



IMAGE ENHANCEMENT USING FUZZY LOGIC

تقدمة الطلاب:

عادل كبول – عبد الرحمن القطيفاني – أحمد السلطان

المشرفين:

د. ميساء أبو قاسم

م. عليا حموي

م. باسل المدني

Table of Contents

4	الخوارزمية الأولى (Image contrast enhancement using fuzzy clustering)
4	توصيف الخوارزمية:
4	مراحل الخوارزمية:
4	المرحلة الأولى:
4	المرحلة الثانية:
5	المرحلة الثالثة:
5	ملاحظات:
5	الخوارزمية الثانية (Image enhancement by increasing contrast using fuzzy logic)
5	توصيف الخوارزمية:
6	توصيف مجالات الانتماء:
7	مراحل عمل الخوارزمية:
8	الخوارزمية الثالثة (Image enhancement by adjusting contrast values using fuzzy logic)
8	توصيف الخوارزمية:
8	توصيف مجالات الانتماء:
9	مراحل عمل الخوارزمية:
9	النتيجة:
9	مراجع:

مقدمة:

قمنا باستخدام ثلاث خوارزميات معالجة صور تعتمد على المنطق الضبابي, و قمنا بمقارنة الأداء لاختيار افضل خوارزمية.

الخوارزمية الأولى:

تعتمد على ال-c-means لتجميع pixels الصورة المتقاربة من حيث الإضاءة, ثم نقوم بعمل histogram equalization (HE) بشكل موضعي لكل تجمع pixels على حدى.

الخوارزمية الثانية:

قمنا بإنشاء نظام منطق ضبابي, بحيث يتم التحكم بتباين الصورة, حيث يكون دخل النظام هو pixel من الصورة, و الخرج هو كمية التعديل في التباين الواجب إضافتها أو طرحها من ال-pixel.

الخوارزمية الثالثة:

كما في الخوارزمية الثانية, فهي تعتمد على نظام المنطق الضبابي, و لكن الاختلاف هو تابع الانتماء الخاص بعمليات الدخل, حيث يتم انشاء تابع انتماء membership function خاص اعتمادا على قيمة التباين الوسطي للصورة mean contrast, و اعتمادا على ذلك يتم تعديل قيم التباين لpixels الصورة.

الخوارزمية الأولى (Image contrast enhancement using fuzzy clustering)

توصيف الخوارزمية:

كما ذكرنا، الخوارزمية تعتمد على الfuzzy clustering، و تمر بعدة مراحل للحصول على النتيجة. ضمن الكود، لدينا الصف المسؤول عن عملية المعالجة ImageHEWithFuzzyClusteringLocally، حيث يحوي على التابع التالي:

```
def enhance(self, image, m, n, is_gray)
```

حيث يتم تمرير الصورة (image)، و تحديد فيما اذا كانت الصورة رمادية ام لا (is_gray)، و يتم التعامل مع الصورة على عدة خطوات.

مراحل الخوارزمية:

المرحلة الأولى:

اعتمادا على خوارزمية الC-means يتم تجميع pixels الصورة ذات القيم المتقاربة، حيث تم استخدام الخوارزمية المبرمجة سابقا، ضمن التابع enhance يتم استدعاء تابع اخر:

```
def preprocess_image(self, real_image, image_vector, m, n, chunking, is_gray)
```

يقوم التابع بأخذ الصورة و تطبيق عملية الClustering عليها، عن طريق تعريف متغير من الصف FuzzyClustering و تطبيق عملية fit على الصورة، بحيث يكون ناتج هذه العملية هي مراكز العناقيد و مصفوفة الانتماء الخاصة بpixels الصورة، و اعتمادا على ذلك يتم تجميع الpixels ضمن كتل بحيث يتم معالجة كل كتلة على حدة، و يتم التجميع ضمن بنية dictionary، لتسهيل عملية المطابقة لاحقا.

