูลชุดที่ 2

Music Genres	Male Singer	Female Singer	Total
Pop Rock	1,105	1,432	2,537
hard rock	1,734	503	2,237
soft rock	4,473	1,466	5,939
dance	1,121	964	2,085
hip-pop	162	149	311
soul	208	358	566
R&B	1,155	840	1,995
folk	329	355	684
Acoustic	462	940	1,402
	Total		17756

ตารางที่ 4 คำร้องภาษาไทยและภาษาอังกฤษที่ใช้ในข้อมูลชุดที่ 2

Class.	Singing word	Time duration(min-max)	Pronounce (in Thai)
1	I love you	0.65s-2.95s	
2	Love you	0.57s-2.92s	
3	Together	1.04s-2.11s	
4	Tomorrow	1.07s-6.63s	
5	Yesterday	0.81s-5.90s	
6	”ความรัก”	0.52s-3.65s	”kwarm-luck”
7	”คิดถึง”	0.88s-1.11s	”kit-thun”
8	”ใครสักคน”	0.99s-4.62s	”krai-sak-kon”
9	”ไม่เคย”	0.41s-1.99s	”mai-koey ”
10	”ไม่มี”	0.57s-1.17s	”mai-mee”
11	”รักเธอ”	0.47s-1.93s	”luck-ther”
12	”หัวใจ”	0.73s-1.46s	”hua-jai”

4. การทดลองและสรุปผล (EXPERIMENTAL EVALUATION)

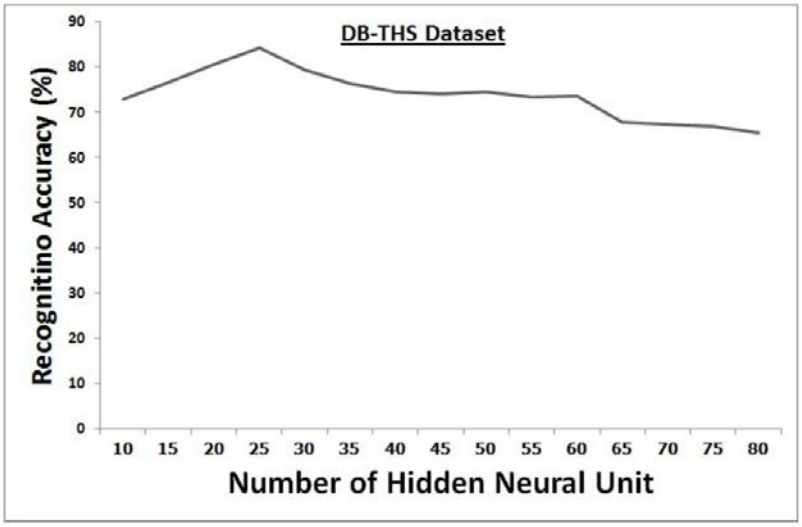
สำหรับการทดลองเพื่อหาประสิทธิภาพการทดลองทั้งหมดใช้งานโปรแกรม Matlab 2012a และทำงานบนเครื่อง Intel Dual Core E6750 2.66 GHz หน่วยความจำขนาด 6G และเพื่อความแม่นยำและไม่คลาดเคลื่อนของผลการทดลอง งานทดลองนี้จึงใช้การวัดประสิทธิภาพการแบ่งชุดข้อมูลในการฝึกฝนและทดสอบนั้นงานนี้เลือกใช้หลักการ เคโฟลด์ครอสวาลิเดชัน(K-Flow Cross validation) โดยใช้ K = 5 ซึ่งจะทำการทดลองซ้ำทั้งหมด 50 ครั้งและนำผลลัพย์ที่ได้ทั้งหมดมาหาค่าเฉลี่ยของประสิทธิภาพ

จากนั้นจะประยุกต์ใช้เทคนิคที่ใช้ในการรู้จำที่หลากหลายประกอบไปด้วย Fisher classifier, K-nearest neighbor, Feed-Forward, Naive Bayes Classifier, Parzen Classifier และ Decision tree เข้าไปรู้จำข้อมูล

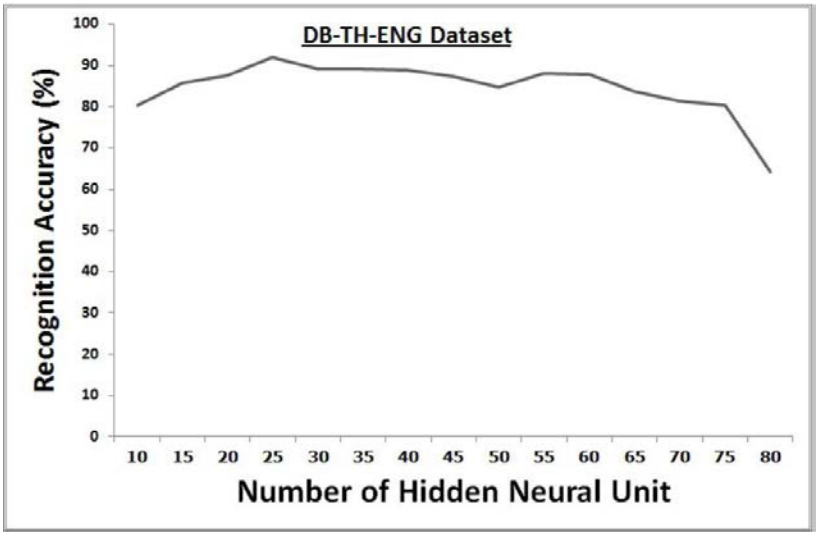
4.1 จำนวนนิวรอนภายในโครงข่ายประสาทเทียม (Artificial Neural Networks) กับประสิทธิภาพ ในการรู้จำหรือไม่

ในงานวิจัยนี้ได้ใช้โครงข่ายประสาทเทียม (Artificial Neural Networks) แบบ three-layer feed-forward network สำหรับการรู้จำเสียงร้องหรือคำร้องในข้อมูลทดลองชุดที่ 1 และ 2 ซึ่งการใช้โครงข่ายประสาทเทียม (Artificial Neural Networks) จำนวนนิวรอนภายในโครงข่ายมีผลกับประสิทธิภาพในการรู้จำ ดังนั้นเราจึงจำเป็นต้องหาจำนวนนิวรอนจำนวนเท่าใดที่เหมาะสมกับการใช้งาน

ในรู้ที่ 6 และ 7 เป็นกราฟแสดงประสิทธิภาพในการรู้จำคำร้องในชุดข้อมูลที่ 1 และ 2 ที่ทดลองโดยใช้จำนวนนิวรอนภายในโครงข่ายที่แตกต่างกัน ซึ่งใช้สเปกโตรแกรม(Spectrogram) ที่สร้างโดยใช้วินโดว์(Windows Length) ขนาด 512 จุด ที่มีการทับซ้อนกัน 25% ของสัญญาณเสียงที่เข้ามาในแต่ละคำร้อง จากนั้นทำการลดขนาดของ สเปกโตรแกรม(Spectrogram) ให้อยู่ที่ขนาด 128x5 จุด ด้วยเทคนิค DCT-based compressed (สำหรับขนาดของวินโดว์ที่ใช้ในการสร้าง สเปกโตรแกรม(Spectrogram) จะถูกอธิบายในส่วนถัดไป)



รูปที่ 6 กราฟแสดงประสิทธิภาพในการรู้จำคำร้องในชุดข้อมูลที่ 1 ที่ทดลองโดยใช้จำนวนนิวรอนภายในโครงข่ายที่แตกต่างกัน

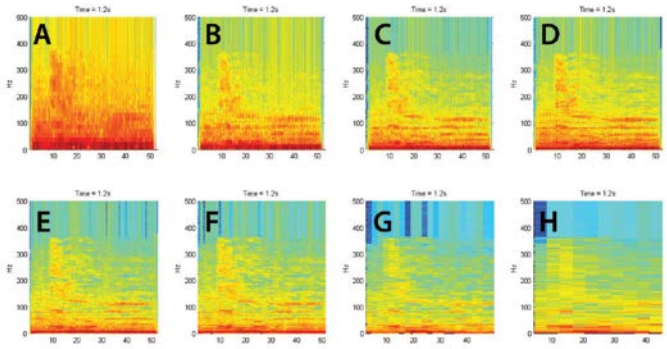


รูปที่ 7 กราฟแสดงประสิทธิภาพในการรู้จำคำร้องในชุดข้อมูลที่ 2 ที่ทดลองโดยใช้จำนวนนิวรอนภายในโครงข่ายที่แตกต่างกัน

จากข้อมูลที่แสดงภาพที่ 6 และ 7 จึงพอสรุปได้ว่า เราสามารถใช้สเปกโตรแกรม (Spectrogram) ร่วมกับโครงข่ายประสาทเทียม (Artificial Neural Networks) ในการรู้จำคำร้องโดยไม่กรองเสียงพื้นหลังได้จริงและให้ประสิทธิภาพที่สูงกว่า 90% และจำนวนนิวรอนภายในโครงข่ายมีผลกับประสิทธิภาพในการรู้จำจริง และจำนวนนิวรอนที่ดีที่สุดในการใช้งานจะอยู่ในช่วง 20-30 นิวรอน

4.2 ขนาดของวินโดว์(Windows Length) ที่ใช้ในการสร้างสเปกโตรแกรม (Spectrogram) มีผลกับประสิทธิภาพ ในการรู้จำหรือไม่

อีกปัจจัยที่สำคัญของสเปกโตรแกรม(Spectrogram) ก็คือ Windows Length ซึ่งจะมีผลกับสเปกโตรแกรม(Spectrogram)โดยตรงตามตัวอย่างของภาพที 8 ที่แสดงลักษณะของ สเปกโตรแกรมที่สร้างจาก Windows Length ที่มีความแตกต่างกัน ดังนั้นในงานวิจัยนี้จึงเลือกขนาดของ Windows Length ในการทดลองประกอบไปด้วย 64, 128, 256, 512, 1024, 2048, 4096 และ 8192 เพื่อหาขนาดของ Windows Length ที่มีประสิทธิภาพสูงสุด



ภาพที 8 ลักษณะของ สเปกโตรแกรมที่สร้างจาก Windows Length ที่มีความแตกต่างกัน a) 64, b) 128, c) 256, d) 512, e) 1024, f) 2048, g) 4096, h) 8192.

ในส่วนนี้ได้ทดลองกับข้อมูลชุดที่ 1 และ 2 ที่แสดงในตารางที่ 2 และ 4 โดยกำหนดตัวแปรควบคุมคือ ใช้สเปกโตรแกรมที่ใช้จะสร้างจากวินโดว์(Windows Length) ขนาด 1024 512 256 128 จุด และมีการทับซ้อนกันของสัญาณเสียง 25% ของขนาดวินโดว์(Windows Length)

จากตารางที่ 5 ถึง 12 เป็นตารางแสดงข้อมูลโดยเราจะเห็นว่าการใช้วินโดว์(Windows Length) ที่แตกต่างมีผลกับประสิทธิภาพในการรู้จำ สเปกโตรแกรมที่สร้างจาก Windows Length ขนาดใหญ่จะให้ประสิทธิภาพในการรู้จำที่สูงกว่า สเปกโตรแกรมที่สร้างจาก Windows Length ที่มีขนาดเล็กทั้ง 2 ชุดข้อมูล

ดังนั้นจึงสรุปได้ว่าขนาดขนาดของวินโดว์(Windows Length) ที่ใช้ในการสร้างสเปกโตรแกรม(Spectrogram) มีผลกับประสิทธิภาพในการรู้จำ

Table 5 : Average accuracy of all classes from a feedforward neural network having 25 hidden neurons with DB-TH Data set using a spectrogram window of 1024 Pixel

Size of the y-axis of spectrogram	Size of the x-axis of spectrogram									
	1 (Pixel)	2 (Pixel)	3 (Pixel)	4 (Pixel)	5 (Pixel)	6 (Pixel)	7 (Pixel)	8 (Pixel)	9 (Pixel)	10 (Pixel)
512 Pixel	52.374	70.276	*85.865	*88.374	*88.276	*87.606	*87.095	*87.951	*87.419	*87.036
461 Pixel	61.918	79.080	*83.941	*82.711	*82.227	*81.287	*83.527	*84.447	*85.649	*87.468
410 Pixel	62.791	72.020	*81.396	*82.430	*84.591	*83.534	*81.327	*85.938	*85.163	*84.790
358 Pixel	63.159	76.706	*80.801	*81.923	*81.652	*82.463	*84.407	*84.814	*83.034	*84.604
307 Pixel	60.138	70.115	73.491	*81.023	*82.218	*84.670	*83.163	*83.161	*83.080	*84.552
256 Pixel	58.548	75.343	*81.118	*84.782	*85.034	*85.622	*86.910	*84.205	*85.844	*84.926
205 Pixel	60.591	71.616	72.657	70.489	*80.034	*80.736	*82.693	*82.836	*81.990	*84.580
154 Pixel	49.714	73.438	74.069	72.466	66.483	*80.023	*80.502	*83.547	*84.087	*83.453
102 Pixel	54.069	63.514	*82.483	66.522	78.785	70.056	*80.286	63.593	62.529	77.051
51 Pixel	47.626	67.225	67.823	68.250	64.775	74.640	76.460	77.458	64.217	64.906

Table 6: Average accuracy of all classes from a feedforward neural network having 25 hidden neurons with DB-TH-ENG Data set using a spectrogram window of 1024 Pixel

Size of the y-axis of spectrogram	Size of the x-axis of spectrogram									
	1 (Pixel)	2 (Pixel)	3 (Pixel)	4 (Pixel)	5 (Pixel)	6 (Pixel)	7 (Pixel)	8 (Pixel)	9 (Pixel)	10 (Pixel)
512 Pixel	67.730	*88.710	*90.054	*94.062	*94.919	*94.394	*94.378	*94.788	*94.942	*94.865
461 Pixel	76.162	*84.687	*84.672	*86.425	*88.371	*93.320	*93.212	*94.440	*94.556	*94.672
410 Pixel	73.869	*84.170	*90.834	*92.811	*92.602	*93.266	*94.479	*94.934	*94.792	*94.726
358 Pixel	75.093	*86.139	*87.579	*92.564	*88.263	*93.614	*93.340	*94.734	*94.865	*94.212
307 Pixel	74.255	*84.313	*87.247	*87.151	*90.626	*92.718	*93.517	*94.776	*94.023	*94.224
256 Pixel	74.236	*83.066	*87.807	*91.904	*91.857	*92.564	*92.305	*94.961	*94.479	*94.409
205 Pixel	69.815	*81.093	*85.259	*89.544	*90.166	*90.888	*90.336	*91.873	*92.869	*92.278
154 Pixel	65.568	*83.112	*84.143	*86.413	*89.884	*89.251	*90.830	*91.409	*92.174	*92.938
102 Pixel	69.745	76.077	*89.027	*88.104	*88.475	*88.490	*89.398	*89.510	*89.220	*89.023
51 Pixel	58.382	76.637	*83.224	*85.185	*86.606	*87.216	*87.896	*87.266	*87.475	*88.637

Table 7: Average accuracy of all classes from a feedforward neural network having 25 hidden neurons with DB-TH Data set using a spectrogram window of 512 Pixel

Size of the y-axis of spectrogram	Size of the x-axis of spectrogram									
	1 (Pixel)	2 (Pixel)	3 (Pixel)	4 (Pixel)	5 (Pixel)	6 (Pixel)	7 (Pixel)	8 (Pixel)	9 (Pixel)	10 (Pixel)
256 Pixel	50.858	70.585	75.490	*80.651	*81.062	*81.368	*81.283	*81.007	*81.332	*82.337
230 Pixel	50.229	70.407	78.242	78.719	79.986	*80.742	*80.588	*80.796	*81.545	**82.111
205 Pixel	49.250	72.203	76.007	77.209	79.339	79.986	*80.690	*80.347	*81.234	*82.861
179 Pixel	48.690	71.192	73.463	77.700	77.002	79.112	79.906	*80.114	*80.164	*81.003
154 Pixel	53.392	62.224	76.590	79.742	79.329	*80.973	*80.568	*81.748	*84.711	*85.341
128 Pixel	46.217	71.141	78.163	76.763	75.322	*80.328	*80.878	*81.885	*83.199	*83.802
102 Pixel	50.574	67.197	73.644	76.343	76.821	76.571	*81.468	76.767	*82.240	*82.536
77 Pixel	43.628	62.732	73.399	77.178	76.551	76.087	76.033	77.615	77.633	79.062
51 Pixel	46.725	64.578	67.725	69.666	72.526	75.507	73.295	77.039	77.215	77.621
26 Pixel	41.048	62.873	64.191	67.246	69.663	73.030	73.332	74.264	75.795	77.758

