



جامعة ديالى
كلية التربية للعلوم الصرفة
قسم علوم الحاسوب

نظام ضغط الصور الرقمية



بحث تخرج تقدم به الطالب:

سامح كريم غايب

الى قسم علوم الحاسوب في كلية التربية للعلوم الصرفة / جامعة ديالى وهو
جزء من متطلبات نيل شهادة البكالوريوس في الحاسوب

بإشراف

م.د. عادل إبراهيم



الفصل الأول

مقدمة

1.1 مقدمة البحث:

البيانات التي تخزن على الحاسبة بأي نوع من أنواع الملفات غالبا ما تكون هذه الملفات كبيرة الحجم داخل الحاسبة على الرغم من أن المعلومات المطلوبة تكاد تكون قليلة لكنها تشغل حيزا كبيرا من الذاكرة وتحتاج إلى زمن كبير في حالة نقلها على شكل حزم لذا أصبحت الحاجة إلى ضغط البيانات) الذي هو تحويل البيانات إلى حجم أقل (لتكوين ملفات ذات كفاءة أعلى في التخزين وكذلك في حالة نقل البيانات عبر الأوساط المتعددة (Multimedia) وان التقدم الحاصل في مجال تقنية نقل المعلومات لم يكن عمليا ان لم يضع البيانات على شبكة النقل بدون ضغط (compression)

اغلب الأشخاص العاملين في مجال الصور أو مجال نقل البيانات أو حتى العاملين على الصور المتحركة يستخدمون الخوارزميات الخاصة في عملية الضغط ليتسنى لهم نقل البيانات بحجم أقل من حجمها الأصلي ولاجل أن يكون زمن النقل أقل ما يمكن بحيث أن البيانات المنقولة لا تتشوه ويتعامل مع البيانات على أساس أنها بيانات رقمية التمثيل . فإذا أردنا تخزين بيانات لصورة صغيره بحجم (4*4) انج ملونة مقروءة (300) نقطة /انج بحجم (24bit/pixel) وبملف حجمه (4Mbyte) فإنه في هذه الحالة تحتاج الى ما يعادل ثلاثة أقراص مرنة كما انه إذا أردنا نقلها عبر وسط ناقل للبيانات بسعة (64Kbyte) فإنها تحتاج إلى أكثر من دقيقة لنقلها ولو أن زيادة عرض الحزمة هو حل محتمل إلا انه ذو كلفة عالية، إذن لابد من الحاجة إلى عملية ضغط (compression)، وعليه تظهر الحاجة الكبيرة لاستخدام إحدى خوارزميات الضغط في معالجة المشاكل الحاصلة وقد تطورت خلال العقود الماضية خوارزميات ضغط البيانات النصية أو الصور أو البيانات و الأصوات [2].

هنالك عدة أسباب دعت إلى تطوير فرع الحاسب المتعلق بضغط البيانات منها توفير المساحة التخزينية على القرص التي تحتلها الملفات كبيرة الحجم مما يسمح بتخزين ملفات أكثر , وبالنسبة لعالم الشبكات والإنترنت فالموضوع حيوي و هام للغاية ,حيث أنه اليوم يتم التعامل مع ملفات كبيرة الحجم كملفات الصوت و الصور وملفات الفيديو و غيرها، فلو تم إرسال هذه الملفات بصيغتها الأصلية بدون ضغط سيحدث اختناق في حركة المرور عبر الشبكة , وفي نفس الوقت سيتم نقل بيانات أقل بكثير لأن الوقت الذي تستغرقه عملية إرسال ملف واحد سيكون كبيرا .

[3] .

2.1 هدف البحث

البيانات التي تخزن على الحاسبة بأي نوع من أنواع الملفات غالبا ما تكون هذه الملفات كبيرة الحجم داخل الحاسبة على الرغم من أن المعلومات المطلوبة تكاد تكون قليلة لكنها تشغل حيزا كبيرا من الذاكرة وتحتاج إلى زمن كبير في حالة نقلها على شكل حزم لذا أصبحت الحاجة إلى (ضغط البيانات) الذي هو تحويل البيانات إلى حجم أقل (لتكوين ملفات ذات كفاءة أعلى في التخزين وكذلك في حالة نقل البيانات عبر الأوساط المتعددة (Multimedia) وان التقدم الحاصل في مجال تقنية نقل المعلومات لم يكن عمليا ان لم يضع البيانات على شبكة.

3.1 نطاق البحث

هناك العديد من الخوارزميات التي تقوم بتحويل معلومات الصورة من المجال المكاني إلى المجال الترددي من هذه التحويلات تحويل فوريير السريع (Fast Fourier Transform (FFT و بما أن معلومات الصورة لا تحوي أية مركبات تخيلية يتم استخدام خوارزمية تحويل أسرع من FFT و مشتقة منها هي تحويل جيب التمام المنقطع (Discrete Cosine Transform (DCT و هي تتطلب عمليات ضرب أقل من FFT التي تتعامل مع الأعداد الحقيقية 2D DCT. يُنتج مركبات هامة أقل و بذلك يؤدي إلى ضغط أكبر للصورة اعتمادا على طريقتين المستخدمتين في البحث :

• تحويلات المويجة. (wavelet)

• Arithmetic coding.