المرحلة الثانية:

في حال كانت الصورة ملونة، يتم تحويل نمط الصورة من RGB الى HSV (Hue - Saturation - Value)، و يتم التعديل على قيمة الValue التي تمثل الإضاءة عند تنفيذ عملية موازنة الهيستوغرام Histogram equalization، و لكن للقيام بهذه العملية بشكل موضعي، تم تطبيق عملية التحويل بعلاقة خاصة للمحافظة على حدود قيم التباين ضمن كل كتلة، و العملية تعطى بالشكل التالي:

$$g(k) = start_i + (end_i - start_i) * \left(\frac{n_k}{m_i}\right)$$

حيث Start هي أصغر قيمة تباين ضمن الكتلة الحالية i، و End هي أكبر قيمة تباين، بحيث يمثلان حدود مجال الكتلة i، n_k هي عبارة عن مجموع جميع قيم الكتلة الحالية التي قيمها أصغر

من قيمة ال pixel الحالي k, و m هي مجموع قيم ال pixels للكتلة الحالية, بحيث تمثل النسبة ما يعرف بالCDF (Cumulative Distribution Function), و التي تعبر عن مقدار توزع القيمة في الصورة, بحيث يتم انطلاقا من قيمة أصغر تباين, زيادة نسبة معينة اعتمادا على مقدار توزع القيمة.

في نهاية هذه المرحلة نكون قد قمنا بتعديل قيم التباين لل pixels ضمن كل كتلة بشكل موضعي, و طبعا في حال كانت الصورة رمادية gray scale نقوم بنفس العملية ولكن دون التحويل الى النمط اللوني HSV.

المرحلة الثالثة:

يتم مطابقة القيم المعدلة بالاستفادة من بنية ال dictionary, و إعادة توزيع ال pixels ضمن مواقعها الصحيحة, و في حال الصورة الملونة يتم إعادة تحويل الصورة من النمط HSV الى النمط الأساسي RGB.

ملاحظات:

وجدنا أنه رغم كون الخوارزمية منطقيا صحيحة, الا ان عمل الخوارزمية خاص جدا, ففي حال كان مجال التباين في الصورة صغيرا, فان التعديل على الصورة لن يكون واضحا أبدا, و حتى في حال كان التعديل على صورة ذو مجال تباين كبير, الا أن التغيير لن يكون واضحا, لذلك قمنا بزيادة حجم مجال التباين لكل كتلة بشكل نسبي.

الخوارزمية الثانية (Image enhancement by increasing contrast using fuzzy logic)

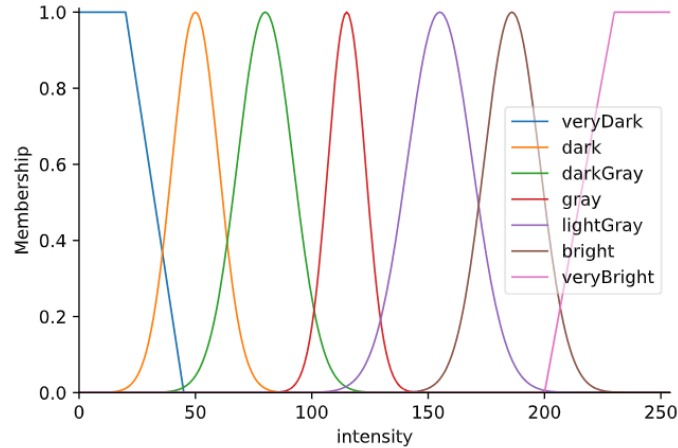
توصيف الخوارزمية:

الخوارزمية تعتمد بشكل أساسي على نظام FIS (Fuzzy Inference System), حيث يتكون النظام من دخل Input معرف بشكل ضبابي, خرج Output أيضا بشكل ضبابي, و مجموعة من القواعد Rules تحدد شكل الخرج اعتمادا على الدخل, حيث يكون نظام الدخل و الخرج معرف بشكل ضبابي عن طريق توابع انتماء membership functions معرفة سابقا, بحيث يكون لها مستويات معينة و قيم و مجالات توزع لكل مستوى, و تعمل الخوارزمية على عدة مراحل.