Table 8 : Average accuracy of all classes from a feedforward neural network having 25 hidden neurons with DB-TH-ENG Data set using a spectrogram window of 512 Pixel

Size of the y-axis of spectrogram	Size of the x-axis of spectrogram									
	1 (Pixel)	2 (Pixel)	3 (Pixel)	4 (Pixel)	5 (Pixel)	6 (Pixel)	7 (Pixel)	8 (Pixel)	9 (Pixel)	10 (Pixel)
256 Pixel	59.152	73.542	*81.299	*81.286	*87.489	*87.405	*88.831	*88.339	*87.248	*89.124
230 Pixel	60.068	74.856	*80.699	*81.202	*83.744	*86.548	*87.260	*87.730	*89.535	*89.439
205 Pixel	64.315	74.974	*80.927	*81.713	*83.417	*86.599	*86.375	*87.709	*87.165	*88.160
179 Pixel	62.906	69.645	*80.722	*82.351	*83.120	*86.191	*86.382	*87.264	*87.339	*87.855
154 Pixel	60.816	68.411	71.009	*81.977	*82.949	*85.879	*85.795	*87.362	*86.099	*87.018
128 Pixel	62.636	73.745	78.260	*81.662	*82.366	*84.705	*85.013	*86.826	*86.124	*87.750
102 Pixel	62.876	66.404	72.636	*81.640	*82.876	*83.976	*85.209	*85.131	*85.867	*86.880
77 Pixel	62.391	69.804	77.398	79.197	79.967	*80.000	*84.801	*85.580	*85.248	*86.037
51 Pixel	54.916	63.721	73.781	74.072	*80.425	*82.055	*82.398	*82.039	*82.964	*83.507
26 Pixel	47.557	59.210	65.170	77.574	72.120	78.030	79.990	74.886	*82.591	*81.485

Table 9 : Average accuracy of all classes from a feedforward neural network having 25 hidden neurons with DB-TH Data set using a spectrogram window of 256 Pixel

Size of the y-axis of spectrogram	Size of the x-axis of spectrogram									
	1 (Pixel)	2 (Pixel)	3 (Pixel)	4 (Pixel)	5 (Pixel)	6 (Pixel)	7 (Pixel)	8 (Pixel)	9 (Pixel)	10 (Pixel)
128 Pixel	47.008	61.143	62.755	71.463	75.665	78.338	79.954	79.924	*80.026	*80.494
115 Pixel	45.491	60.693	62.146	70.647	73.701	80.466	78.200	75.645	*80.982	*81.870
102 Pixel	45.314	60.154	61.228	70.253	71.504	72.309	78.982	79.757	*80.458	*82.386
90 Pixel	43.041	63.987	70.857	71.530	77.005	78.195	78.133	78.322	*80.539	*81.660
77 Pixel	50.299	64.197	66.535	69.560	72.020	75.933	78.034	79.010	*80.000	*80.598
64 Pixel	34.378	54.962	47.842	77.780	80.342	67.402	78.235	71.350	75.455	76.969
51 Pixel	43.685	65.274	71.901	75.048	76.972	67.396	70.877	70.836	69.478	73.800
38 Pixel	38.023	56.118	73.498	70.502	64.007	66.102	68.539	69.901	69.130	72.338
26 Pixel	40.420	55.061	52.946	70.691	54.752	66.076	67.199	69.342	69.281	72.146
13 Pixel	38.903	50.417	58.765	71.967	64.762	69.596	66.660	69.530	72.558	72.893

Table 10 : Average accuracy of all classes from a feedforward neural network having 25 hidden neurons with DB-TH-ENG Data set using a spectrogram window of 256 Pixel

Size of the y-axis of spectrogram	Size of the x-axis of spectrogram									
	1 (Pixel)	2 (Pixel)	3 (Pixel)	4 (Pixel)	5 (Pixel)	6 (Pixel)	7 (Pixel)	8 (Pixel)	9 (Pixel)	10 (Pixel)
128 Pixel	57.768	68.116	79.042	*82.224	*84.710	*84.730	*86.826	*86.046	*87.753	*88.347
115 Pixel	50.409	71.413	70.672	73.251	77.892	77.869	*80.992	*80.324	*83.066	*87.375
102 Pixel	52.934	76.633	77.537	72.054	76.463	74.757	*80.286	*87.772	*84.208	*85.614
90 Pixel	59.869	73.745	69.568	73.598	74.938	77.552	*81.297	*87.676	*82.232	*80.625
77 Pixel	49.189	55.483	68.973	72.317	73.822	76.589	*82.208	*84.803	*89.853	*80.780
64 Pixel	60.100	63.730	68.537	79.514	73.737	75.514	*83.189	*85.548	*88.000	*80.363
51 Pixel	58.479	56.494	66.127	75.205	72.934	76.803	*80.139	78.934	*85.367	70.548
38 Pixel	62.710	64.232	69.274	71.722	72.811	76.255	73.475	77.224	79.066	78.564
26 Pixel	45.985	66.394	69.413	76.494	73.730	72.888	72.417	78.124	79.514	76.618
13 Pixel	42.680	58.116	56.386	63.954	67.591	73.042	60.015	75.907	73.537	76.526

Table 11 : Average accuracy of all classes from a feedforward neural network having 25 hidden neurons with DB-TH Data set using a spectrogram window of 128 Pixel

Size of the y-axis of spectrogram	Size of the x-axis of spectrogram									
	1 (Pixel)	2 (Pixel)	3 (Pixel)	4 (Pixel)	5 (Pixel)	6 (Pixel)	7 (Pixel)	8 (Pixel)	9 (Pixel)	10 (Pixel)
64 Pixel	45.997	63.363	65.215	71.553	67.002	47.652	64.585	77.294	79.908	72.690
58 Pixel	44.913	64.158	68.177	61.438	64.768	71.737	65.163	76.210	72.210	66.332
51 Pixel	44.624	64.926	72.854	76.762	64.053	67.961	70.069	66.706	74.693	71.606
45 Pixel	40.348	50.259	57.445	71.856	75.199	56.519	77.773	65.268	71.146	79.711
38 Pixel	45.576	58.555	68.716	62.463	54.686	59.619	63.074	74.207	75.540	78.516
32 Pixel	38.752	54.798	60.079	66.837	61.911	71.665	50.213	71.179	71.921	61.544
26 Pixel	37.038	53.274	69.793	69.530	62.975	57.432	71.396	67.087	67.146	63.041
19 Pixel	41.452	47.829	61.878	65.366	73.892	66.883	67.429	59.271	57.412	64.624
13 Pixel	33.596	28.388	59.777	61.951	70.312	58.719	77.353	69.780	67.744	55.107
6 Pixel	34.588	50.246	62.837	61.924	47.888	64.887	42.371	60.604	57.438	67.691

Table 12 : Average accuracy of all classes from a feedforward neural network having 25 hidden neurons with DB-TH-ENG Data set using a spectrogram window of 128 Pixel

Size of the y-axis of spectrogram	Size of the x-axis of spectrogram									
	1 (Pixel)	2 (Pixel)	3 (Pixel)	4 (Pixel)	5 (Pixel)	6 (Pixel)	7 (Pixel)	8 (Pixel)	9 (Pixel)	10 (Pixel)
64 Pixel	43.907	54.100	63.166	*80.471	*83.931	*85.197	*85.707	*85.699	*85.483	*85.486
58 Pixel	47.158	48.958	72.355	71.243	*83.398	*83.089	*83.367	*82.548	*84.263	*83.004
51 Pixel	49.405	62.208	74.425	79.205	*82.780	*83.598	*84.247	*85.560	*84.139	*84.660
45 Pixel	50.687	59.876	62.911	*84.139	79.405	79.838	73.066	*86.301	*85.699	*84.950
38 Pixel	48.610	66.903	63.490	71.452	76.803	78.564	*81.266	*85.934	*85.301	*84.683
32 Pixel	47.521	56.625	70.703	63.722	77.429	78.178	68.595	*88.703	*80.625	*83.656
26 Pixel	43.467	64.255	68.741	65.622	61.537	72.178	71.421	72.127	71.413	73.900
19 Pixel	45.699	51.097	63.042	63.158	62.718	70.633	71.537	74.456	76.958	72.927
13 Pixel	45.382	43.359	61.459	65.645	63.050	70.564	71.575	72.347	72.116	72.069
6 Pixel	32.247	40.247	51.822	65.722	64.857	70.293	71.992	71.042	72.332	72.100

4.3 การใช้เทคนิค Image Scaling ในการลดมิติของข้อมูลมีผลกับประสิทธิภาพ ในการรู้จำหรือไม่

ในส่วนนี้ได้ทดลองกับข้อมูลชุดที่ 1 และ 2 ที่แสดงในตารางที่ 2 และ 4 โดยกำหนดตัวแปรควบคุมคือ ใช้สเปกโตรแกรมที่ใช้จะสร้างจากวินโดว์(Windows Length) ขนาด 1024 512 256 128 จุด และมีการทับซ้อนกันของสัญาณเสียง 25% ของขนาดวินโดว์(Windows Length)

จากนั้นเราใช้เทคนิค Image Scaling คือ DCT-based compressed algorithms มาทำการลดขนาดของสเปกโตรแกรมให้มีขนาดลดลงและเท่ากัน ในทุกๆ คำร้องของข้อมูลชุดที่ 1 และ 2 โดยหลักการลดขนาดนั้นจะลดเป็นวินโดว์(Windows Length)ไป โดยจะทำการลดทุก 10% ในทุกวินโดว์(Windows Length) ไปดังต่อไปนี้

1. สเปกโตรแกรมที่สร้างจากวินโดว์(Windows Length) ขนาด 1024 จะทำการลดขนาดสำหรับใช้ในการทดลองลงเหลือ 512 , 461, 410, 358, 307, 256, 205,154, 102 และ 51 จุด.
2. สเปกโตรแกรมที่สร้างจากวินโดว์(Windows Length) ขนาด 512 จะทำการลดขนาดสำหรับใช้ในการทดลองลงเหลือ 256, 230, 205, 179, 154, 128, 102, 77, 51 และ 26 จุด.
3. สเปกโตรแกรมที่สร้างจากวินโดว์(Windows Length) ขนาด 256 จะทำการลดขนาดสำหรับใช้ในการทดลองลงเหลือ 128,115, 102, 90, 77, 64, 51, 38, 26 และ 13 จุด.
4. สเปกโตรแกรมที่สร้างจากวินโดว์(Windows Length) ขนาด 128 จะทำการลดขนาดสำหรับใช้ในการทดลองลงเหลือ 64, 58, 51, 45, 38, 32, 26, 19, 13 และ 6 จุด.

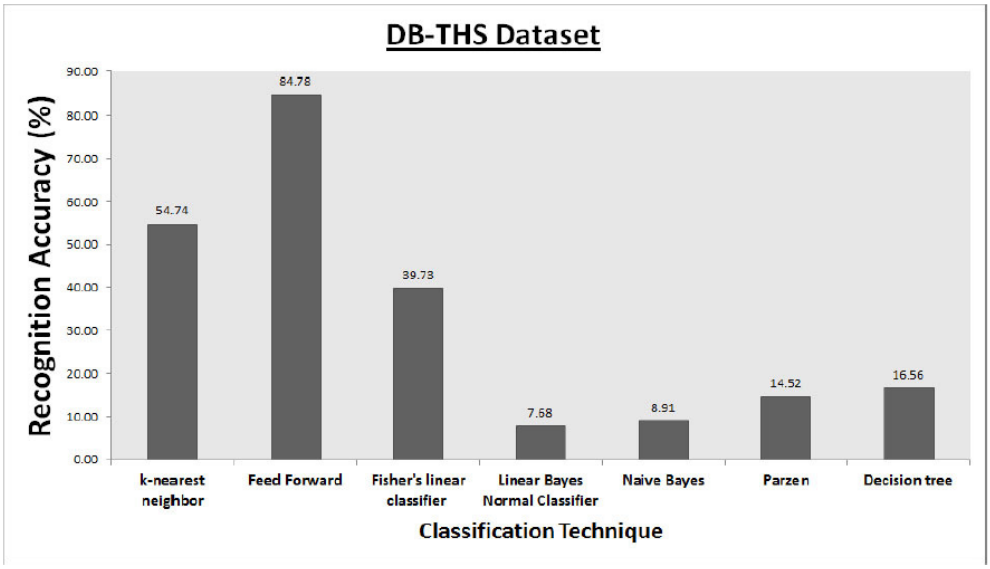
จากตารางที่ 5 ถึง 12 เป็นตารางแสดงข้อมูลโดยเราจะเห็นว่าการใช้เทคนิค Image Scaling คือ DCT-based compressed algorithms มาทำการลดขนาดของสเปกโตรแกรมให้มีขนาดลดลงหรือที่เรียกว่าการลดขนาดมิติข้อมูล สามารถใช้งานได้ดี โดยสามารถลดขนาดของสเปกโตรแกรม ได้มากกว่า 50% จากขนาดวินโดว์(Windows Length) เริ่มต้นปกติ แต่ยังคงให้ประสิทธิภาพในการรู้จำที่สูงกว่า 85% อยู่ในทุกๆ ขนาดวินโดว์(Windows Length) ซึ่งเมื่อนำไปใช้งานจริงแล้วการใช้เทคนิคการลดมิติข้อมูลสามารถประหยัดหน่วยความจำได้มาก

ดังนั้นจึงสรุปได้ว่าใช้เทคนิค Image Scaling แบบ DCT-based compressed algorithms ขนาดมาทำการลดขนาดสเปกโตรแกรมในทุกๆ วินโดว์(Windows Length) สามารถใช้งานได้และยังสามารถลดขนาดได้มากกว่า 50% โดยที่ประสิทธิภาพในการู้จำยังคงสูงกว่า 85%

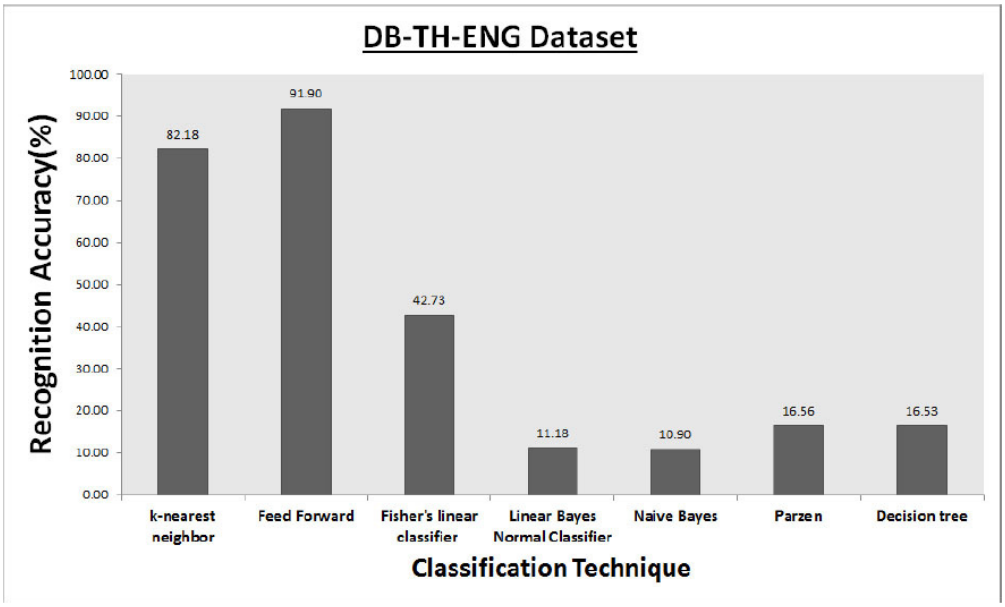
4.4 เปรียบเทียบประสิทธิภาพในการู้จำกับ Classification technique. ตัวอื่น ๆ

ในงานวิจัยนี้ได้ใช้โครงข่ายประสาทเทียม (Artificial Neural Networks) แบบ three-layer feed-forward network สำหรับการรู้จำเสียงร้องหรือคำร้องในข้อมูลทดลองชุดที่ 1 และ 2 แต่ในความเป็นจริงยังคงมีเทคนิคอื่นๆ ในการรู้จำข้อมูลอีกหลายๆ ชนิดด้วยกัน ดังนั้นงานวิจัยนี้จึงจำเป็นต้องทำการทดลองเพื่อเปรียบเทียบประสิทธิภาพในการรู้จำกับ Classification technique. ตัวอื่นๆ โดยงานวิจัยนี้เลือกมาทั้งหมด 6 ชนิดเพื่อทำการเปรียบเทียบประกอบไปด้วย