4.1 خطة بحث

الفصل الأول : في الفصل تم شرح بشكل عام عن الضغط وماهي فوائد الضغط وماهي التطبيقات المستخدمة في الضغط

الفصل الثاني : في هذا الفصل يتم شرح طرق ضغط الصورة في البحث بشكل مفصل

الفصل الثالث : في هذا الفصل تم تصميم وتنفيذ برنامج بلغة فجول بيسك اصدار السادس لضغط الصورة باستخدام أسلوبين

• ضغط بدون فقدان Lossless Compression

• ضغط بفقدان Lossy Compression

الفصل الرابع : الاستنتاجات والاعمال المستقبلية والتوصيات

5.1 مكونات البحث المادية:

استخدم في هذا البحث كمبيوتر لابتوب ذات مواصفات

- CPU: Core i5
- CPU SPEED: Info 2.8 Ghz
- RAM: 5 GB
- OS: Windows 7
- VIDEO CARD: Nvidia GTX 660 or AMD Radeon HD 7870
equivalent DX11 GPU

6.1 مكونات البحث البرمجية:

استخدم في هذا البحث لغة الفجول بيسك الإصدار السادس مع مكتبة الخاصة بعرض الصورة في البرنامج.

الفصل الثاني

طرق ضغط بيانات الصورة

1.2 مقدمة عن الضغط

ضغط البيانات هو فرع من فروع علم الحاسب الآلي ، بدأ هذا الفرع يرى النور على يد العالم "شانون 1948" م وارتبط هذا الفرع مع طرق التشفير والترميز وأصبحت تطبيقاتهم تكاد لا تحصى فقد أحدثا نقلة نوعية في مجال تكنولوجيا المعلومات والاتصال وساهما في تطوير بل وظهر عدد من التطبيقات التي اعتمدت بشكل كامل أو شبه كامل على ترميز وضغط المعلومات قبل إرسالها أو تخزينها ، فعلى سبيل المثال لا الحصر يمكن سرد بعض التطبيقات التي تطورت بفضل تطور هذا الفرع من العلوم:

- التصوير الضوئي الرقمي. Digital Photography
- أجهزة الاستقبال من الأقمار الصناعية الرقمية. Digital Satellite Receivers
- أجهزة الترفيه المنزلية. (DVD , VCD, MP3 . etc)
- أجهزة التليفونات المحمولة.
- نقل الصوت والصورة عبر قنوات الاتصال المختلفة مثل الإنترنت، أجهزة
- المحمول، الشبكات اللاسلكية وغيرها.
- الأجهزة الطبية المتعاملة مع الصور مثل السونار وأجهزة الأشعة الرقمية

2.2 أساليب الضغط

1.2.2 الضغط بدون فقدان:

يتم الضغط بهذه الطريقة بحيث أن البيانات الأصلية للصورة تسترد بشكل دقيق بدون فقدان أي جزء منها ، أي أن هذا النوع لا يسمح بأي اختلاف بين الصورة الأصلية والصورة المركبة بعد أعادتها من الضغط أي أن بيانات الصورة يتم أعادتها بدون فقدان ومن تلك الخوارزميات:

- 1- Run Lenght Code (RLE) .
- 2- Huffman coding.
- 3- Lemple-Ziv-Wett (LZW).
- 4- Arithmetic coding.

ففي خوارزمية (RLE) تتم عملية التشفير على طول مرحلة تنفيذ الضغط وتسمى في بعض الأحيان التشفير المتكرر ، وهي واحدة من أبسط خوارزميات الضغط التي تكون مجاميع من البيانات بأطوال متكررة من رموز متكررة على سبيل المثال إذا كان لدينا الجملة التالية "AAAAAABBBBBCCCCC" التي تتكون من 15 حرفاً يتم ضغطها بوضع اسم الرمز

المكرر وعدد مرات التكرار مع وضع راية (flag) يشير الى هذه المعلومات المكبوسة ،أي يتم ضغطها بالطريقة الآتية ($A6*B4*C5$) بحيث أصبح طولها 9 أحرف وهكذا ، ولو كانت المعلومات بهذه الحالة (AAA) لا يتم ضغطها لأنها مساوية لحجم النص المكبوس.

أما طريقة هوفمان (Huffman) الذي يعتمد على الاحتمالية الحاصلة على الرموز التي تحدد عدد الكلمات المرمره (codeword) ومعرفة ترددات كل حرف موجود في الملف النصي. هذه الطريقة تتطلب مسارين متعاقبين :

المسار الأول : توليد احتمالية تكرار البيانات في الملف .