توصيف مجالات الانتماء:

ضمن الinput يوجد عدة مجالات تمثل قيمة الإضاءة للpixel :

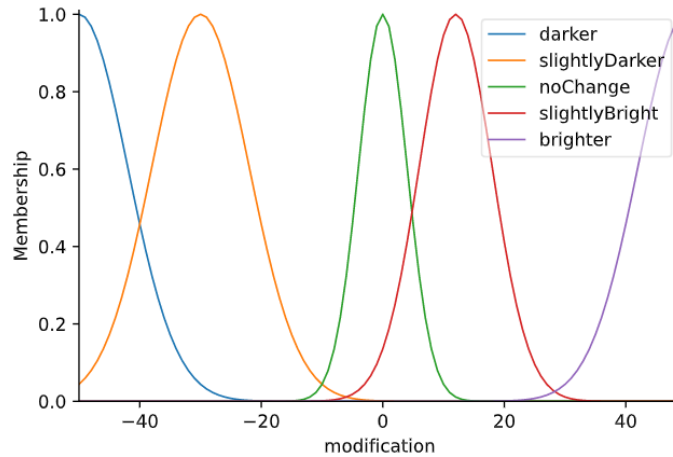
(Very Dark, Dark, Dark Gray, Gray, Light Gray, Bright, Very Bright), و قيم المجالات



موضحة ضمن المخطط التالي:

ضمن الoutput يوجد مجالات تمثل كمية الإضاءة الواجب اضافتها لتحسن التباين:

(Darker, Slightly Darker, No Change, Slightly Brighter, Brighter), حيث تم تحديد قيم



المجالات اعتمادا على تجارب و عينات من مجموعة من الصور تم دراستها, و هذه المجالات موضحة ضمن المخطط:

و يتم الربط عن طريق مجموعة من القواعد موصفة كالتالي:

- 1) IF input is Very Dark(VD) THEN output is Slightly Dark(SD)
- 2) IF input is Dark Gray(DG) THEN output is Slightly Dark(SD)
- 3) IF input is Gray(G) THEN output is Slightly Dark(SD)
- 4) IF input is Bright(B) THEN output is Slightly Bright(SB)
- 5) IF input is Dark(D) THEN output is Very Dark(VD)
- 6) IF input is Very Bright(VB) THEN output is Slightly Bright(SB)
- 7) IF input is Light Gray(LG) THEN output is Slightly Dark(SD)

مراحل عمل الخوارزمية:

ضمن الكود يتم انشاء متغير من الصف FuzzyImageEnhancement لتنفيذ عملية التعديل, حيث تتم العملية ضمن التابع التالي:

```
def init_fuzzy_system(self, image, is_gray)
```

أولا في حال كانت الصورة ملونة يتم تحويل نمط الألوان من RGB الى HSV للتعامل مع قيمة الإضاءة V, ثم يتم انشاء توابع ال input و ال output و rules الخاصة بال fuzzy system اعتمادا على مكتبة sk-fuzzy, حيث تؤمن البيئة اللازم لتشكيل توابع الانتماء لكل عنصر في النظام و الربط بين دخل و خرج النظام عن طريق القواعد.

ضمن الكود المتغيرات الأساسية التي تمثل النظام الضبابي هي, intensity تمثل دخل النظام, adjustment قيمة التعديل أو خرج النظام, rules قواعد النظام, modification المتحكم الذي يربط بين المكونات السابقة.

بعد اكتمال عملية الانشاء, يتم أخذ كل pixel من الصورة و ادخال قيمته ضمن النظام للحصول على القيمة التباين الواجب اضافتها, و تعديل هذه القيمة ضمن الصورة الأساسية, و بعد تطبيق العملية على كل الصورة, نكون قد حصلنا على الصورة المحسنة, و في حال كانت الصورة ملونة, يتم إعادة تحويل الصورة الى النمط اللوني RGB.

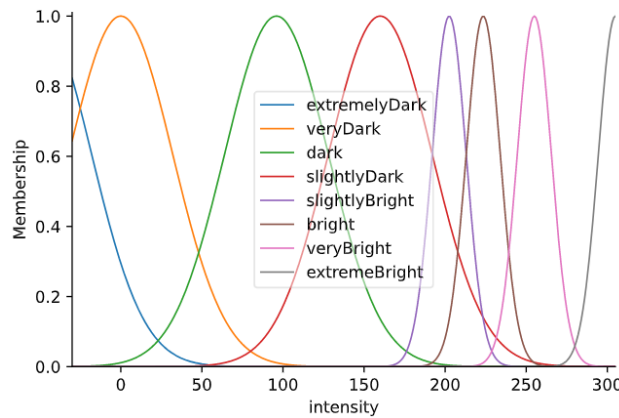
الخوارزمية الثالثة (Image enhancement by adjusting contrast) (values using fuzzy logic)