- K-nearest neighbor (KNN)
- Fisher's linear classifier
- Linear Bayes Normal Classifier
- Naive Bayes Classifier
- Parzen Classifier
- Decision tree



ภาพที่ 9 ข้อมูลประสิทธิภาพในการรู้จำของแต่ละเทคนิคบนข้อมูลการทดลองชุดที่ 1



ภาพที่ 10 ข้อมูลประสิทธิภาพในการรู้จำของแต่ละเทคนิคบนข้อมูลการทดลองชุดที่ 2

จากภาพที่ 9 และ 10 ซึ่งแสดงข้อมูลประสิทธิภาพในการรู้จำของแต่ละเทคนิคบนข้อมูลการทดลองชุดที่ 1 และ 2 นั้นจะเห็นว่าการใช้โครงข่ายประสาทเทียม (Artificial Neural Networks) แบบ three-layer feed-forward network สามารถให้ประสิทธิภาพในการรู้จำสูงกว่าทุกๆ เทคนิค และเทคนิค K-nearest neighbor (KNN) ให้ประสิทธิภาพในการู้จำเป็นลำดับถัดมา

4.5 เปรียบเทียบประสิทธิภาพในการู้จำกับ Automatic Speech Recognition (ASR). **ตัวอื่น ๆ**

การรู้จำเสียงพูด (Speech Recognition) คือ การที่คอมพิวเตอร์สามารถรับรู้ เสียงของมนุษย์ได้โดยอัตโนมัติ โดยทั่วไปแล้วจะอาศัยระบบโปรแกรมคอมพิวเตอร์ที่สามารถแปลงเสียงพูด (Audio File) เป็นข้อความตัวอักษร (Text) โดยสามารถแจกแจงคำพูดต่างๆ ที่มนุษย์สามารถพูดใส่ไมโครโฟน โทรศัพท์หรืออุปกรณ์อื่นๆ และเข้าใจคำศัพท์ทุกคำอย่างถูกต้องเกือบ 100%

ซึ่งเทคนิคการรู้จำเสียงพูด (Speech Recognition) เป็นเทคนิคแรกๆ ที่ถูกนำมาใช้แก้ปัญหาการรู้จำเสียงร้องหรือคำร้องในเพลง ดังนั้นเราจึงทดลองเปรียบเทียบประสิทธิภาพของการรู้จำเสียงพูด (Speech Recognition) กับ เทคนิคการแก้ปัญหาของงานวิจัยนี้ สำหรับการทดลองนั้น เราเลือกใช้ Hidden Markov Model (HMM) สำหรับเป็นอัลกอลิทึมในการรู้จำข้อมูล จากนั้นใช้ LPC และ MFCC 13 coeficients. สำหรับเป็นตัวสกัดข้อมูล

จากข้อมูลที่แสดงในภาพที่ 11 นั้นเราจะเห็นว่าการใช้โครงข่ายประสาทเทียม (Artificial Neural Networks) แบบ three-layer feed-forward network ร่วมกับสเปกโตรแกรม สามารถให้ประสิทธิภาพในการรู้จำสูงกว่าทุกๆ เทคนิค ของเทคนิคการรู้จำเสียงพูด (Speech Recognition) ทั้ง 2 ชุดข้อมูล

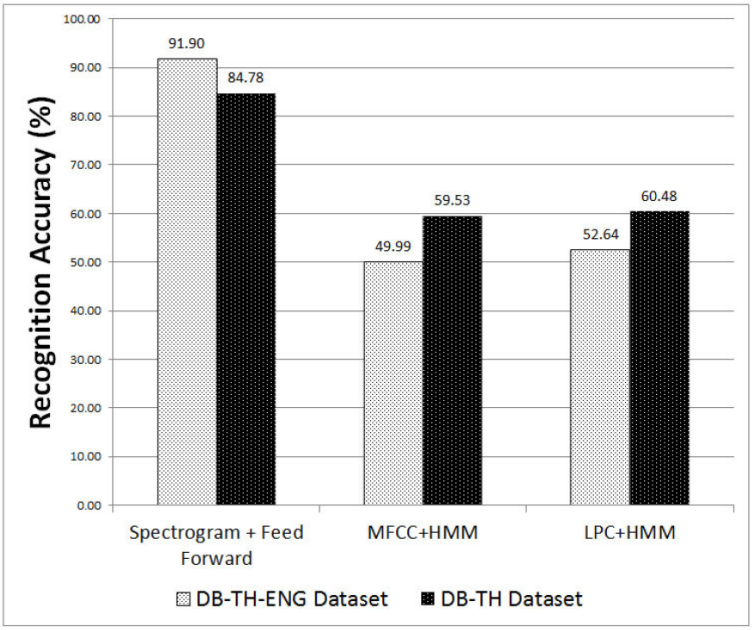


Figure 11: ข้อมูลประสิทธิภาพในการรู้จำของแต่ละเทคนิคบนข้อมูลการทดลองชุดที่ 1 และ 2.

6. บทวิจารณ์

ปัญหาในการรู้จำคำร้อง(Singing voice recognition) จากเพลงนั้นมีความยุ่งยากและซับซ้อนมาก เนื่องจากคำที่ออกเสียงจากการร้องเพลงนั้นมีคุณสมบัติที่แตกต่างจากการพูดโดยทั่วๆ ไปหลายประการ เช่น รูปแบบการออกเสียงเฉพาะตัวบุคคลที่ขึ้นกับประเภทของดนตรี ระยะเวลาในการออกเสียงคำที่แตกต่างกันจากปัจจัยการเอื้อนหรือการลากเสียง ลูกคอในการร้องเพลง และจังหวะของเพลงต่างๆ จึงทำให้อัลกอลิทึมเดิมที่ใช้กับปัญหาการรู้จำเสียงพูดนั้นไม่ประสบผลสำเร็จเท่าที่ควร

อีกปัจจัยหนึ่งที่ลดทอนคุณภาพของการรู้จำคือเพลงจะมีเสียงดนตรีประกอบจะมีผลเทียบได้กับเสียงรบกวน(Noise) ซึ่งจะทำให้ประสิทธิภาพในการรู้ลดลง โดยทั่วไปแล้วการแก้ปัญหาคือการใช้ตัวรองเสียงรบกวน (Noise filter) ชนิดต่างๆ เข้ามากรองเสียงดนตรีออกไป แต่การใช้ตัวกรองเสียงรบกวนนั้นจะเป็นการเพิ่มขั้นตอนเข้าไปทำให้เวลาในการประมวลผลเพิ่มขึ้นตาม อีกทั้งการใช้ตัวรองเสียงรบกวนอาจไปทำลายเสียงที่ต้องการใช้จริงไปด้วย

ในงานวิจัยชิ้นนี้อาศัยหลักการของการรู้จำรูปภาพเข้ามาประยุกต์ใช้ในการแก้ปัญหา อย่างแรกนำเอาสัญาณเสียงที่ได้ไปแปลงให้อยู่ในรูปแบบของภาพที่เราเรียกว่า สเปกโตรแกรม (Spectrogram) ที่มีลักษณะข้อมูลเป็นแบบ MxN มิติ แต่เนื่องจาก สเปกโตรแกรม(Spectrogram) มีขนาดของมิติข้อมูลที่สูงเกินไปการนำเอามาใช้งานโดยตรงนั้นเป็นไปได้ยากและต้องใช้เครื่องประมวลผลที่ประสิทธิภาพสูงมาก อีกทั้งคำร้องแต่ละคำเมื่อนำมาแปลงเป็น สเปกโตรแกรม (Spectrogram) ขนาดจะไม่เท่ากันไม่สามารถนำไปทำการรู้จำได้ ดังนั้นงานวิจัยนี้จึงนำเอา สเปกโตรแกรม(Spectrogram) มาลดขนาดของมิติข้อมูลลงก่อนนำไปทำการรู้จำด้วยเทคนิคการปรับขนาดภาพ(Image resizing algorithm) สำหรับขั้นตอนในการรู้จำนั้นงานวิจัยนี้เลือกใช้ Feed-Forward neural Network

จากการทดลองพบว่างงานวิจัยนี้สามารถแก้ปัญหาการรู้จำเสียงร้องหรือคำร้องในเพลงที่มีเสียงดนตรีพื้นหลังได้เป็นอย่างดี โดยประสิทธิภาพในการรู้จำอยู่ที่ 90.0% ขึ้นไปในทุก ๆ ชุดทดลอง อีกทั้งยังสามารถรู้จำได้หลายๆ ภาษาพร้อมๆ กันในชุดข้อมูลเดียวกันอีกด้วย

7. หนังสืออ้างอิง

1. Ajmera, J., McCowan, I., Bourlard, H., May 2003. Speech/music segmentation using entropy and dynamism features in a hmm classification framework. Speech Commun. 40, 351{363.URL http://portal.acm.org/citation.cfm?id=781675.781682
2. Berenzweig, A., Ellis, D., 2001. Locating singing voice segments within music signals. In: Applications of Signal Processing to Audio and Acoustics, 2001 IEEE Workshop on the. pp. 119 {122.
3. Berenzweig, Adam L.; Ellis, D. P. W. L. S., 6 2002. Using voice segments to improve artist classi_cation of music. In: Audio Engineering Society Conference: 22nd International Conference: Virtual, Synthetic, and Entertainment Audio.
4. Chou, W., Gu, L., 2001. Robust singing detection in speech/music discriminator design. In: Proceedings of the Acoustics, Speech, and Signal Processing, 200. on IEEE International Conference - Volume 02. IEEE Computer Society, Washington, DC, USA, pp. 865-868.
5. Cullity, B. D., 2003. Music information retrieval. Vol. 35. Information Today Books.
6. Dugad, R., Ahuja, N., 2001. A fast scheme for image size change in the compressed domain. IEEE Trans. Circuits Syst. Video Techn. 11 (4), 461-474.
7. Esmaili, S., Krishnan, S., Raahemifar, K., may 2004. Content based audio classification and retrieval using joint time-frequency analysis. In: Acoustics, Speech, and Signal Processing, 2004. Proceedings. (ICASSP '04).IEEE International Conference on. Vol. 5. pp. 665.
8. Fujihara, H., Goto, M., Ogata, J., Komatani, K., Ogata, T., Okuno, H. G.,2006. Automatic synchronization between lyrics and music cd recordingsbased on viterbi alignment of segregated vocal signals. In: Proceedings of the Eighth IEEE International Symposium on Multimedia. ISM '06. IEEE Computer Society, Washington, DC, USA, pp. 257-264.
9. Gerhard, D. B., 2003. Computationally measurable difierences between speech and song. Ph.D. thesis, Burnaby, BC, Canada, Canada, aAINQ81587.
10. Gruhne, M., Schmidt, K., Dittmar, C., Sep 23-27 2007. Phoneme recognition in pop-pular music. In: 8th International Conference on Music Information Retrieval. Vienna, Austria, pp. 290-294.

11. Hayashi, T., Ishii, N., Yamaguchi, M., Sept 2014. Fast music information retrieval with indirect matching. In: Signal Processing Conference (EUSIPCO), 2014 Proceedings of the 22nd European. pp. 1567-1571.
12. Hu, Y., Liu, G., Jun. 2014. Singer identi_cation based on computational auditory scene analysis and missing feature methods. J. Intell. Inf. Syst 42 (3), 333{352.
13. URL http://dx.doi.org/10.1007/s10844-013-0271-6
14. Huang, P.-S., Chen, S., Smaragdis, P., Hasegawa-Johnson, M., March 2012. Singing-voice separation from monaural recordings using robust principal component analysis. In: Acoustics, Speech and Signal Processing(ICASSP), 2012 IEEE International Conference on. pp. 57-60.
15. Kan, M.-Y., Wang, Y., Iskandar, D., Nwe, T. L., Shenoy, A., 2008. Lyrically: Automatic synchronization of textual lyrics to acoustic music signals. IEEE Transactions on Audio, Speech & Language Processing 16 (2), 338-349.
16. Kim, Y. E., 2002. Singer identification in popular music recordings using voice coding features. In: In Proceedings of the 3rd International Conference on Music Information Retrieval. pp. 164-169.
17. Lin, C.-C., Chen, S.-H., Truong, T.-K., Chang, Y., sept. 2005. Audio classification and categorization based on wavelets and support vector machine.Speech and Audio Processing, IEEE Transactions on 13 (5), 644-651.
18. M. Gruhne, K. S., Dittmar, C., Sep 23-27 2007. Phoneme recognition in pop-pular music. In: 8th International Conference on Music Information Retrieval. Vienna, Austria., pp. 2027-2030.
19. Maddage, N., Wan, K., Xu, C., Wang, Y., june 2004. Singing voice detection using twice-iterated composite fourier transform. In: Multimedia and Expo, 2004. ICME '04. 2004 IEEE International Conference on. Vol. 2. pp.1347 -1350 Vol.2.
20. Maddage, N. C., Xu, C., Wang, Y., 2003. An svm-based classification approach to musical audio. In: ISMIR.
21. Makeyev, O., Sazonov, E., Schuckers, S., Lopez-Meyer, P., Melanson, E., Neuman, M., aug. 2007a. Limited receptive area neural classifier for recognition of swallowing sounds using continuous wavelet transform. In: Engineering in Medicine and Biology Society, 2007. EMBS 2007. 29th Annual International Conference of the IEEE. pp. 3128-3131.

22. Makeyev, O., Sazonov, E., Schuckers, S., Melanson, E., Neuman, M., aug.2007b. Limited receptive area neural classi_er for recognition of swallowing sounds using short-time fourier transform. In: Neural Networks, 2007.IJCNN 2007. International Joint Conference on. pp. 1601 -1606.
23. McVicar, M., Santos-Rodriguez, R., Ni, Y., Bie, T. D., Feb 2014. Automatic chord estimation from audio: A review of the state of the art. Audio, Speech, and Language Processing, IEEE/ACM Transactions on 22 (2),556-575.
24. Mesaros, A., Virtanen, T., march 2010. Recognition of phonemes and words in singing. In: Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on. pp. 2146 -2149.
25. Nwe, T. L., Shenoy, A., Wang, Y., 2004. Singing voice detection in popular music. In: Proceedings of the 12th annual ACM international conference on Multimedia. MULTIMEDIA '04. ACM, New York, NY, USA, pp. 324-327.
26. Raj, B., 2007. Separating a foreground singer from background music.
27. Rocamora, M., Herrera, P., sep 2007. Comparing audio descriptors for singing voice detection in music audio _les. In: Brazilian Symposium on Computer Music, 11th. San Pablo, Brazil.
28. Ryynanen, M., Virtanen, T., Paulus, J., Klapuri, A., June 2008. Accompaniment separation and karaoke application based on automatic melody transcription. In: Multimedia and Expo, 2008 IEEE International Conference on. pp. 1417-1420.
29. Sasou, A., Goto, M., Hayamizu, S., Tanaka, K., 18-23, 2005a. An autoregressive, non-stationary excited signal parameter estimation method and an evaluation of a singing-voice recognition. In: Acoustics, Speech, and Signal Processing, 2005. Proceedings. (ICASSP '05). IEEE International Conference on. Vol. 1. pp. 237 - 240.
30. Sasou, A., Goto, M., Hayamizu, S., Tanaka, K., 18-23, 2005b. An autoregressive, non-stationary excited signal parameter estimation method and an evaluation of a singing-voice recognition. In: Acoustics, Speech, and Signal Processing, 2005. Proceedings. (ICASSP '05). IEEE International Conference on. Vol. 1. pp. 237 - 240.
31. Shenoy, A., 2005. Singing voice detection for karaoke application. Proceedings of SPIE 5960, 752-762. URL http://link.aip.org/link/PSISDG/v5960/i1/p596028/s1Agg=doi

32. Su, L., Yeh, C.-C., Liu, J.-Y., Wang, J.-C., Yang, Y.-H., Aug 2014. A systematic evaluation of the bag-of-frames representation for music information retrieval. Multimedia, IEEE Transactions on 16 (5), 1188-1200.
33. Suzuki, M., Hosoya, T., Ito, A., Makino, S., January 2007. Music information retrieval from a singing voice using lyrics and melody information. EURASIP J. Appl. Signal Process. 2007, 151-151. URL http://dx.doi.org/10.1155/2007/38727
34. Toyoda, Y., Huang, J., Ding, S., Liu, Y., sept. 2004a. Environmental sound recognition by multilayered neural networks. In: Computer and Information Technology, 2004. CIT '04. The Fourth International Conference on.pp. 123 - 127.
35. Toyoda, Y., Huang, J., Ding, S., Liu, Y., 2004b. Environmental sound recognition by the instantaneous spectrum combined with the time pattern of power, 169-172.
36. Tsai, W.-H., Wang, H.-M., Rodgers, D., Cheng, S.-S., Yu, H.-M., 2003. Blind clustering of popular music recordings based on singer voice characteristics. In: ISMIR.
37. Tzanetakis, G., june 2004. Song-speci_c bootstrapping of singing voice structure. In: Multimedia and Expo, 2004. ICME '04. 2004 IEEE International Conference on. Vol. 3. pp. 2027 - 2030 Vol.3.
38. Vaizman, Y., McFee, B., Lanckriet, G., Oct 2014. Codebook-based audio feature representation for music information retrieval. Audio, Speech, and Language Processing, IEEE/ACM Transactions on 22 (10), 1483-1493.
39. Wang, J.-C., Lee, H.-P., Wang, J.-F., Lin, C.-B., jan. 2008. Robust environmental sound recognition for home automation. Automation Science and Engineering, IEEE Transactions on 5 (1), 25 -31.
40. Wong, C., Szeto, W., Wong, K., Mar. 2007. Automatic lyrics alignment for Cantonese popular music. Multimedia Systems 12 (4/5), 307-323.
41. Yaguchi, Y., Oka, R., 2005. Song wave retrieval based on frame-wise phoneme recognition. In: Lee, G., Yamada, A., Meng, H., Myaeng, S. (Eds.), Information Retrieval Technology. Vol. 3689 of Lecture Notes in Computer Science. Springer Berlin / Heidelberg, pp. 503-509.
42. Yoshii, K., Goto, M., Okuno, H. G., jan. 2007. Drum sound recognition for polyphonic audio signals by adaptation and matching of spectrogram templates with harmonic structure suppression. Audio, Speech, and Language Processing, IEEE Transactions on 15 (1), 333 -345.