المسار الثاني : هو ضغط الملف باستخدام الخوارزمية الآتية :

1- ترتيب احتمالية ترددات الحروف بشكل تنازلي واعتبارها عقداً نهائية وعلى هيئة

مخطط شجري. tree

2- تكرر العملية اذا كان هناك اكثر من عقدة واحدة موجودة في المخطط الشجري.

أ -جمع كل عقدتين لهما اقل احتمالية تكرار للتردد لتكوين عقدة جديدة لها احتمالية تكرار مساوٍ لنتائج تردد العقدتين .

ب - ترميز كل زوج من فروع المخطط الشجري بهيئة ثنائية. (1, 0)

أما طريقة (LZW) التي قدمت من الباحثين (Lemple' Ziv) ثم طورها الباحث (Welsh) عام 1984 فانه يتم وضع سلسلة من الرموز مقابل شفره معينة، وهي لا تحمل تحليلاً الى النص القادم ، ويتولد جدول معين يتم الضغط بوضع شفرة معينة بدلا من سلسلة من الرموز، واخراج الشفرة في هذه الخوارزمية ممكن ان يكون (arbitrary).

وهناك ثلاثة تطبيقات لهذه الطريقة:

1. Unix compression (file compression GIF).
2. Image compression(GIF)
3. Compression over modem

أما الخوارزمية الرياضية (Arithmetic coding) التي تسمى الشفرة الرياضية فهي مشابهة لطريقة هوفمان ألا أنها تستخدم أنماطاً مختلفة من الرموز المحتملة وينجز ترميزاً أفضل من طريقة هوفمان، حيث يكون الطول المثالي للشفرة باحتمالية $(1/2^x)$ حيث ان X عدد صحيح

1. الضغط بفقدان (lossy compression)

هذا النوع من الضغط يتطلب بعض الخسارة من البيانات التي لا يمكن أن تسترد أو يعاد تركيبها بصورة مطابقة للصورة الأصلية بحيث أن هذا الفقدان لا يكون مؤثراً عند إعادة تركيب الصورة أي أن نسبة التشوه في الصورة قليل جداً ، وكلما كان التشوه اقل ونسبة الضغط أعلى تكون النتيجة أفضل ، ومن تلك الخوارزميات:

1. تحويلات الموجة (wavelet)

2. الكمم الاتجاهي (vector quantization)

3. الشفرة التنبؤية (predictive coding)

4. الضغط الكسوري (Fractal Compression)

3.2 الصور الرقمية Digital Images :

الصور الرقمية Digital Images بصورة عامة هي عبارة عن ملفات تحوي معلومات عن الصورة، وتمثل هذه المعلومات قيماً رقمية تدل على البعد المكاني لقيم الصورة والشدة اللونية وهذه المعلومات تخزن على شكل قيم رقمية ضمن ملف الصورة وكل قيمة تعرف على أنها نقطة او Pixel. [4]

تختلف أنواع الصور الرقيم حسب تمثيل هذه الصور في الحاسوب حيث انه يتم تمثيل القيم الرقيم (Pixel) للصور في الحاسوب باستخدام وحدات الحاسوب الثنائية (Bits) وهناك عدة أنواع من هذا التمثيل وهي كما يأتي [4] :

1. تمثيل الصور باستخدام وحدتين ثنائيتين وهي (0,1) وهو ما يعرف بالصور الثنائية او الـ (Binary Image) وهذا النوع من الصور يمثل بلونين فقط وهما الأسود والأبيض.
2. تمثيل الصور باستخدام مجموعة من الوحدات الثنائية لكل نقطة (Byte per Pixel) وهي تعرف بالصور الرمادية وتظهر في هذه الصور مجموعة من التدرجات الرمادية وتكون بين (0-255) تدرج لوني في حالة اعتماد ثمانية وحدات ثنائية لكل نقطة .

3. كما يمكن تمثيل الصور باستخدام ثلاث وحدات ثنائية لكل نقطة (3 byte per pixel) وهو ما يعرف بالتدرج اللوني الحقيقي للصورة (True color image) وتصل تدرجات اللون إلى ما يزيد (4294967296) لون .

لأجل خزن الصور الرقمية في الحاسوب هناك عدة طرائق وحسب الحاجة وطبيعة التعامل مع الصور، والطريقة القياسية لخزن ملفات الصور ضمن الحاسوب هي Bit Map Picture والتي تمتاز بكونها تعطي وصفا فيزيائيا حقيقيا عن الصورة ولهذا السبب يكون حجم الملف كبيراً نسبة إلى باقي أنواع ملفات الصور مثل (JPG) الخ....[5]

يتم اكتساب الصور الرقمية في الحاسبة الإلكترونية وتخزينها داخل ملفات صورية أما باستخدام جهاز الماسح الضوئي (Scanner) او باستخدام آلة تصوير رقمية، وقد أصبحت عملية معالجة الصور اسهل عن طريق التعامل مع ملفات الصور باستخدام الحاسوب وبشكل واسع، إذ أن الهدف الرئيس من معالجة الصور هو تحسين الصور والحصول على نتائج أكثر ملائمة من الصور الأصلية لتطبيق معين[5] .