توصيف الخوارزمية:

نظام هذه الخوارزمية مشابه لنظام الخوارزمية السابقة, الاختلاف هو طريقة تشكل مجالات انتماء توابع الدخل و الخرج, حيث كانت هذه القيم في الخوارزمية السابقة مدروسة و محددة مسبقا, أما في هذه الخوارزمية فيتم حساب التباين المتوسط للصورة أو تباين الصورة العام mean intensity, و اعتمادا على تلك القيمة يتم تحديد مجالات انتماء الإضاءة لكل مستوى, و ضمن الكود نجد المعادلات الخاصة بتعريف الانتماء موضحة ضمن التابع `def _init_input(self, m)` لتابع انتماء الدخل, و ضمن التابع `def _init_output` لتابع انتماء الخرج.

توصيف مجالات الانتماء:

مجالات الدخل و الخرج تأخذ نفس القيم, و كمثال في حال كانت قيمة التباين الوسطي $m=192$, تكون مجالات الانتماء على الشكل التالي:



و توصف القواعد بالشكل التالي:

- 1) IF input is Very Dark(VD) THEN output is Extremely Dark(ED)
- 2) IF input is Dark(D) THEN output is Very Dark(VD)
- 3) IF input is Slightly Dark(SD) THEN output is Dark(D)
- 4) IF input is Slightly Bright(SB) THEN output is Bright(B)
- 5) IF input is Bright(B) THEN output is Very Bright(VB)
- 6) IF input is Very Bright(VB) THEN output is Extremely Bright(EB)

مراحل عمل الخوارزمية:

كما في السابق, يتم انشاء متغير من الصف FuzzyImageEnhancement لتنفيذ عملية التعديل, و لكن في هذه الخوارزمية, في حالة الصورة الملونة يتم تحويل نمط الألوان من RGB الى النمط CIELAB حيث يتمثل بثلاث متغيرات, L هو الإضاءة و a و b لتحديد اللون, و نلاحظ أن مجال التغير ضمن هذا النمط يتراوح من 50- أي 306, و عمل الخوارزمية يتم كما في السابق, و كخطوة إضافية, عند انشاء توابع انتماء الدخل و الخرج, يتم حساب التباين الأوسطي للصورة الحالية m و اعتمادا عليه يتم تشكيل توابع الانتماء للدخل ضمن التابع `def _init_input(self, m)`, و تشكيل توابع الانتماء للخرج ضمن التابع `def _init_output(self, m)`, ثم من أجل كل pixel يتم ادخال قيمة الإضاءة L للنظام الضبابي و الحصول على القيمة الجديدة الواجب تعديلها على ال pixel.

ضمن الكود المتغيرات الأساسية التي تمثل النظام الضبابي هي, intensity تمثل دخل النظام, new_intensity القيمة الجديدة أو خرج النظام, rules قواعد النظام, modification المتحكم الذي يربط بين المكونات السابقة.

النتيجة:

بمقارنة النتائج بين الخوارزميات السابقة, نلاحظ أن الخوارزمية الثالثة أدائها أفضل بكثير من الخوارزميتين السابقتين, و لا يوجد حالات خاصة للصورة يجب تواجدها حتى تعمل الخوارزمية, و انما تتكيف مع شكل توزيع قيم الصورة قبل القيام بعملية التحسين, و ضمن الملفات المرفقة يوجد أمثلة مطبقة على كل خوارزمية لنوعي الصور الملونة و الرمادية.

مراجع:

:Image contrast enhancement using fuzzy logic

<https://arxiv.org/ftp/arxiv/papers/1809/1809.04529.pdf>

Image contrast enhancement using fuzzy clustering with adaptive cluster
:parameter and sub-histogram equalization

<https://www.sciencedirect.com/science/article/abs/pii/S1051200416301828>

:Fuzzy Logic – Image Contrast Enhancement

<https://www.kaggle.com/nguyenvlm/fuzzy-logic-image-contrast-enhancement>