43. Zwan, P., Szczuko, P., Kostek, B., Czyzewski, A., 2008. Transactions on rough sets ix. Springer-Verlag, Berlin, Heidelberg, Ch. Automatic Singing Voice Recognition Employing Neural Networks and Rough Sets, pp. 455-473.

Output จากโครงการวิจัยที่ได้รับทุนจาก สกว.

1. ผลงานตีพิมพ์ในวารสารวิชาการนานาชาติ (ระบุชื่อผู้แต่ง ชื่อเรื่อง ชื่อวารสาร ปี เล่มที่ เลขที่ และหน้า) หรือผลงานตามที่คาดไว้ในสัญญาโครงการ
2. การนำผลงานวิจัยไปใช้ประโยชน์
   - เชิงพาณิชย์ (มีการนำไปผลิต/ขาย/ก่อให้เกิดรายได้ หรือมีการนำไปประยุกต์ใช้โดยภาคธุรกิจ/บุคคลทั่วไป)
   - เชิงนโยบาย (มีการกำหนดนโยบายอิงงานวิจัย/เกิดมาตรการใหม่/เปลี่ยนแปลงระเบียบข้อบังคับหรือวิธีทำงาน)
   - เชิงสาธารณะ (มีเครือข่ายความร่วมมือ/สร้างกระแสความสนใจในวงกว้าง)
   - เชิงวิชาการ (มีการพัฒนาการเรียนการสอน/สร้างนักวิจัยใหม่)
3. อื่นๆ (เช่น ผลงานตีพิมพ์ในวารสารวิชาการในประเทศ การเสนอผลงานในที่ประชุมวิชาการ หนังสือ การจดสิทธิบัตร)

ภาคผนวก

ประกอบด้วย reprint หรือ manuscript

Elsevier Editorial System(tm) for Information Sciences

Manuscript Draft

Manuscript Number: INS-D-16-417

Title: Single Signal Entity Approach for Thai Singing Word Recognition Using Images of Power Spectrogram and Image Processing Techniques

Article Type: SI:Life Sci. Data Analysis

Keywords:
Spectrogram, Singing voice recognition, Automatic speech recognition (ASR), Feed-Forward Neural Network , Image Scaling Algorithm

Corresponding Author: Dr. Peerapol Khunarsa,

Corresponding Author's Institution: Chulalongkorn University

First Author: Peerapol Khunarsa

Order of Authors: Peerapol Khunarsa

Single Signal Entity Approach for Singing Word Recognition Using Images of Power Spectrogram and Image Processing Techniques

Peerapol Khunarsa[a], Chidchanok Lursinsap[b]

[a]*Faculty of Science and Technology, Uttaradit Rajabhat University, Uttaradit, Thailand. (email: peerapol_utt@hotmail.com).*
[b]*Advanced Virtual and Intelligent Computing Center (AVIC), Department of Mathematics Chulalongkorn University, Bangkok 10330, Thailand.(email: lchidcha@chula.ac.th).*

Abstract

Singing word recognition is one of the interesting research topics in the area of Music Information Retrieval (MIR). The first approach to solve this problem used successful techniques in Automatic Speech Recognition (ASR). Moving from monophonic to polyphonic audio signal, the problem has become more complex. The background instrumental accompaniment is regarded as the noise source degrading the performance of the recognition system. The papers proposed a statistical learning method for recognition of the word in a singing signal with background music and for classification of singing voice region in a polyphonic audio signal.

The goal of this paper is to solve singing word recognition without using any method to separated instrumental from background music . The papers also applied the concept of image recognition by using a spectrogram feature as an image to solve the problem. An audio signal that accompanies music was analyzed and transformed into a spectrogram feature. A dimension of spectrogram feature is very high and time interval of each singing word is not equal. Then we apply image resizing algorithm to solve both problem. To recognize it, the whole spectrogram feature was sliced and formed as a feature vector for a neural classifier.

Several classification functions are compared, such as Fisher classifier, K-nearest neighbor and Feed-Forward can effectively recognize the word in music with the accuracy rate more than 90.0% Especially, we can recognize Cross-Language Music Data.

Keywords: Spectrogram, Singing voice recognition, Automatic speech recognition (ASR), Feed-Forward Neural Network.

1. Introduction

The problems of recognizing a singing word under some noisy background has been an interesting topics. In this paper, we are interested in the problem of Music Information Retrieval (MIR) Cullity (2003) Hayashi et al. (2014) Vaizman et al. (2014) McVicar et al. (2014) Su et al. (2014) , which is a particular application of recognizing words under a mixture of several music instruments. The difficulty of recognition lies at the types of instruments and their strength. The background instrumental accompaniment is regarded as the noise source degrading the performance of the recognition system.

During a singing period, the power of a singing voice may be stronger or weaker than the power of the music instruments. If the power of a singing voice is stronger than the musical background, then the recognition is rather simple. On the contrary, it is quite complex when the power of a singing voice is rather weak. Besides these two factors, the spectrum of each instrument is unknown a priori for a given song. The spectrum can be varied according to duration of voice sound, loudness, pitch, vibrato, formant, rhythm and rhyme Gerhard (2003) Sasou et al. (2005a) Yaguchi and Oka (2005) Gruhne et al. (2007). So many methods based on the features extracted directly from the accompanied vocal segments are difficult to achieve good performance when accompaniment is stronger or singing voice is weaker.

This makes the background filtering process more complicated in terms of computational cost. There have been several proposed techniques which may be relevant to the problem of recognizing singing words with complex musical background. Some interesting techniques are the following.

Hu and Liu (2014) exploited computational auditory scene analysis (CASA) to segregate singing voice units for each time frame. Those segregated singing voice units were regarded as reliable components. And then two missing feature methods were used respectively together with those reliable components to perform the tasks of singer identication. The reconstruction method was exploited to obtain a complete singing spectrum which were further used to extract the features for singer identication, and the marginalization method was exploited to directly perform the identication task based solely on reliable components.

Raj (2007) applied Probabilistic Latent Component Decomposition (PLCD) for separating singing voices from background music in popular songs. The set of basis vectors described by the frequency marginal were learned for each component signal from a separated unmixed training recording (vocal or background music). The spectrograms for the voice-only and music-only components of the mixed recordings were obtained using PLCD.

Huang et al. (2012) proposed using robust principal component analysis (RPCA) for singing-voice separation from music accompaniment. By RPCA, they obtained two output matrices, one is the sparse matrix containing formant structures which indicates vocal activity, and another is low-rank matrix which indicates musical notes. It was based on the assumption that repetition is a core principle in music and the singing voice has more variation and is relatively sparse within a song.

Ryynanen et al. (2008).used fundamental frequency (F0) to separated accompaniment from polyphonic music based on automatic melody transcription. This method used sinusoidal modeling to estimate, synthesize, and remove the lead vocals. In their system, the pitches of singing also needed to be estimated in advance.

Several techniques concerning to solve the problem of audio recognition Makeyev et al. (2007b)Lin et al. (2005)Esmaili et al. (2004) Wang et al. (2008)Yoshii et al. (2007)Toyoda et al. (2004b) Makeyev et al. (2007a)Ajmera et al. (2003)Toyoda et al. (2004a). Most of the proposed methods consisted of two processing steps: feature extraction and classification. In the first step, feature exaction, the redundant information contained in the signal were transformed into descriptors used as the input of a classifier for recognition in the second step. Shenoy (2005) used the amplitude variation over time in each sub-band and a threshold method on the energy function such as the proportion of frames classified as vocals to be equivalent to the proportion of the singing in the entire song. Nwe et al. (2004) used Harmonic Attenuated LFPCs with Hidden Markov Model HMM models based on three parameters, e.g. section type (intro, verse, chorus, bridge and outro), tempo, and loudness. Tsai Tsai et al. (2003) used Mel-frequency cepstral coefficients (MFCCs) and GMM models to classify vocal from non-vocal signals. Berenzweig and Ellis (2001) used vector of posterior probability as a feature and HMM framework with two states, "singing" and "non-singing". Chou and Gu (2001) used 4 Hz modulation energy, harmonic coefficient, 4Hz harmonic coefficient, delta MFCC and delta log energy as features and GMM model to detect singing voice. Berenzweig (2002) applied 13 PLPCs and MLP.

Maddage et al. (2003) considered LPC, LPC derived cepstrums (LPCC), MFCC, spectral power (SP), short time energy (STE), and ZCR as features and a multi-layer neural network, a SVM, and a GMM for classification. SVM was found to outperform the other classifiers. Maddage Maddage et al. (2004) latter tried Twice Iterated Composite Fourier Transform (TICFT) to each audio frame. Rocamora and Herrera (2007) used different sets of features such as MFCCs with their deltas, LFPC with their deltas and double deltas, PLPCs with their deltas, HC and pitch and different classifiers such as a SVM, a back propagation NN, a decision tree classifier, and two different K-nearest neighbors. Tzanetakis (2004) used spectral shape feature, MFCCs, mean and deviation of pitch , centroid and LPCs for feature extraction and a naive bayes network, nearest neighbor algorithms, back-propagation ANN, a decision tree classifier based on the C4.5 algorithm, a SVM classifiers. Kim (2002) used a harmonic measure, defined as the ratio of the total signal energy to the maximally harmonically attenuated signal and threshold method on the harmonic measure to classify the segment.

As compared to other areas in audio such as speech or music, research on general unstructured audio-based scene recognition has received little attention. To the best of our knowledge, only a few systems (and frameworks) have been proposed to investigate of singing voice recognition with raw audio. Most of investigations of singing voice recognition deal with recognition phoneme first and used a speech recognizer for lyrics recognition. Sasou et al. (2005b) tested an Auto Regressive HMM with pure singing voice signals from the RWC database. These studies presumed pure monophonic singing voices without accompaniment, posing additional difficulties for practicable use with musical audio signals like CD recordings. Suzuki et al. (2007) combined both the melody and the lyrics of the user's singing voice to retrieve a song from a database. They also used a large vocabulary speech recognition system with a HMM as the acoustic model adopted to the singing voice using the speaker adaptation technology.

Wong et al. (2007) proposed a system for real-time alignment of Cantonese music, which is a particular tone language. The meaning of a word changes when pronounced with a different pitch. A MLP was used to segregate the vocal from the non-vocal segments taking as input the spectral flux, the HC, the ZCR, the MFCCs, the amplitude level and the 4Hz modulation energy. DTW algorithm was used to align the two sequences. However, this method is not consistently effective because the durations of uttered phonemes depend on locations, even though they are the same phonemes.

Kan et al. (2008) was probably the first English lyrics sentence level alignment system for aligning the lyrics to the music signals for a specific structure of songs. M. Gruhne and Dittmar (2007) implemented a system that performed automatic classification of 15 voiced sung phonemes in polyphonic audio. Their procedure was based on harmonics extraction and re-synthesis of a number of partials as a preprocessing step, in order to reduce influences from accompanying sounds. Then, low-level features were extracted from the audio and classified using different classification techniques like SVM, GMM and MLP. Fujihara et al. (2006) performed automatic synchronization between lyrics and polyphonic music signals for Japan CD recordings. Their proposed system included detection of vocal segments, segregation of vocals and adaptation of a speech recognizer to the segregated vocal signals. During the first step, harmonics extraction and re-synthesis was performed as in M. Gruhne and Dittmar (2007). A simple HMM was used in order to keep only the vocal regions and remove the non-vocal sections. Last, features were extracted from the audio (MFCCs, delta MFCCs, and delta power) and the Viterbi algorithm was used to align the segmented vocal parts with the corresponding lyrics. Zwan et al. (2008) presented an automatic singing voice recognition using neural network and rough sets. The method also required and combined many type of feature vector for classification method. Mesaros and Virtanen (2010) studied the use of n-gram language models in recognizing phonemes and words in monophonic and polyphonic music. They considered uni-, bi-, and tri-gram language models for phonemes and bi- and tri-grams for words. In the recognition, a hidden Markov model based phonetic recognizer was adapted to singing voice. Their word recognition system achieved only 24% correct recognition rate.

In this paper, we are interested singing voice recognition in polyphonic recordings of popular music. Our hypothesis is that, for any song, it is unnecessary to filter the instrumental background from the singing voice to recognize the singing words. Since the complexity of musical background in terms of relevant factors as previously mentioned is too high and uncontrollable, it would be better not to eliminate the musical background from the singing voice. Our objectives concern two essential issues. The first issue is the recognition speed. Without filtering the musical background from singing voice, the processing time is expected to be tremendously reduced. The second issue emphasizes on the independence of the following factors: duration of voice sound, loudness, pitch, vibrato, formant, rhythm and rhyme. These two issues lead to the problem of which representation domain is the most

suitable for any song so that the highest recognition accuracy of singing words can be achieved from this representation. In our algorithm, we transformed the problem of recognizing one dimensional signal of song into the problem of recognizing a color image. The features of image are extracted and classified. The details will be discussed in the following sections.

The rest of the paper is organized as follows. Section 2 formulates our studied problem and constraints. Section 3 reviews related backgrounds. Section 4 discusses the concept of our proposed algorithm. Section 5 explains the experimental set-up. Section 6 evaluates the results. Section 7 concludes the paper. Each song The musical background By following this direction, we expect to achieve high recognition accuracy.

2. Problem Formulation and Constraints

We considered the following situation. Given a song as a mixture of musical background and singing voice and recognize them. There are two procedures involved in this situation. The first procedure concerns the problem of time interval of each singing word in each song is different and it depends on depending on the singer and rhythm. Then, how to make time interval of each singing words are of the same. Audio music with instrumental interference: In polyphonic music recordings, the instrumental interference is treated as the noise source that causes degradation to the intelligibility of the singing voice signal. The second problem is how to recognize the singing word in music with instrumental. The solutions to these two problems are independent from each other. In this paper, we concentrate on the both procedure. Hence, it is assumed that the input to our algorithm an audio signal was already contains a singing word. The input is in the form of a set of sampled audio signal values in time domain, i.e. $\{x(1), \ldots, x(n)\}$. Our study is constrained by the following factors and conditions.

Constraints

1. Our system took polyphonic music audio signal as the input sampled from music CD recording.
2. Different music genres were included in experiment such as Rock ,hard rock, Soft Rock , Dance , Hip-Pop, Soul, R&B ,folk and Acoustic from different artists.
3. All music genres have man and woman singers.

4. Singing words can be either Thai or English. Only frequently occurred and composed words, phases, and sentences in most of the sampled songs were considered. Table 4 summarizes the frequently used Thai words and their duration. Table 6 summarizes the frequently used Thai and English words, phases and sentences with their duration.

The problems discussed in this paper are the following. Let $\mathbf{S} = \{x(1), \ldots, x(n)\}$ be a given interval of sampled signals of a song. Each $x(i)$ may be a mixture of singing voice with musical background or singing voice alone.