3.2 كيفية تمثيل الصور الرقمية:

تمثل الصور الرقمية كمصفوفة ثنائية الأبعاد (2-Dimensional array) من الأعداد التي قيمة كل منها تمثل شدة الإضاءة لوحدة صورية (pixel) في الصورة سواء أكانت الصورة ذات تدرج رمادي أو ملونة. هذه القيم يستخدمها ماسح الصورة (Raster) لعرض اللون على الشاشة في الموقع الذي تمثله كل قيمة من هذه القيم .

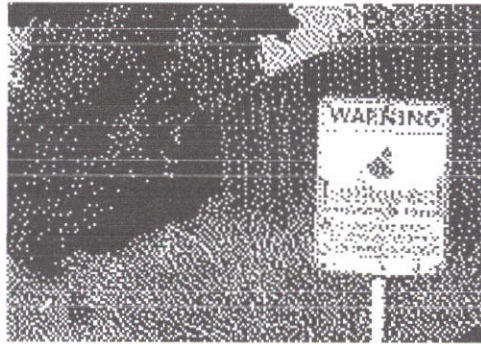
إن عملية الوصول إلى قيم الوحدات الضوئية من قبل الماسح تعتمد على التقنية التي تم بموجبها تمثيل هذه الوحدات ، وهذا بدوره يعتمد على عدد الخلايا الثنائية المستخدمة لتمثيل كل وحدة ، ففي الحاسبات القديمة كان يتم تمثيل كل وحدة بخلية واحدة ثم مع التطور في مجال صناعة الحاسوب تم زيادة عدد الخلايا من خلية واحدة تعطي لونين الى (4)خلايا وتعطي (16) لوناً ، وهكذا حتى الوصول الى التمثيل بثمان خلايا وتعطي (256) لوناً وكانت الأكثر استخداماً حتى ظهرت تقنية التمثيل بـ (24) خلية ثنائية وهي الصور المستخدمة حالياً.

4.2 أنواع الصور: (Image types)

تختلف الصور من حيث عدد الألوان التي تحتويها ومن حيث حجمها بخلاف عدد الخلايا (Bits) التي يحجزها كل لون ، وأهم هذه الأنواع:

1. الصور الثنائية : (Binary images)

في بداية ظهور الحاسبات كانت الصور تمثل بخلية واحدة فكل وحدة صورية لها تكون قيمتها (1 أو 0) أي أسود أو أبيض وتخزن الصورة كمصفوفة ثنائية الأبعاد من الأصفار والواحدات ، لذلك تسمى هذه الصور بـ(صور الأسود والأبيض (أو صورة أحادية اللون ، [6] (Monochrome) والشكل (1.2) يوضح ذلك:



شكل (1.2) صورة ثنائية تحتوي على لونين فقط الأبيض والأسود

2. الصور ذات التدرج الرمادي: (Gray-level images)

بعد صور الأسود والأبيض ، ظهرت الصور ذات التدرج الرمادي وهي صور أحادية اللون أيضاً، لكنها تتكون من ظلال من التدرجات الرمادية تعطي معلومات عن شدة الإضاءة فقط دون اللون . وأن عدد (Bit) المستخدمة لتمثيل كل نقطة تبين عدد مستويات شدة الإضاءة ، والصور الشائعة هي التي تستخدم (8 Bit) أي بإمكانها عرض (256) تدرجاً لونياً , إن هذه الصور على الرغم من إنها غير ملونة لكنها لا تزال تستخدم لحد الآن في كثير من التطبيقات ، الشكل (2-2) يوضح ذلك:



شكل (2.2) صورة رمادية تحتوي على 256

3. الصور الملونة: (Color images)

منذ ظهور الصور الملونة ومع مرور الزمن زاد عرض اللون وزاد عدد التدرجات التي يمكن عرضها لكل لون، لكن بقي المبدأ الأساسي لتمثيل الصور كما هو، حيث تتكون كل (pixel) عن طريق تجميع الألوان الرئيسية الثلاثة الأحمر (Red)، الأخضر (Green)، والأزرق (Blue)، ودمج شدة الإضاءة لهذه الألوان الثلاثة يتم تشكيل اللون المطلوب (RGB)، فمثلاً لوحة صورية موقعها تعطي اللون الأحمر إذا كانت قيمتها (20، 60، 100) وإذا كانت قيمتها (30، 90، 10) ستعطي اللون الأخضر، وسيكون لونها أزرق إذا كانت قيمتها [5] (6,40,200) والشكل (3.2) يوضح ذلك.