1. How to recognize the singing word in interval $\mathbf{S}$ without eliminating the musical background ?
2. What are the essential features to the recognition rate achieving high accuracy ?
3. Can the recognizing algorithm be robust to the previously mentioned constraints ?

3. Proposed Concept

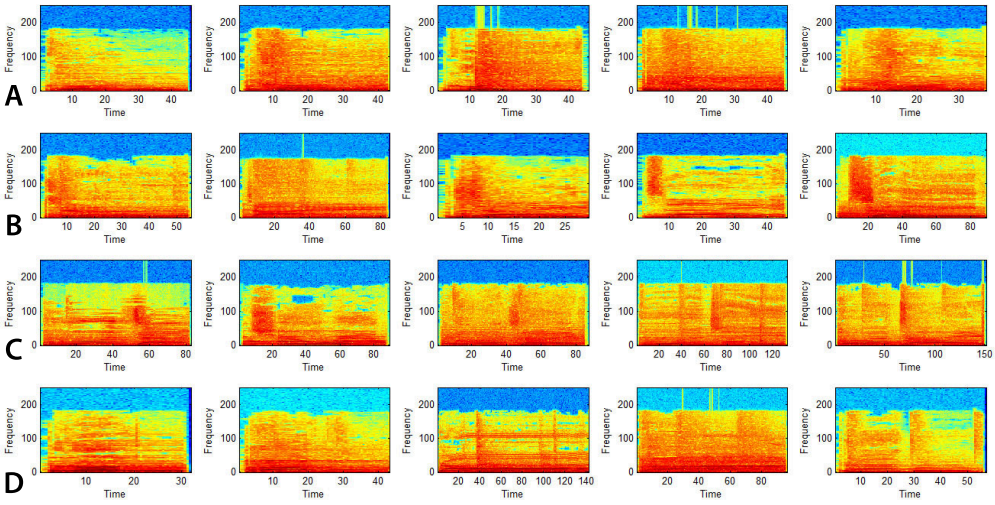


Figure 1: Examples of four singing words represented in forms of spectrograms. (A) Word 1. (B) Word 2. (C) Word 3. (D) Word 4.

Recognizing a singing word is more complex than recognizing a spoken word without any musical background. The strength and clearness of a

signing word is always deteriorated by several factors such as signing styles, duration of sing voice, instrumental background signal under uncontrollable loudness, pitch, vibrato, formant, and rhythm. To effectively eliminating the musical background, the types of musical instruments must be known in advance to properly filter the corresponding musical signal frequencies from the signing word signal. In fact, these frequencies are unknown a priori to the filtering process. If the musical background cannot be completely separated from the signing signal, then the percentage of recognition accuracy is obviously not high. Furthermore, the unpredictable singing duration can make the recognition process rather complicated in terms of time complexity.

Our solution is based on the following observation and hypothesis. The hypothesis is that for a singing word, there are various ways to sing the word with different backgrounds. But if we plot the spectrograms feature of all different intervals of songs having this words, then we should have similar spectrograms feature. Figure 1 shows some examples of spectrograms feature of the same words. There are four words, named A, B, C, and D, and their spectrograms feature are in rows 1 (top row), 2, 3, and 4 (bottom), respectively. These four words were sung by different persons with different musical backgrounds and duration. Observe that the spectrograms feature of any singing word are similar to each other but different from the spectrograms feature of the other singing words. Note that each spectrogram feature was derived from the mixture of singing word and musical background. Therefore, it is unnecessary to filter any background from the signing word prior to the recognition. A spectrogram feature can be considered as a color image. In our approach, the problem of recognizing a singing word with musical background is transformed into the problem of recognizing a spectrogram feature. Our recognizing algorithm consists of the following steps.

1. Transform the input audio signals $\mathbf{S}$ into a spectrogram feature.
2. Extract the features to represent the spectrogram feature.
3. Eliminate some less informative pixels from the spectrogram feature to reduce the number of features.
4. Classify the features.

The results from our proposed technique will be compared with automatic speech Recognition (ASR) algorithm. The detail of each step is given in the following section.

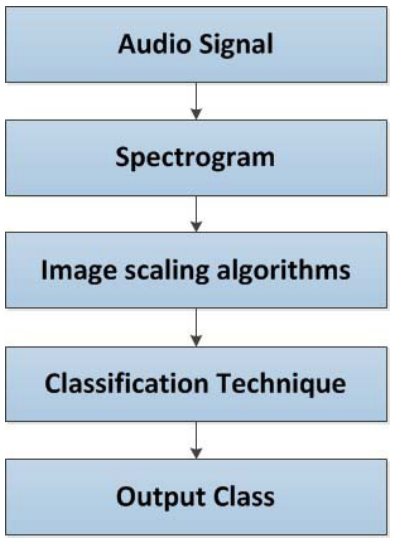


Figure 2: Our proposed algorithms

3.1. Spectrogram Feature Representation

A spectrogram feature is a visual representation of the distribution of acoustic energy across frequencies in time domain. The horizontal axis of a spectrogram feature typically represents the time intervals of audio signal snapshots. The vertical axis represents the power spectrum of discrete frequency steps. The strength of power detected is represented as the intensity at each time-frequency pixel.

First, the input audio signal $x(n)$ of each singing word is sliced into a number of small windows or frames whose size is equal to a power of two. Each signal window is calculated by using the short-time Fourier transform (STFT) defined as follows.

$$X(k) = \sum_{n=0}^{N-1} w(n)x(n)exp(-\frac{2\pi kn}{N}) \tag{1}$$

for $k = 0, 1, \ldots, N-1$, where k corresponds to the frequency $f(k) = (\frac{kf_s}{N})$; f_s is the sampling frequency in Hertz; and $w(n)$ is Hamming time-window given by

$$w(n) = 0.54 - 0.46\cos\left(\frac{\pi n}{N}\right) \tag{2}$$

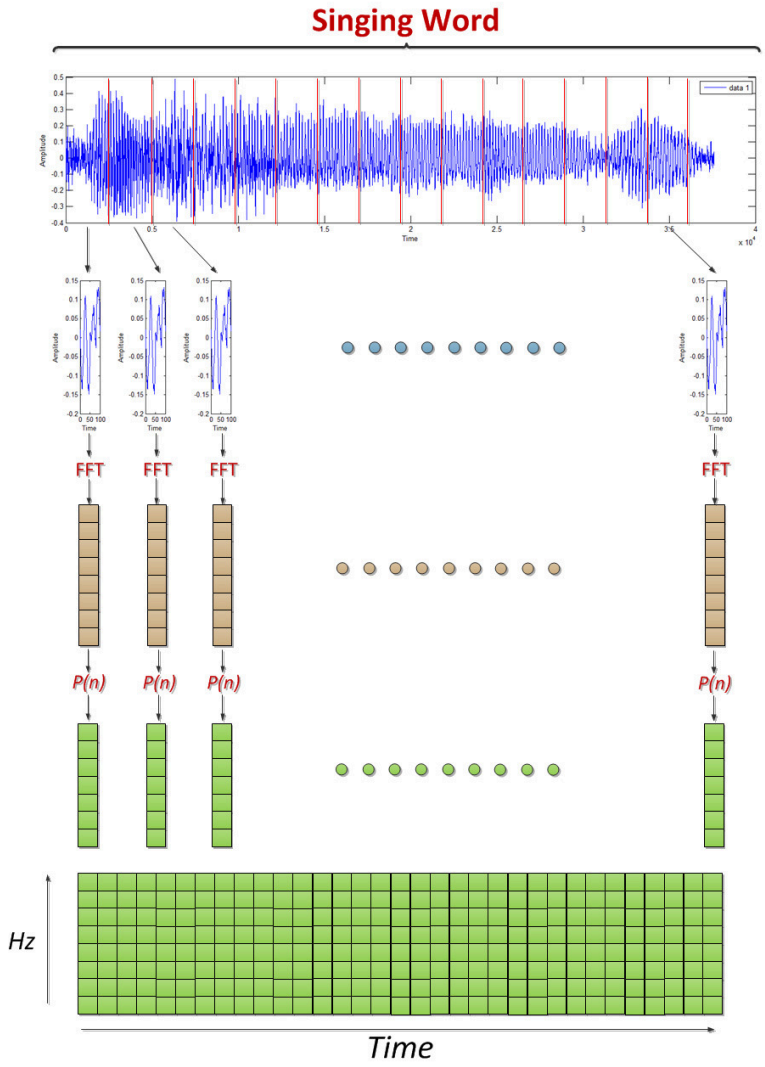


Figure 3: The process of computing the power spectrum of an input audio signal and forming the spectrogram feature in our algorithm.

The power of each $X(k)$, denoted by $P(k)$, is computed by the following equation.

$$P(k) = 10log_{10}(X(k)) \tag{3}$$

Each $P(k)$ and its time interval are plotted to form a spectrogram feature of each singing word. Figure 3 shows an example of how to create a spectrogram feature. This spectrogram feature is used as the features of the song and used in the classifying process. In this paper, we used a neural network as a classifier whose input must be in the form of a vector. A power spectrogram feature can be viewed as a collection of columns of power spectrums. Therefore, the spectrogram feature can be transformed into a vector by concatenating these columns of power spectrums as shown in Figure 4.

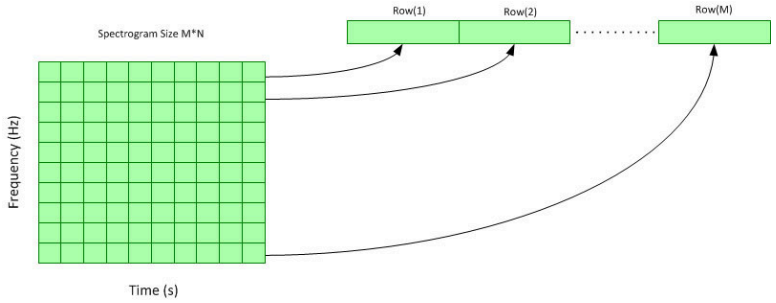


Figure 4: Forming the input vector of neural classifier by concatenating columns of power spectrums.

3.2. Feature Reduction with Image Scaling Algorithm

To speed up the classification, the less informative features must be eliminated. Note that the size of power spectrogram feature of each song may not be equal due to the length of each song and the sampling rate.

The y-axis or vertical axis represents the frequency of the spectrogram feature. The size will depend on the size of window that is used to created a spectrogram feature. As showing in figure 3, an input audio signal $x(n)$ of each singing word was cut in to small windows. Then, a spectrogram feature can be obtained from different sizes of windowed segment and size of windowed segment can be equal a power of two. Figure 5 show a spectrogram feature obtained from different sizes of windowed segment. From the figure we can see a different characteristic of a spectrogram feature obtained from different sizes of windowed segment.

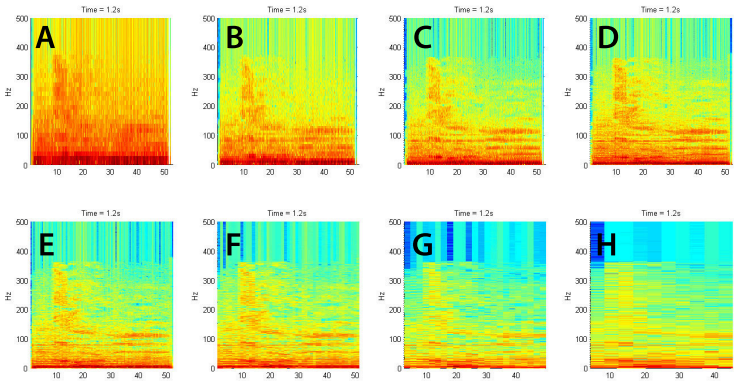


Figure 5: Example of spectrogram feature obtained from different sizes of windowed segment a) 64, b) 128, c) 256, d) 512, e) 1024, f) 2048, g) 4096, h) 8192.

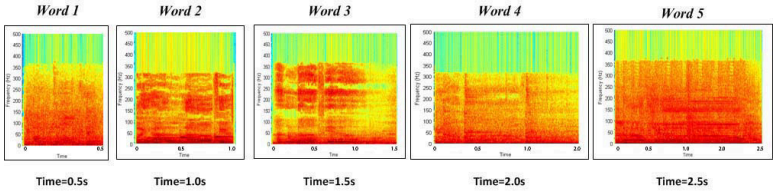


Figure 6: Example a size of spectrogram feature column obtained from different period of time .

The x-axis or horizontal axis represents the time of the spectrogram feature. The interval of each singing word in each song is different and it depending on the singer and rhythm. When convert each singing to a spectrogram feature the size of specture feaure in horizontal axis was not equal. Figure 6 show a spectrogram feature that created from an audio sound with a different period of time. The period of time of each audio sound are 0.5s, 1.0s, 1.5s, 2.0s and 2.5s. From this figure, a spectrogram feature that created from an audio sound with a different period of time a size of column of each spectrogram feature was not equal.

Depending on the length of each singing word (x-axis) and size of window that is used to created a spectrogram feature (y-axis). The size of spectrogram feature of each singing word is not the same.

Figure 7 shows an example feature reduction and size normalization of five different words. As showing in figure 7, we apply image resizing method for resized spectrogram feature. The first reason is to reduce the size of the spectrogram feature. The second reason is to make the data become equal in all singing word. Several efficient image scaling techniques such as nearest neighbour sampling, bilinear interpolation, bicubic interpolation, and discrete cosine transform-based compression, can be adapted to this problem. A brief summary of each technique is given in the followings.

Image scaling is the process of resizing a digital image, wherein an image is converted from one resolution/dimension to another resolution/dimension without losing the visual content. It has many terminologies in literature such as Image Interpolation, image re-sampling, digital zooming, image magnification or enhancement, etc .

3.2.1. Nearest Neighbor Interpolation

Nearest neighbor interpolation guesses each pixel as having the same visual quality as its closest neighbor. It is one of the fastest and simplest forms of interpolation technique. During enlarging (upscaling), the empty spaces will be replaced with the nearest neighboring pixel. Shrinking, on the other hand involves reduction of pixels.

3.2.2. Bilinear Interpolation

Bilinear interpolation is slightly more sophisticated and calculates each new pixel as a linear weighted sum of the 4 closest neighboring pixels. It is an extension of linear interpolation for interpolating functions of two variables (x and y) on a regular 2D grid. This algorithm is a combination of two

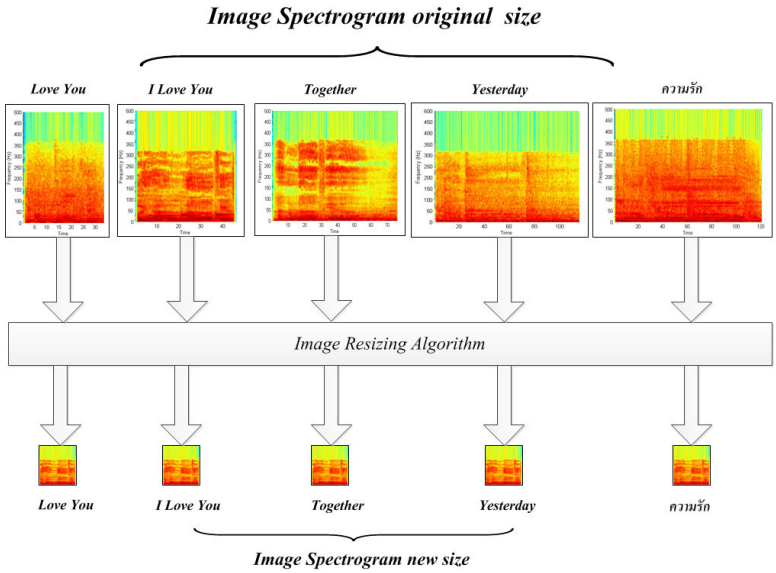


Figure 7: An example of feature reduction and normalization of spectrogram size. There are five different words in this example.

linear interpolations. The idea is to perform linear interpolation first in one direction, and then again in the other direction.

3.2.3. Bicubic Interpolation

The bicubic interpolation is advancement over the cubic interpolation in two dimensional regulargrid. The interpolated surface is smoother thancorresponding surfaces obtained by above mentionedmethods bilinear interpolation and nearest-neighbourinterpolation. It uses polynomials, cubic, or cubic convolution algorithm. The Cubic ConvolutionInterpolation determines the grey level value from theweighted average of the 16 closest pixels to thespecified input coordinates, and assigns that value tothe output coordinates, the first four one-dimension.For Bicubic Interpolation (cubic convolutioninterpolation in two dimensions), the number of gridpoints needed to evaluate the interpolation function is 16, two grid points on either side of the point underconsideration for both horizontal and perpendiculardirection.