شكل (3.2) متجه (RGB) لوحة صورية في الموقع (x,y)

إن قيم مصفوفة الألوان التي تمثل نقاط الصورة في هذا النوع والأنواع السابقة التي ذكرت، لا تمثل القيم الحقيقية للون الوحدات الضوئية بل تمثل مدخلاً (Entity لعنوان موقع في جدول تواجدات الألوان الحقيقية) (Color index أو ما يسمى بلوحة الألوان (Palette) الموجود في بادئة ملف الصورة. (Header)

ويعتمد حجم لوحة الألوان على عدد (Bit) التي تمثل كل وحدة صورية فصورة من نوع (8-bit /pixel) التي لها لوحة ألوان (pallet) يتكون (256) مستوى لونياً، أي أن جدول تواجدت الألوان يحوي (256 مدخل، ومنذ سنوات قليلة انتشرت الصور ذات (24 Bit)، كل وحدة صورية تحجز (3 byte) وبإمكانها عرض (16) مليون لون في الصورة يتم اعتمادها مباشرة من قبل الماسح لإظهار اللون المطلوب للموقع المقابل لها على الشاشة، ولم تعد هناك حاجة للوحة الألوان ولهذا السبب يطلق على هذا النوع من التمثيل بتمثيل اللون الحقيقي (Color true) وعلى الرغم من ذلك لا يوجد فعلاً صورة تتكون أكثر من (16) مليون لون

لأن وببساطة عدد الألوان الحقيقية في أي صورة هو أقل من هذا الرقم، ثم أن العين البشرية لا تستطيع أن تميز أكثر من (20) ألف لون. [5]

5.2 التمثيل الرياضي الصورة (Mathematical representation image)

تمثل الصور الرقمية كمصفوفة ثنائية الأبعاد (2-Dimensional array) من الأعداد التي قيمة كل منها تمثل شدة الإضاءة لوحدة صورية (pixel) في الصورة سواء أكانت الصورة ذات تدرج رمادي أو ملونة وهذه القيم يستخدمها ماسح الصورة (Raste) لعرض اللون على الشاشة في الموقع الذي تمثله كل قيمة من هذه القيم، إن عملية الوصول إلى قيم الوحدات الضوئية من قبل الماسح تعتمد على التقنية التي تم بموجبها تمثيل هذه الوحدات. [6]

وهذا بدوره يعتمد على عدد الخلايا الثنائية المستخدمة لتمثيل كل وحدة، ففي الحاسبات القديمة كان يتم تمثيل كل وحدة بخلية واحدة ثم مع التطور في مجال صناعة الحاسوب تم زيادة عدد الخلايا من خلية واحدة تعطي لونين إلى (4) خلايا وتعطي (16) لوناً، وهكذا حتى الوصول إلى التمثيل بثمان خلايا وتعطي (256) لوناً وكانت الأكثر استخداماً حتى ظهرت تقنية التمثيل بـ (24) خلية ثنائية وهي الصور المستخدمة حالياً.

6.2 الملفات الصورية image files

هناك عدة أنواع من الملفات لخصن الصور بعضها تجارية ومحددة لبرامج معينة أو استخدامات معينة وفيما يأتي عرض لأهم الأنواع استخداماً :-

جدول (2-2) أنواع الملفات الأكثر استخداماً

نوع الملف File Type	وصف الملف Description
BMP (Bitmap)	ملف قياسي لاستخدامات النوافذ
EPS (Encapsulated Postscript)	يستخدم لتمثيل الملفات التي تعمل على مبدأ (Vector –Based)
GIF (Graphic Interchange format)	يستخدم على صفحات الويب وله فقط (256) لوناً وهو غير مناسب للصور العلمية ، وغير مناسب للصور التي تحتوي على الكثير من التفاصيل الصغيرة ، ويستخدم بكثرة في صور الأيقونات (Icons) والنماذج المصغرة من الصور (Thumber)

يستخدم مع الكاميرات الرقمية وعلى صفحات الويب وهو مناسب للصور ذات الأحجام الكبيرة التي فيها الكثير من التفاصيل الدقيقة وهو من نوع (24 Bit)	JPG , JPEG (Joint Photographic Expert Group)
نوع ظهر حديثاً ولم ينتشر بعد ، يدعم الصور بـ (48 Bit)	PNG (Portable Network Graphics)