3.2.4. Discrete Cosine Transform (DCT)-based compressed

Dugad and Ahuja (2001) have proposed an elegant scheme for changing the image sizes in the Discrete Cosine Transform(DCT) space. The algorithm

while halving an image, from a 8x8 DCT block, 4x4 block in the spatial domain is obtained. This is carried out by applying a 4-point inverse DCT (IDCT) on the 4x4 lower frequency-terms. In the next stage, this image (in the spatial domain) is once again compressed by 8x8 block DCT encoding (JPEG standard). For doubling the images, first the DCT encoded image is transformed to its spatial domain. Then for each 4x4 block, the DCT coefficients are computed applying a 4-point DCT. These 4x4 DCT coefficients are directly used as the low frequency components of 8x8 blocks, which are subsequently converted to a 8X8 block in spatial domain by applying a 8-point inverse DCT (IDCT).

The four techniques described above, we compare the values after halving and doubling a spectrogram feature using Peak Signal to Noise Ratio (PSNR) measure.

Table 1: PSNR values after halving and doubling a spectrogram feature that create from Table 4.

	PSNR(dB)			
SingingWord	Nearest	Bilinear	Bicubic	DCT
”คน”	34.748	35.505	37.406	39.233
”ความ”	34.823	35.540	37.483	39.390
”เคย”	34.510	35.349	37.219	39.065
”ใคร”	34.471	35.301	37.188	38.992
”ใจ”	34.635	35.455	37.334	39.110
”ฉัน”	33.890	35.106	36.748	38.340
”ที่”	34.551	35.571	37.336	39.116
”เธอ”	34.186	35.199	36.885	38.604
”มี”	34.458	35.307	37.120	38.896
”รัก”	34.207	35.279	36.948	38.617
”รู้”	33.912	35.095	36.720	38.304
”เรา”	34.750	35.315	37.385	39.357

Table 2: PSNR values after halving and doubling a spectrogram feature that create from Table 6.

	PSNR(dB)			
SingingWord	Nearest	Bilinear	Bicubic	DCT
I love you	30.374	32.695	33.374	33.993
Love you	31.240	33.353	34.198	34.945
Together	30.509	32.801	33.496	34.001
Tomorrow	30.383	32.603	33.325	33.941
Yesterday	29.978	32.461	33.018	33.476
”ความรัก”	32.020	33.935	35.036	36.094
”คิดถึง”	32.444	34.383	35.613	36.713
”ใครสักคน”	31.003	33.242	34.071	34.827
”ไม่เคย”	32.015	34.148	35.153	36.039
”ไม่มี”	29.454	31.953	32.478	32.859
”รักเธอ”	32.874	34.582	35.873	37.069
”หัวใจ”	32.920	34.469	35.810	37.185

The PSNR is most commonly used as a measure of quality of reconstruction of lossy compression codecs. The signal in this case is the original data, and the noise is the error introduced by compression. Typical values for the PSNR in lossy image and video compression are between 30 and 50 dB, where higher is better **? ? ?** .

The PSNR values obtained after halving and doubling of spectrogram feature are shown in the Table 1 and 2 . For spectrogram feature DCT-based compressed algorithms perform better than another algorithm in most cases and value of PSNR was over 30 dB. Therefore , we chooses to DCT-based compressed algorithms reduce a spectrogram feature size for image scaling algorithm in this research.

4. DATA COLLECTION

Our system take polyphonic music audio signal as input, which are sampled from music CD recording and different music genres are included in experiment such as Pop Rock ,Hard rock, Soft Rock , Dance , Hip-Pop, Soul, R&B ,folk and Acoustic. The files are all from different artists. We investigated the performance of a spectrogram feature of audio features to solve the problem of Singing Voice Recognition and provide an empirical evaluation on two data set.

The first data set, denoted as DB-THS, was a collection of songs randomly chosen from Thai popular music CDs. It contains over 1500 Album. All detail was showing in table 3. The DB-THS data set consists of a 12 Thai One syllable singing word, 7200 sound samples and 600 for each word. The 12 considered singing word were showing in table 4.

The second database, denoted as DB-TH-ENG, was a collection of songs randomly chosen from English and Thai popular music CDs. For the second data set that consists of a greater than equal to two syllables singing word. It contains over 1600 Album. All detail was showing in table 5. The DB-TH-ENG data set was consists 12 singing word. We used 5 word in English and 7 word in Thai. DB-TH-ENG that contains 7200 sound samples and 600 for each word. The 12 considered singing word were showing in table 6. All singing words in table 4 and 6 was selected from the most frequently in the all song.

All each singing word audio was selected and cut by manual by using Sony Sound Forge program. All Sample files in Table4 and 6 was coded in stereo of frequency 44.2 kHz with 128/s bit rate.

Table 3: The music used in DB-THS DATASET.

Music Genres	Male Singer	Female Singer	Total
Pop Rock	1,768	1,545	3,313
hard rock	978	667	1,645
soft rock	2,284	2,100	4,384
dance	1,177	467	1,644
hip-pop	304	160	464
soul	250	108	358
R&B	1,135	652	1,787
folk	297	162	459
Acoustic	1,288	982	2,270
	Total		16324

Table 4: DATABASES DB-THS USED IN EXPERIMENTS

Class.	Singing word	Time duration(min-max)	Pronounce (in Thai)
1	”คน”	0.65s-2.95s	”kon”
2	”ความ”	0.26s-0.60s	”kwarm”
3	”เคย”	0.33s-0.62s	”koey”
4	”ใคร”	0.33s-0.70s	”krai”
5	”ใจ”	0.44s-1.38s	”jai”
6	”ฉัน”	0.26s-1.23s	”chan”
7	”ที่”	0.26s-0.54s	”tee”
8	”เธอ”	0.23s-0.78s	”ther”
9	”มี”	0.28s-0.86s	”mai”
10	”รัก”	0.18s-1.48s	”luck”
11	”รู้”	0.28s-0.47s	”roo”
12	”เรา”	0.26s-0.73s	”raw”

Table 5: The music used in DB-THS-ENG DATASET.

Music Genres	Male Singer	Female Singer	Total
Pop Rock	1,105	1,432	2,537
hard rock	1,734	503	2,237
soft rock	4,473	1,466	5,939
dance	1,121	964	2,085
hip-pop	162	149	311
soul	208	358	566
R&B	1,155	840	1,995
folk	329	355	684
Acoustic	462	940	1,402
	Total		17756

Table 6: DATABASES DB-TH-ENG USED IN EXPERIMENTS

Class.	Singing word	Time duration(min-max)	Pronounce (in Thai)
1	I love you	0.65s-2.95s	
2	Love you	0.57s-2.92s	
3	Together	1.04s-2.11s	
4	Tomorrow	1.07s-6.63s	
5	Yesterday	0.81s-5.90s	
6	”ความรัก”	0.52s-3.65s	”kwarm-luck”
7	”คิดถึง”	0.88s-1.11s	”kit-thun”
8	”ใครสักคน”	0.99s-4.62s	”krai-sak-kon”
9	”ไม่เคย”	0.41s-1.99s	”mai-koey ”
10	”ไม่มี”	0.57s-1.17s	”mai-mee”
11	”รักเธอ”	0.47s-1.93s	”luck-ther”
12	”หัวใจ”	0.73s-1.46s	”hua-jai”

5. EXPERIMENTAL EVALUATION

This section discusses the methodology used in our proposed techniques. It includes the description of the experiment setup, the comparative study method and the implementation details. All calculations were done using Matlab 2012a on a Intel Dual Core E6750 2.66 GHz Desktop machine with 6G of RAM.

5.1. Experimental Setup

Our recognizing algorithm perform the following proposed steps in section 3. , all audio signals were converted to mono and down-sampling types at rate of 11,000 Hz. The experiment consists of tests on DB-THS and DB-TH-ENG data set. Each singing word in DB-THS and DB-TH-ENG dataset are randomly divided into four groups of equal sizes. Then, arbitrarily selected three groups are used for training and the rest is used for testing. For cross-validation procedure, the same process is repeated 50 times with the different training and test sets, to ensure that all samples are included at least once in the test set. The mean recognition rate was calculated based on the error average for one run on test set.

In this paper, a three-layer feed-forward network was used for sound classifying into correct type of singing words as showing in Table 4 and 6. A sigmoid transfer function was used in hidden layer and output layer. The network will be train with scaled conjugate gradient backpropagation function. The network consists of 12 outputs corresponding to 12 classes in each data sets. The value of each output is between $[0, 1]$. The number of hidden neurons was adjusted to achieve the high accuracy. Although selecting a good learning rule can generate a good result, in this paper, we will not concern the learning rules since the learning rules are not the focus of this study.

5.2. Selected number of hidden neurons

The number of hidden neurons is also another relevant factor affecting the accuracy. However, theoretically estimating this number is rather difficult. To select parameter in number of hidden neurons criterion. We use a rectangular window of 512 pixels with a 25% overlap This corresponds to the window size used for all spectrogram feature. After that, all spectrogram of each singing words was resized to 128×5 pixel with DCT-based compressed algorithms. (We will consider effect of different window size of spectrogram

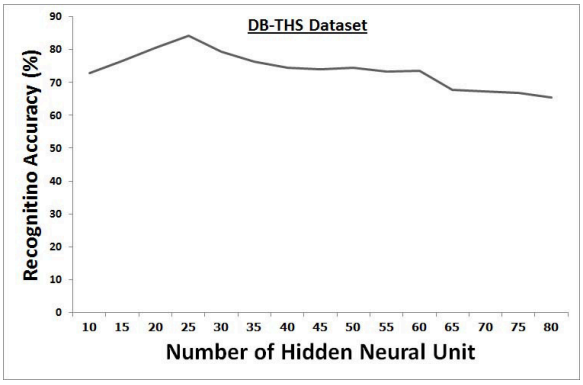


Figure 8: Overall recognition accuracy using Feed Forward Neural Network with varying number of Hidden Neural Unit on DB-THS Dataset

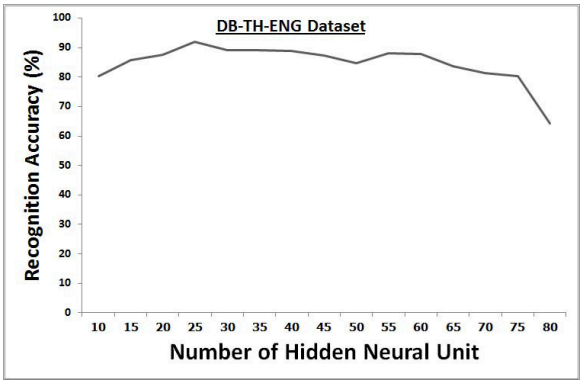


Figure 9: Overall recognition accuracy using Feed Forward Neural Network with varying number of Hidden Neural Unit on DB-TH-ENG Dataset

feature and size of spectrogram feature after resized in the next section). We examine the results from verity number of hidden neural unit and using the same for each environment type.

We plot the classification performance in Fig 8 and 9 highest recognition rate was obtained using 25 hidden neural units, with an average accuracy of 84.16% for data in Table 4 and 91.904% for data in Table 6. Thus, we chased 25 hidden neural units for three-layer feed-forward network in our experiments and used for sound classifying in all experiment.

5.3. Experimental Results

The problem of reducing the size of spectrogram feature is suitable size of spectrogram feature to be used. Then, the size of spectrogram feature considering both x-axis and y-axis. In this experiment, the size of the spectrogram considering both the x-axis and y-axis.

This Experimented, we used the data from Table 4 and Table 6. Base on our experimental setup, we use a window of 1024, 512, 256, 128 pixels with a 25% overlap for y-axis. The number of frequency components in the y-axis spectrogram feature output is equal to half window length. To determine the suitable size of the y-axis, we reduced the size of the y-axis by reducing the percentage from 100 to 10 by 10 percent each of every windows size.

- By using a window of 1024 to experimented , we reduced the size of the y-axis of spectrogram feature to 512 , 461, 410, 358, 307, 256, 205, 154, 102 and 51 Pixel.
- By using a window of 512 to experimented , we reduced the size of the y-axis of spectrogram feature to 256, 230, 205, 179, 154, 128, 102, 77, 51 and 26 Pixel.
- By using a window of 256 to experimented , we reduced the size of the y-axis of spectrogram feature to 128,115, 102, 90, 77, 64, 51, 38, 26 and 13 Pixel.
- By using a window of 128 to experimented , we reduced the size of the y-axis of spectrogram feature to 64, 58, 51, 45, 38, 32, 26, 19, 13 and 6 Pixel.

The x-axis is consider reducing the number of columns to suitable size of spectrogram feature. The number of columns of spectrogram consider on

average value of spectrogram that created from data in table 4 and 6. The average of spectrogram in x-axis is 10. Therefore, we use the average size of spectrogram as the maximum amount of data in the x-axis. The number of columns in the x-axis, we expanded the size of the x-axis of spectrogram feature to 1, 2, 3, 4, 5, 6, 7, 8, 9 and 10 Pixel. The overall recognition rate by use a window of 1024, 512, 256, 128 pixels are given in table 7, 8, 9, 10, 11, 12, 13 and 14.

By using windows of 1024, Performance on DB-TH-ENG data set higher than DB-THS data set. A recognition accuracy in DB-THS data set shows in table 7 that 60 out of 100 is higher than 80%. But 88 of 100 from DB-TH-ENG data set in Table 8 was show classification rate more than than 80%.

When reduced a window size to 512, 256 and 128 ,efficiency of recognition began to decrease. Efficiency of recognition was show in table 9 and 10 was a spectrogram that used a window of 512. In DB-THS data set 32 out of 100 is higher than 80% and 66 of 100 from DB-TH-ENG data set was show classification rate more than than 80% .

By using a window size 256, Efficiency of recognition was show in table 11 and 12 for DB-THS and DB-TH-ENG data set. In DB-THS data set 10 out of 100 is higher than 80% and 29 of 100 from DB-TH-ENG data set was show classification rate more than 80%.

By using a window size 128 was show the lowest performance. Efficiency of recognition was show in table 13 and 14 for DB-THS and DB-TH-ENG data set. Any size of spectorgram feature in DB-THS dataset was show classification rate more than 80% in Table 13 and 29 of 100 from DB-TH-ENG data set was show classification rate more than 80% in Table 14.

As showing in table 7, 8, 9, 10, 11, 12, 13 and 14. a spectrogram that created from a large size of windows segment gives better classification accuracy than a spectrogram that created from a small size of windows segment.

5.4. Compare with the other classification technique.

The following classification techniques are used for speech/speaker recognition or have, in the past, been used for this paper. They are:

- K nearest neighbor (KNN)
- Fisher's linear classifier
- Linear Bayes Normal Classifier

Table 7: Average accuracy of all classes from a feedforward neural network having 25 hidden neurons with DB-TH Data set using a spectrogram window of 1024 Pixel

Size of the y-axis of spectrogram	Size of the x-axis of spectrogram									
	1 (Pixel)	2 (Pixel)	3 (Pixel)	4 (Pixel)	5 (Pixel)	6 (Pixel)	7 (Pixel)	8 (Pixel)	9 (Pixel)	10 (Pixel)
512 Pixel	52.374	70.276	*85.865	*88.374	*88.276	*87.606	*87.095	*87.951	*87.419	*87.036
461 Pixel	61.918	79.080	*83.941	*82.711	*82.227	*81.287	*83.527	*84.447	*85.649	*87.468
410 Pixel	62.791	72.020	*81.396	*82.430	*84.591	*83.534	*81.327	*85.938	*85.163	*84.790
358 Pixel	63.159	76.706	*80.801	*81.923	*81.652	*82.463	*84.407	*84.814	*83.034	*84.604
307 Pixel	60.138	70.115	73.491	*81.023	*82.218	*84.670	*83.163	*83.161	*83.080	*84.552
256 Pixel	58.548	75.343	*81.118	*84.782	*85.034	*85.622	*86.910	*84.205	*85.844	*84.926
205 Pixel	60.591	71.616	72.657	70.489	*80.034	*80.736	*82.693	*82.836	*81.990	*84.580
154 Pixel	49.714	73.438	74.069	72.466	66.483	*80.023	*80.502	*83.547	*84.087	*83.453
102 Pixel	54.069	63.514	*82.483	66.522	78.785	70.056	*80.286	63.593	62.529	77.051
51 Pixel	47.626	67.225	67.823	68.250	64.775	74.640	76.460	77.458	64.217	64.906

Table 8: Average accuracy of all classes from a feedforward neural network having 25 hidden neurons with DB-TH-ENG Data set using a spectrogram window of 1024 Pixel

Size of the y-axis of spectrogram	Size of the x-axis of spectrogram									
	1 (Pixel)	2 (Pixel)	3 (Pixel)	4 (Pixel)	5 (Pixel)	6 (Pixel)	7 (Pixel)	8 (Pixel)	9 (Pixel)	10 (Pixel)
512 Pixel	67.730	*88.710	*90.054	*94.062	*94.919	*94.394	*94.378	*94.788	*94.942	*94.865
461 Pixel	76.162	*84.687	*84.672	*86.425	*88.371	*93.320	*93.212	*94.440	*94.556	*94.672
410 Pixel	73.869	*84.170	*90.834	*92.811	*92.602	*93.266	*94.479	*94.934	*94.792	*94.726
358 Pixel	75.093	*86.139	*87.579	*92.564	*88.263	*93.614	*93.340	*94.734	*94.865	*94.212
307 Pixel	74.255	*84.313	*87.247	*87.151	*90.626	*92.718	*93.517	*94.776	*94.023	*94.224
256 Pixel	74.236	*83.066	*87.807	*91.904	*91.857	*92.564	*92.305	*94.961	*94.479	*94.409
205 Pixel	69.815	*81.093	*85.259	*89.544	*90.166	*90.888	*90.336	*91.873	*92.869	*92.278
154 Pixel	65.568	*83.112	*84.143	*86.413	*89.884	*89.251	*90.830	*91.409	*92.174	*92.938
102 Pixel	69.745	76.077	*89.027	*88.104	*88.475	*88.490	*89.398	*89.510	*89.220	*89.023
51 Pixel	58.382	76.637	*83.224	*85.185	*86.606	*87.216	*87.896	*87.266	*87.475	*88.637

Table 9: Average accuracy of all classes from a feedforward neural network having 25 hidden neurons with DB-TH Data set using a spectrogram window of 512 Pixel

Size of the y-axis of spectrogram	Size of the x-axis of spectrogram									
	1 (Pixel)	2 (Pixel)	3 (Pixel)	4 (Pixel)	5 (Pixel)	6 (Pixel)	7 (Pixel)	8 (Pixel)	9 (Pixel)	10 (Pixel)
256 Pixel	50.858	70.585	75.490	*80.651	*81.062	*81.368	*81.283	*81.007	*81.332	*82.337
230 Pixel	50.229	70.407	78.242	78.719	79.986	*80.742	*80.588	*80.796	*81.545	**82.111
205 Pixel	49.250	72.203	76.007	77.209	79.339	79.986	*80.690	*80.347	*81.234	*82.861
179 Pixel	48.690	71.192	73.463	77.700	77.002	79.112	79.906	*80.114	*80.164	*81.003
154 Pixel	53.392	62.224	76.590	79.742	79.329	*80.973	*80.568	*81.748	*84.711	*85.341
128 Pixel	46.217	71.141	78.163	76.763	75.322	*80.328	*80.878	*81.885	*83.199	*83.802
102 Pixel	50.574	67.197	73.644	76.343	76.821	76.571	*81.468	76.767	*82.240	*82.536
77 Pixel	43.628	62.732	73.399	77.178	76.551	76.087	76.033	77.615	77.633	79.062
51 Pixel	46.725	64.578	67.725	69.666	72.526	75.507	73.295	77.039	77.215	77.621
26 Pixel	41.048	62.873	64.191	67.246	69.663	73.030	73.332	74.264	75.795	77.758

Table 10: Average accuracy of all classes from a feedforward neural network having 25 hidden neurons with DB-TH-ENG Data set using a spectrogram window of 512 Pixel

Size of the y-axis of spectrogram	Size of the x-axis of spectrogram									
	1 (Pixel)	2 (Pixel)	3 (Pixel)	4 (Pixel)	5 (Pixel)	6 (Pixel)	7 (Pixel)	8 (Pixel)	9 (Pixel)	10 (Pixel)
256 Pixel	59.152	73.542	*81.299	*81.286	*87.489	*87.405	*88.831	*88.339	*87.248	*89.124
230 Pixel	60.068	74.856	*80.699	*81.202	*83.744	*86.548	*87.260	*87.730	*89.535	*89.439
205 Pixel	64.315	74.974	*80.927	*81.713	*83.417	*86.599	*86.375	*87.709	*87.165	*88.160
179 Pixel	62.906	69.645	*80.722	*82.351	*83.120	*86.191	*86.382	*87.264	*87.339	*87.855
154 Pixel	60.816	68.411	71.009	*81.977	*82.949	*85.879	*85.795	*87.362	*86.099	*87.018
128 Pixel	62.636	73.745	78.260	*81.662	*82.366	*84.705	*85.013	*86.826	*86.124	*87.750
102 Pixel	62.876	66.404	72.636	*81.640	*82.876	*83.976	*85.209	*85.131	*85.867	*86.880
77 Pixel	62.391	69.804	77.398	79.197	79.967	*80.000	*84.801	*85.580	*85.248	*86.037
51 Pixel	54.916	63.721	73.781	74.072	*80.425	*82.055	*82.398	*82.039	*82.964	*83.507
26 Pixel	47.557	59.210	65.170	77.574	72.120	78.030	79.990	74.886	*82.591	*81.485

Table 11: Average accuracy of all classes from a feedforward neural network having 25 hidden neurons with DB-TH Data set using a spectrogram window of 256 Pixel

Size of the y-axis of spectrogram	Size of the x-axis of spectrogram									
	1 (Pixel)	2 (Pixel)	3 (Pixel)	4 (Pixel)	5 (Pixel)	6 (Pixel)	7 (Pixel)	8 (Pixel)	9 (Pixel)	10 (Pixel)
128 Pixel	47.008	61.143	62.755	71.463	75.665	78.338	79.954	79.924	*80.026	*80.494
115 Pixel	45.491	60.693	62.146	70.647	73.701	80.466	78.200	75.645	*80.982	*81.870
102 Pixel	45.314	60.154	61.228	70.253	71.504	72.309	78.982	79.757	*80.458	*82.386
90 Pixel	43.041	63.987	70.857	71.530	77.005	78.195	78.133	78.322	*80.539	*81.660
77 Pixel	50.299	64.197	66.535	69.560	72.020	75.933	78.034	79.010	*80.000	*80.598
64 Pixel	34.378	54.962	47.842	77.780	80.342	67.402	78.235	71.350	75.455	76.969
51 Pixel	43.685	65.274	71.901	75.048	76.972	67.396	70.877	70.836	69.478	73.800
38 Pixel	38.023	56.118	73.498	70.502	64.007	66.102	68.539	69.901	69.130	72.338
26 Pixel	40.420	55.061	52.946	70.691	54.752	66.076	67.199	69.342	69.281	72.146
13 Pixel	38.903	50.417	58.765	71.967	64.762	69.596	66.660	69.530	72.558	72.893

Table 12: Average accuracy of all classes from a feedforward neural network having 25 hidden neurons with DB-TH-ENG Data set using a spectrogram window of 256 Pixel

Size of the y-axis of spectrogram	Size of the x-axis of spectrogram									
	1 (Pixel)	2 (Pixel)	3 (Pixel)	4 (Pixel)	5 (Pixel)	6 (Pixel)	7 (Pixel)	8 (Pixel)	9 (Pixel)	10 (Pixel)
128 Pixel	57.768	68.116	79.042	*82.224	*84.710	*84.730	*86.826	*86.046	*87.753	*88.347
115 Pixel	50.409	71.413	70.672	73.251	77.892	77.869	*80.992	*80.324	*83.066	*87.375
102 Pixel	52.934	76.633	77.537	72.054	76.463	74.757	*80.286	*87.772	*84.208	*85.614
90 Pixel	59.869	73.745	69.568	73.598	74.938	77.552	*81.297	*87.676	*82.232	*80.625
77 Pixel	49.189	55.483	68.973	72.317	73.822	76.589	*82.208	*84.803	*89.853	*80.780
64 Pixel	60.100	63.730	68.537	79.514	73.737	75.514	*83.189	*85.548	*88.000	*80.363
51 Pixel	58.479	56.494	66.127	75.205	72.934	76.803	*80.139	78.934	*85.367	70.548
38 Pixel	62.710	64.232	69.274	71.722	72.811	76.255	73.475	77.224	79.066	78.564
26 Pixel	45.985	66.394	69.413	76.494	73.730	72.888	72.417	78.124	79.514	76.618
13 Pixel	42.680	58.116	56.386	63.954	67.591	73.042	60.015	75.907	73.537	76.526

Table 13: Average accuracy of all classes from a feedforward neural network having 25 hidden neurons with DB-TH Data set using a spectrogram window of 128 Pixel

Size of the y-axis of spectrogram	Size of the x-axis of spectrogram									
	1 (Pixel)	2 (Pixel)	3 (Pixel)	4 (Pixel)	5 (Pixel)	6 (Pixel)	7 (Pixel)	8 (Pixel)	9 (Pixel)	10 (Pixel)
64 Pixel	45.997	63.363	65.215	71.553	67.002	47.652	64.585	77.294	79.908	72.690
58 Pixel	44.913	64.158	68.177	61.438	64.768	71.737	65.163	76.210	72.210	66.332
51 Pixel	44.624	64.926	72.854	76.762	64.053	67.961	70.069	66.706	74.693	71.606
45 Pixel	40.348	50.259	57.445	71.856	75.199	56.519	77.773	65.268	71.146	79.711
38 Pixel	45.576	58.555	68.716	62.463	54.686	59.619	63.074	74.207	75.540	78.516
32 Pixel	38.752	54.798	60.079	66.837	61.911	71.665	50.213	71.179	71.921	61.544
26 Pixel	37.038	53.274	69.793	69.530	62.975	57.432	71.396	67.087	67.146	63.041
19 Pixel	41.452	47.829	61.878	65.366	73.892	66.883	67.429	59.271	57.412	64.624
13 Pixel	33.596	28.388	59.777	61.951	70.312	58.719	77.353	69.780	67.744	55.107
6 Pixel	34.588	50.246	62.837	61.924	47.888	64.887	42.371	60.604	57.438	67.691

Table 14: Average accuracy of all classes from a feedforward neural network having 25 hidden neurons with DB-TH-ENG Data set using a spectrogram window of 128 Pixel

Size of the y-axis of spectrogram	Size of the x-axis of spectrogram									
	1 (Pixel)	2 (Pixel)	3 (Pixel)	4 (Pixel)	5 (Pixel)	6 (Pixel)	7 (Pixel)	8 (Pixel)	9 (Pixel)	10 (Pixel)
64 Pixel	43.907	54.100	63.166	*80.471	*83.931	*85.197	*85.707	*85.699	*85.483	*85.486
58 Pixel	47.158	48.958	72.355	71.243	*83.398	*83.089	*83.367	*82.548	*84.263	*83.004
51 Pixel	49.405	62.208	74.425	79.205	*82.780	*83.598	*84.247	*85.560	*84.139	*84.660
45 Pixel	50.687	59.876	62.911	*84.139	79.405	79.838	73.066	*86.301	*85.699	*84.950
38 Pixel	48.610	66.903	63.490	71.452	76.803	78.564	*81.266	*85.934	*85.301	*84.683
32 Pixel	47.521	56.625	70.703	63.722	77.429	78.178	68.595	*88.703	*80.625	*83.656
26 Pixel	43.467	64.255	68.741	65.622	61.537	72.178	71.421	72.127	71.413	73.900
19 Pixel	45.699	51.097	63.042	63.158	62.718	70.633	71.537	74.456	76.958	72.927
13 Pixel	45.382	43.359	61.459	65.645	63.050	70.564	71.575	72.347	72.116	72.069
6 Pixel	32.247	40.247	51.822	65.722	64.857	70.293	71.992	71.042	72.332	72.100

- Naive Bayes Classifier
- Parzen Classifier
- Decision tree

This Experimented, we used the data from Table 4 and Table 6. Base on our experimental setup, we use a window of 1024 pixels with a 25% overlap. This corresponds to the window size used for all feature extractions. After that, we apply DCT-based compressed algorithms for resize a spectrogram feature to 256×4. As showing in table 7 and 8, we selected an spectrogram feature size 256×4 because it was the smallest size that can provide performance higher than 80% for DB-THS dataset and higher than 90% for DB-TH-ENG data set and this size used for all classification technique in this section.

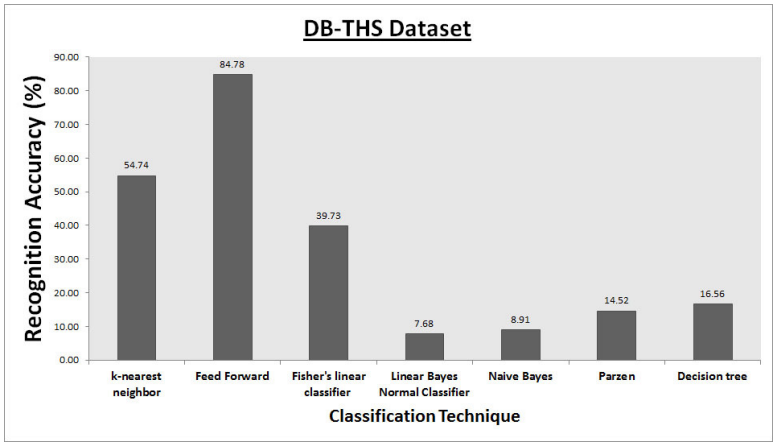


Figure 10: Test classification performance of different classification technique using spectrogram feature on DB-THS Dataset.

We compare the overall recognition accuracy using spectrogram feature and their combination for 12 classes of singing word in DB-THS and DB-TH-ENG data set with 7 classification technique in Fig 10 and 11. As shown in figures,by using a spectrogram feature with image resize technique and feedforward neural network having the highest recognition rate at 84.782% for DB-THS data set and 91.904% for DB-TH-ENG data set.

In this section we will see that the spectrogram feature and feed forward neural network to solve the recognition can be achieved. Especially, spectro-

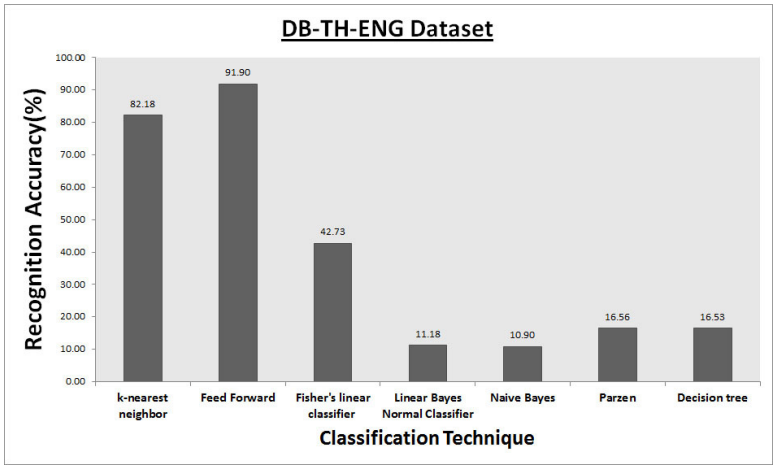


Figure 11: Test classification performance of different classification technique using spectrogram feature on DB-TH-ENG Dataset.

gram feature can recognize Cross-Language Music Data in Table 6 without using any method to separated music in background.

5.5. Compare with Automatic speech Recognition (ASR) algorithm.

An interesting benchmark is shown in Fig. 12,13 and 14 , we ran the same experiments using spectrogram feature feature and compare with Automatic speech Recognition (ASR) algorithm. With Automatic speech Recognition (ASR) algorithm, we used Hidden Markov Model (HMM) with the same data using LPC and MFCC 13 coefficients.

To Compare with Automatic speech Recognition (ASR) algorithm and our algorithm, each singing word in Tables 4 and 6. are randomly divided into four groups of equal sizes. Then, arbitrarily selected three groups are used for training and the rest is used for testing. For cross-validation procedure, the same process is repeated 50 times with the different training and test sets, to ensure that all samples are included at least once in the test set. The mean recognition rate was calculated based on the error average for one run on test set. For our algorithm, we used windows of 1024 to create a spectrogram feature and resize to 256×4 with DCT-based compressed algorithms . A spectrogram feature was performed on Feed Forward Neural Network, We set 25 Hidden Neural Unit for DB-THS dataset and DB-THS-ENG.

To compare the experimental results with Automatic speech Recognition (ASR) algorithm, we used Hidden Markov Model (HMM) with the same

data using LPC and MFCC 13 coefficients. Table III shows a results on ASR experiment, This ASR algorithm gives a lower accuracy result when it applied to singing voice recognition with background music.

Results presented Fig 12 and 13 was show a detail for 12 classes of singing word in Tables 4 and 6 by using feed forward neural network. As shown in this figure, spectrogram features with feed forward neural network tend to best performance. They perform better than ASR in 11 of the examined singing words on DB-THS data set and all singing words on DB-TH-ENG data set.