7.2 معيار الـ (JPEG) في ضغط الصور

أعدّ هذا المعيار من قبل مجموعة من خبراء التصوير Joint Photographic Experts Group. وقد تم تصميمه لمعالجة الصور الثابتة الملونة، ويحقق نسبة ضغط تصل إلى 1:200. ويعتمد على أسلوب تحويل الكتل المميز، Discrete Cosine Transform (DCT) التي تمثل الطريقة التي ترى العين البشرية بها الألوان، حيث أن العين البشرية لا ترى كل الألوان و بالتالي يوجد فائض أو إسهاب (Redundancy) من المعلومات يكون موجوداً في الصورة إذا تمت إزالته لن يؤثر ذلك على رؤية الإنسان. كما يتم في هذا المعيار تقسيم الصورة إلى مجموعة كتل (Blocks) كل كتلة حجمها (8x8) أي ما يعادل 64 بكسل يتم فيها معالجة كل 64 بكسل على حده، ويسقط منها الكتل التي لا تؤثر على العين، بمعنى أن العين البشرية لا تلاحظها و بالتالي فإن إزالتها لن يؤثر على جودة الصورة، كما تقوم بدراسة هذه الكتل و تخزين معلومات عنها فإذا كان لونه أزرق مثلاً فلا داعي لتخزين كل نقطة و إنما يكفي بمعادلة تصف ذلك. لذلك تعتمد تقنية JPEG تقسيم الصورة إلى قطاعات مربعة و تخزين شفرة ألوان للقطاع بأكمله بدلاً من تخزينها لكل نقطة.

تمثل الصورة عادةً كتابع لـ $f(i,j)$ في المجال المكاني، حيث يتم استخدام تحويل جيب التمام المتقطع في البعد الثنائي 2D DCT كخطوة أولية في هذا المعيار من أجل الحصول على الاستجابة الترددية frequency response و التي تمثل بالتابع $F(u,v)$ في المجال المكاني الترددي.

المرحلة الأولى: تقسيم الصورة إلى كتل Blocks

إن القيام بضغط الصورة بشكل كامل سوف يؤدي إلى نتائج ضغط غير مثالية، لهذا السبب يقترح المعيار JPEG أن يتم تقسيم الصورة إلى كتل بحجم 8x8، هذا يسمح للخوارزمية من استغلال حقيقة أن الألوان المتشابهة تميل لأن تظهر معاً في أجزاء صغيرة من الصورة.

تبدأ الكتل من الجزء اليساري العلوي للصورة ثم يتم إنشاؤها تباعاً حتى الجزء السفلي اليميني منها

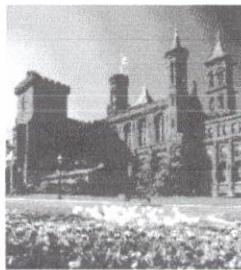
إذا كانت أبعاد الصورة ليست من مضاعفات العدد 8 عندها تتم إضافة بكسلات (حشو) إلى الجزء السفلي و اليميني من الصورة حتى تصبح أبعاد الصورة ملائمة للتقسيم ,البكسلات المضافة عند الحشو يتم إزالتها بسهولة خلال عملية فك الضغط.

بدءاً من هذه المرحلة كل كتلة من هذه الكتل المؤلفة من 64 بكسل تُعالج بشكل منفصل عن بقية الكتل .تتضمن هذه المرحلة أيضاً تغيير للنموذج اللوني للصورة اختياريّاً ,حيث تستخدم 8 بت لتمثيل كل بكسل.

على أي حال ,عمليات الضغط ستكون أقل تعقيداً إذا تم تحويل قيم النموذج اللوني - RGB و الذي يحوي ثلاث مستويات لونية - إلى نموذج لوني آخر .عادة تقوم الـ (JPEG) بتحويل النموذج RGB إلى النموذج (YCbCr) في هذا النموذج تشير الـ Y إلى الإنارة (luminance)و التي تمثل كثافة اللون .أما Cb و Cr هي القيم اللونية و هي تمثل اللون فعلياً. YCbCr يميل لأن يُضغط بفعالية أكثر من النموذج RGB. تجدر الإشارة إلى أن التحويلات بين النماذج اللونية للصورة يمكن أن تتم قبل عملية التقسيم إلى كتل و هذا يرجع إلى كيفية تحقيق الخوارزمية.

أخيراً ,تقوم الخوارزمية بطرح 128 من قيمة كل بكسل في الكتلة ,Block يؤدي ذلك إلى تغيير مجال قيم بايتات البكسلات من 0..255 إلى -128..127 و هكذا معدل القيم لمجموعة كبيرة من البكسلات تصبح قريبة من الصفر.

تبيين الصور التالية مثال عن تقسيم الصورة إلى كتل بحجم 8x8.



Before:

After:

الشكل (4.2) تقسيم الصورة إلى كتل

المرحلة الثانية: التحويل إلى المجال الترددي:

يمكن تخطي هذه المرحلة و الانتقال مباشرة إلى المرحلة التالية) التكميم, (و لكن عملية التحويل إلى المجال الترددي ستجعل عملية التكميم Quantization أسهل في معرفة أي الأجزاء من الصورة أقل أهمية. إن قيمة كل بكسل في الكتلة تمثل شدة ذلك البكسل, و كما ذكرنا أن العين البشرية تُسقط المركبات ذات الترددات العالية, و بالتالي فإن حذف هذه المركبات من الصورة سيعطي صورة جديدة مختلفة جداً من وجهة نظر الحاسب و لكنها تبدو قريبة جداً من الصورة الأصل بالنسبة للعين. إن مرحلة التكميم Quantization تعتمد على هذه الحقيقة لحذف المركبات عالية التردد من الصورة.

هناك العديد من الخوارزميات التي تقوم بتحويل معلومات الصورة من المجال المكاني إلى المجال الترددي من هذه التحويلات تحويل فوريير السريع (Fast Fourier Transform) (FFT) و بما أن معلومات الصورة لا تحوي أية مركبات تخيلية يتم استخدام خوارزمية تحويل أسرع من FFT و مشتقة منها هي تحويل جيب التمام المنقطع Discrete Cosine Transform (DCT) و هي تتطلب عمليات ضرب أقل من FFT التي تتعامل مع الأعداد الحقيقية 2D DCT. يُنتج مركبات هامة أقل و بذلك يؤدي إلى ضغط أكبر للصورة.

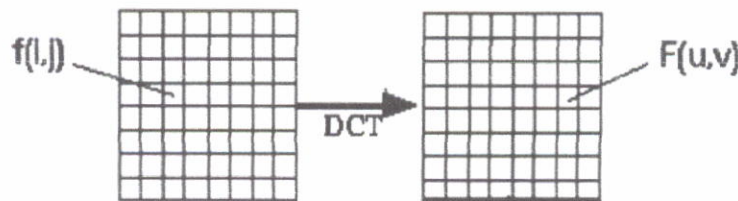
بشكل عام يعبر عن التحويل 2D DCT بالمعادلة التالية:

$$F(u,v) = \frac{1}{4} \cdot C(u)C(v) \left[\sum_{x=0}^7 \sum_{y=0}^7 f(x,y) \cos \frac{(2x+1) \cdot u\pi}{16} \cos \frac{(2y+1) \cdot v\pi}{16} \right]$$

حيث أن:

و التابع هو قيمة بكسل الصورة ذو الإحداثيات و التابع هو القيمة الجديدة لبكسل الصورة في المجال الترددي.

كما في الشكل التالي:



الشكل (5.2) التحويل إلى المجال الترددي

الكتلة الجديدة الناتجة تحوي مركبات ترددية ضمن المجال 1023..1024- القيمة العليا اليسارية من الكتلة Block تسمى مركبة الـ DC و هي المركبة الأعلى تردداً و هي تمثل متوسط كامل الكتلة ، أما بقية المركبات و التي تنقص قيمها الترددية بالاتجاه يميناً و نحو الأسفل فتسمى مركبات الـ AC. المركبات الأقل تردداً تحدث في الجزء السفلي اليميني من الكتلة و هذه المركبات تميل لأن تكون أقل أهمية من المركبات ذات الترددات الأكبر حيث أنها تساهم بشكل قليل في الصورة. نموذجياً ، إن كامل النصف اليميني السفلي من الكتلة سوف يُحذف عند عملية التكميم و بالتالي هذا يعني حذف نصف معلومات الكتلة ، و لهذا السبب تعتبر تقنية JPEG فعالة إلى حد كبير في ضغط الصورة. بالمقابل يُعتبر التحويل DCT من أكثر المراحل استهلاكاً للوقت.

المرحلة الثالثة: التكميم: Quantization

إن تحويل معلومات الصورة إلى المجال الترددي يسمح للخوارزمية بتجاهل الأجزاء أو المركبات الأقل أهمية من الصورة أو الكتلة. تقوم خوارزمية JPEG بتحقيق ذلك من خلال تقسيم مركبات الكتلة على قيم ثابتة) محسوبة سابقاً من خلال التجارب (و من ثم تقريب الناتج إلى أقرب عدد صحيح. القيم الثابتة التي تستخدم في التقسيم قد تكون عشوائية ، و لكن تم وضع قيم نموذجية تم التوصل إليها من خلال الأبحاث. تُعتبر هذه المرحلة الأكثر فقداً لمعلومات الصورة. بشكل أكثر وضوحاً ، فإن التقسيم على قيم ثابتة كبيرة يؤدي إلى إنتاج أخطاء أكثر في عملية التقريب ، و لكن كلما كانت القيم كبيرة يؤدي ذلك إلى اقتراب ناتج القسمة من الصفر و هذا أفضل من أجل المركبات الترددية ذات القيم الكبيرة ، و بذلك تقترب معظم المركبات الترددية من الصفر. تعتمد المرحلة الرابعة على هذه الحقيقة لتضيف ضغط آخر لبيانات الصورة. يبين الشكل التالي قيم مصفوفة التكميم النموذجية:

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

الشكل (6.2) مصفوفة التكميم

نلاحظ بأن قيم مصفوفة التكميم تميل لأن تكون كبيرة في الزاوية اليمينية السفلى , هذا يؤدي إلى فقدان أكبر للمعلومات عند الترددات العالية.