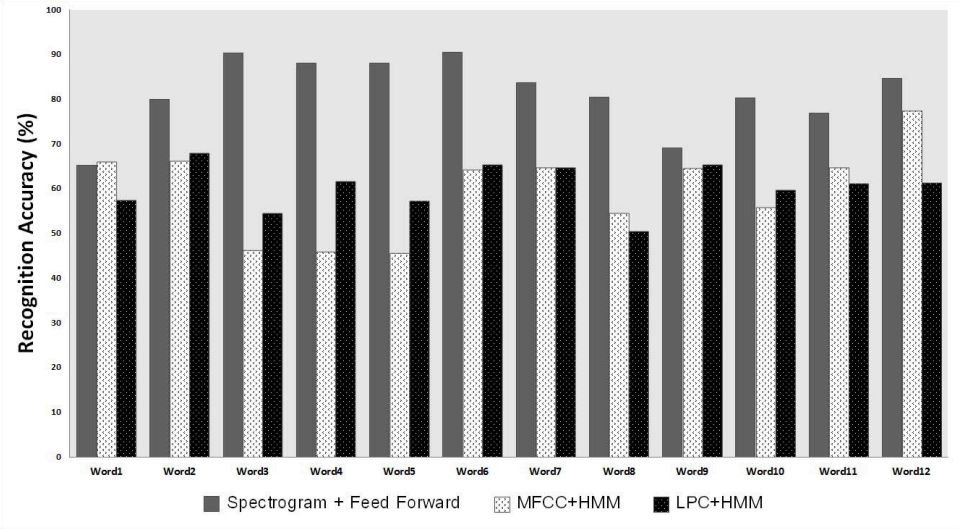


Figure 12: Overall recognition rate 12 classes of DB-THS dataset using spectrogram feature with Feed Forward Neural Network, spectrogram feature with kNN, MFCC feature with HMM and LPC feature with HMM.

Results presented Fig 14 was show overall recognition accuracy comparing spectrogram features with Feed Forward Neural Network for all data set of sounds and Automatic speech Recognition (ASR) algorithm. As shown in this figure, spectrogram features having the highest recognition rate at 84.782% for DB-THS data set and 91.904% for DB-TH-ENG data set. They perform better than Automatic speech Recognition (ASR) algorithm in all data set. Because, Automatic speech Recognition (ASR) algorithm having the highest recognition rate at 60.48% for DB-THS data set and 52.64% for DB-TH-ENG data set for LPC feature and 59.53% for DB-THS data set and 49.99% for DB-TH-ENG data set for MFCC feature.

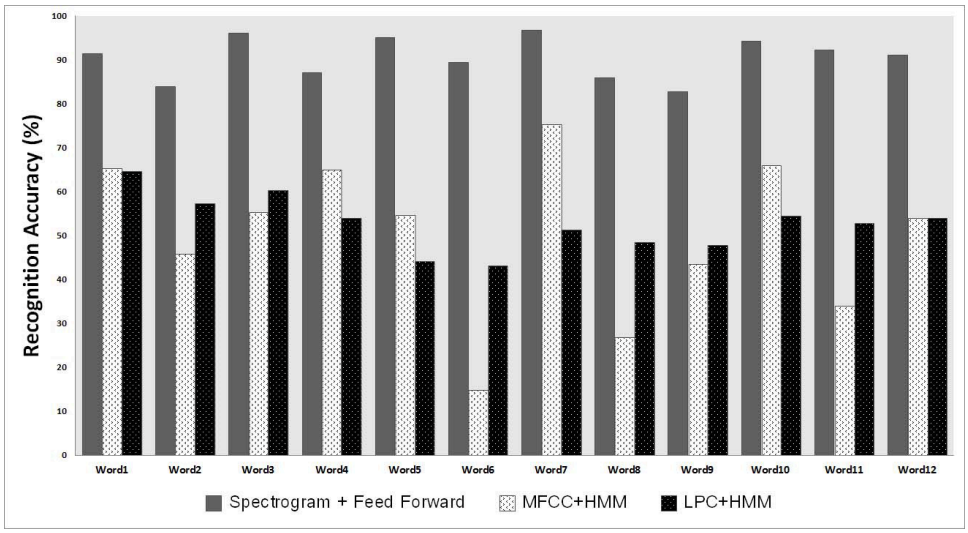


Figure 13: Overall recognition rate 12 classes of DB-THS-ENG dataset using spectrogram feature with Feed Forward Neural Network, spectrogram feature with kNN, MFCC feature with HMM and LPC feature with HMM.

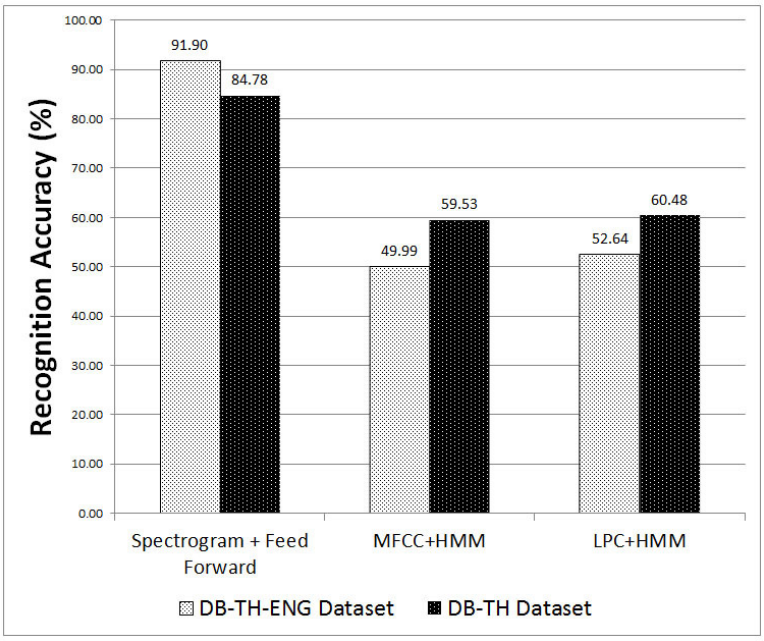


Figure 14: Overall recognition rate.

6. Conclusion

In this paper, we propose an algorithm for singing voice recognition in monaural polyphonic music based on the images of spectrogram with neural classifier, image resizing algorithm and classification algorithms. But a spectrogram is also limited. A dimension of spectrogram feature is very high and time interval of each singing word is not equal. Then we apply image resizing algorithm to solve both problem. The results show all classifiers can recognize a singing word with background music. The experiment showed that feed-forward network performed better than Automatic speech Recognition(ASR) a with accuracy rate 91.904%. Especially, A algorithm can recognize Cross-Language Music Data.

References

Ajmera, J., McCowan, I., Bourlard, H., May 2003. Speech/music segmentation using entropy and dynamism features in a hmm classification framework. Speech Commun. 40, 351–363.
URL http://portal.acm.org/citation.cfm?id=781675.781682

Berenzweig, A., Ellis, D., 2001. Locating singing voice segments within music signals. In: Applications of Signal Processing to Audio and Acoustics, 2001 IEEE Workshop on the. pp. 119 –122.

Berenzweig, Adam L.; Ellis, D. P. W. L. S., 6 2002. Using voice segments to improve artist classification of music. In: Audio Engineering Society Conference: 22nd International Conference: Virtual, Synthetic, and Entertainment Audio.

Chou, W., Gu, L., 2001. Robust singing detection in speech/music discriminator design. In: Proceedings of the Acoustics, Speech, and Signal Processing, 200. on IEEE International Conference - Volume 02. IEEE Computer Society, Washington, DC, USA, pp. 865–868.
URL http://portal.acm.org/citation.cfm?id=1258236.1259164

Cullity, B. D., 2003. Music information retrieval. Vol. 35. Information Today Books.

Dugad, R., Ahuja, N., 2001. A fast scheme for image size change in the compressed domain. IEEE Trans. Circuits Syst. Video Techn. 11 (4), 461–474.

Esmaili, S., Krishnan, S., Raahemifar, K., may 2004. Content based audio classification and retrieval using joint time-frequency analysis. In: Acoustics, Speech, and Signal Processing, 2004. Proceedings. (ICASSP '04). IEEE International Conference on. Vol. 5. pp. V – 665–8 vol.5.

Fujihara, H., Goto, M., Ogata, J., Komatani, K., Ogata, T., Okuno, H. G., 2006. Automatic synchronization between lyrics and music cd recordings based on viterbi alignment of segregated vocal signals. In: Proceedings of the Eighth IEEE International Symposium on Multimedia. ISM '06. IEEE Computer Society, Washington, DC, USA, pp. 257–264.

Gerhard, D. B., 2003. Computationally measurable differences between speech and song. Ph.D. thesis, Burnaby, BC, Canada, Canada, aAINQ81587.

Gruhne, M., Schmidt, K., Dittmar, C., Sep 23-27 2007. Phoneme recognition in pop-pular music. In: 8th International Conference on Music Information Retrieval. Vienna, Austria, pp. 290–294.

Hayashi, T., Ishii, N., Yamaguchi, M., Sept 2014. Fast music information retrieval with indirect matching. In: Signal Processing Conference (EUSIPCO), 2014 Proceedings of the 22nd European. pp. 1567–1571.

Hu, Y., Liu, G., Jun. 2014. Singer identification based on computational auditory scene analysis and missing feature methods. J. Intell. Inf. Syst. 42 (3), 333–352.
URL http://dx.doi.org/10.1007/s10844-013-0271-6

Huang, P.-S., Chen, S., Smaragdis, P., Hasegawa-Johnson, M., March 2012. Singing-voice separation from monaural recordings using robust principal component analysis. In: Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on. pp. 57–60.

Kan, M.-Y., Wang, Y., Iskandar, D., Nwe, T. L., Shenoy, A., 2008. Lyrically: Automatic synchronization of textual lyrics to acoustic music signals. IEEE Transactions on Audio, Speech & Language Processing 16 (2), 338–349.

Kim, Y. E., 2002. Singer identification in popular music recordings using voice coding features. In: In Proceedings of the 3rd International Conference on Music Information Retrieval. pp. 164–169.

Lin, C.-C., Chen, S.-H., Truong, T.-K., Chang, Y., sept. 2005. Audio classification and categorization based on wavelets and support vector machine. Speech and Audio Processing, IEEE Transactions on 13 (5), 644 – 651.

M. Gruhne, K. S., Dittmar, C., Sep 23-27 2007. Phoneme recognition in pop-pular music. In: 8th International Conference on Music Information Retrieval. Vienna, Austria., pp. 2027–2030.

Maddage, N., Wan, K., Xu, C., Wang, Y., june 2004. Singing voice detection using twice-iterated composite fourier transform. In: Multimedia and Expo, 2004. ICME '04. 2004 IEEE International Conference on. Vol. 2. pp. 1347 –1350 Vol.2.

Maddage, N. C., Xu, C., Wang, Y., 2003. An svm-based classification approach to musical audio. In: ISMIR.

Makeyev, O., Sazonov, E., Schuckers, S., Lopez-Meyer, P., Melanson, E., Neuman, M., aug. 2007a. Limited receptive area neural classifier for recognition of swallowing sounds using continuous wavelet transform. In: Engineering in Medicine and Biology Society, 2007. EMBS 2007. 29th Annual International Conference of the IEEE. pp. 3128 –3131.

Makeyev, O., Sazonov, E., Schuckers, S., Melanson, E., Neuman, M., aug. 2007b. Limited receptive area neural classifier for recognition of swallowing sounds using short-time fourier transform. In: Neural Networks, 2007. IJCNN 2007. International Joint Conference on. pp. 1601 –1606.

McVicar, M., Santos-Rodriguez, R., Ni, Y., Bie, T. D., Feb 2014. Automatic chord estimation from audio: A review of the state of the art. Audio, Speech, and Language Processing, IEEE/ACM Transactions on 22 (2), 556–575.

Mesaros, A., Virtanen, T., march 2010. Recognition of phonemes and words in singing. In: Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on. pp. 2146 –2149.

Nwe, T. L., Shenoy, A., Wang, Y., 2004. Singing voice detection in popular music. In: Proceedings of the 12th annual ACM international conference on Multimedia. MULTIMEDIA '04. ACM, New York, NY, USA, pp. 324–327.

Raj, B., 2007. Separating a foreground singer from background music.

Rocamora, M., Herrera, P., sep 2007. Comparing audio descriptors for singing voice detection in music audio files. In: Brazilian Symposium on Computer Music, 11th. San Pablo, Brazil.

Ryynanen, M., Virtanen, T., Paulus, J., Klapuri, A., June 2008. Accompaniment separation and karaoke application based on automatic melody transcription. In: Multimedia and Expo, 2008 IEEE International Conference on. pp. 1417–1420.

Sasou, A., Goto, M., Hayamizu, S., Tanaka, K., 18-23, 2005a. An autoregressive, non-stationary excited signal parameter estimation method and an evaluation of a singing-voice recognition. In: Acoustics, Speech, and Signal Processing, 2005. Proceedings. (ICASSP '05). IEEE International Conference on. Vol. 1. pp. 237 – 240.

Sasou, A., Goto, M., Hayamizu, S., Tanaka, K., 18-23, 2005b. An autoregressive, non-stationary excited signal parameter estimation method and an evaluation of a singing-voice recognition. In: Acoustics, Speech, and Signal Processing, 2005. Proceedings. (ICASSP '05). IEEE International Conference on. Vol. 1. pp. 237 – 240.

Shenoy, A., 2005. Singing voice detection for karaoke application. Proceedings of SPIE 5960, 752–762.
URL http://link.aip.org/link/PSISDG/v5960/i1/p596028/s1Agg=doi

Su, L., Yeh, C.-C., Liu, J.-Y., Wang, J.-C., Yang, Y.-H., Aug 2014. A systematic evaluation of the bag-of-frames representation for music information retrieval. Multimedia, IEEE Transactions on 16 (5), 1188–1200.

Suzuki, M., Hosoya, T., Ito, A., Makino, S., January 2007. Music information retrieval from a singing voice using lyrics and melody information. EURASIP J. Appl. Signal Process. 2007, 151–151.
URL http://dx.doi.org/10.1155/2007/38727

Toyoda, Y., Huang, J., Ding, S., Liu, Y., sept. 2004a. Environmental sound recognition by multilayered neural networks. In: Computer and Information Technology, 2004. CIT '04. The Fourth International Conference on. pp. 123 – 127.

Toyoda, Y., Huang, J., Ding, S., Liu, Y., 2004b. Environmental sound recognition by the instantaneous spectrum combined with the time pattern of power, 169–172.

Tsai, W.-H., Wang, H.-M., Rodgers, D., Cheng, S.-S., Yu, H.-M., 2003. Blind clustering of popular music recordings based on singer voice characteristics. In: ISMIR.

Tzanetakis, G., june 2004. Song-specific bootstrapping of singing voice structure. In: Multimedia and Expo, 2004. ICME '04. 2004 IEEE International Conference on. Vol. 3. pp. 2027 – 2030 Vol.3.

Vaizman, Y., McFee, B., Lanckriet, G., Oct 2014. Codebook-based audio feature representation for music information retrieval. Audio, Speech, and Language Processing, IEEE/ACM Transactions on 22 (10), 1483–1493.

Wang, J.-C., Lee, H.-P., Wang, J.-F., Lin, C.-B., jan. 2008. Robust environmental sound recognition for home automation. Automation Science and Engineering, IEEE Transactions on 5 (1), 25 –31.

Wong, C., Szeto, W., Wong, K., Mar. 2007. Automatic lyrics alignment for Cantonese popular music. Multimedia Systems 12 (4/5), 307–323.

Yaguchi, Y., Oka, R., 2005. Song wave retrieval based on frame-wise phoneme recognition. In: Lee, G., Yamada, A., Meng, H., Myaeng, S. (Eds.), Information Retrieval Technology. Vol. 3689 of Lecture Notes in Computer Science. Springer Berlin / Heidelberg, pp. 503–509.

Yoshii, K., Goto, M., Okuno, H. G., jan. 2007. Drum sound recognition for polyphonic audio signals by adaptation and matching of spectrogram templates with harmonic structure suppression. Audio, Speech, and Language Processing, IEEE Transactions on 15 (1), 333 –345.

Zwan, P., Szczuko, P., Kostek, B., Czyzewski, A., 2008. Transactions on rough sets ix. Springer-Verlag, Berlin, Heidelberg, Ch. Automatic Singing Voice Recognition Employing Neural Networks and Rough Sets, pp. 455–473.