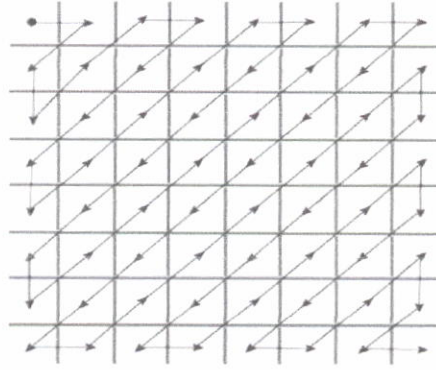
تُعطى معادلة التكميم لكثلة من الصورة بالشكل التالي :

مصفوفة التكميم السابقة يتم حفظها ضمن ملف الصورة بعد الضغط و ذلك من أجل استخدامها عند فك الضغط.

المرحلة الرابعة: الترميز: Entropy Coding

بعد عملية تكميم الصورة , ينتج لدينا كثلة Block مكون من 64 قيمة معظمها أصفار , بالطبع إن أفضل طريقة لضغط هذا النوع من البيانات هي بأن يتم تجميع قيم الأصفار مع بعضها البعض و هذا هو ما تقوم به الـ JPEG فعلياً.

يتم مسح المصفوفة بشكل متعرج Zigzag حيث يتم البدء من مركبة الـ DC و عبور المصفوفة بشكل لولبي باتجاه الأسفل كما يوضح الشكل التالي:



الشكل (7.2) تطبيق الـ Zigzag على الكثلة

سنحصل بالنهاية على شعاع مكون من 64 قيمة , يتم بعدها تطبيق ترميز RLE على مركبات الـ AC فقط حيث يتم تمثيل كل مركبة بالثنائية (skip,value) حيث تمثل الـ skip عدد الأصفار , أما value تمثل القيمة التالية غير الصفرية.

مثال : بفرض لدينا السلسلة التالية الناتجة عن تطبيق مسح zigzag لكثلة ما:

$$(32,6,-1,-1,0,-1,0,0,0,-1,0,0,1,0,0,...,0)$$

القيمة 32 تمثل مركبة الـ DC لذلك نتجاهلها , و بتطبيق RLE على مركبات الـ AC نحصل على

$$:(0,6),(0,-1),(0,-1),(1,-1),(3,-1),(2,1),(0,0)$$

تمثل الثنائية (0,0) نهاية الكثلة بعد الوصول إلى آخر مركبة غير صفرية.

ترميز مركبات الـ DC:

يتم ترميز مركبات الـ DC بشكل مستقل و مختلف حيث يطبق عليها ترميز يسمى DPCM إن قيم مركبات الـ DC بعيدة الاحتمال أن تتغير بشكل كبير و هذا ما يجعل من DPCM ترميز مثالي عندما يطبق عليها و هو يُطبق على كامل الصورة بنفس الوقت .
يتم حساب مركبات الـ DC كما يلي:

$$d_i = DC_{i-1} - DC_i$$
$$d_0 = DC_0$$

حيث أن هي لأول Block من الصورة .

مثلاً :تكن قيم مركبات الـ DC لأول خمس بلوكات من الصورة هي :

150,155,149,152,144 إن تطبيق الـ DPCM عليها يعطي 8,-3,5,-6,150 :

كل قيمة من القيم السابقة يتم تمثيلها بالثنائية (size, Amplitude) حيث أن الـ size هو عدد البتات اللازم لتمثيل مركبة الـ DC و المطال Amplitude هو عدد البتات الفعلي .
و بالتالي قيم الـ (DPCM) التالية سوف تصبح:

(8,10010110), (3,101), (3,001), (2,11), (4,0111)

ترميز مركبات الـ AC:

يتم ترميز هذه المركبات باستخدام الخوارزمية RLC حيث يتم تمثيل كل قيمة بالثنائية (Runlength/Category, Value) حيث أن الـ (Runlength) هو عدد الأصفار و الـ Category يتم حسابها من جدول خاص بها و الـ Value هو القيمة الفعلية لمركبة الـ AC. بحسب المثال السابق (0,0), (2,1), (3,-1), (1,-1), (0,-1), (0,-1), (0,6) :يصبح لدينا بعد الترميز ما يلي :

(0/3,6) (0/1,-1) (0/1,-1) (1/1,-1) (3/1,-1) (2/2,1) (0,0)

و هكذا يكون:

(0/3,6) = 100 110

(0/1,-1) = 00 0

يتم الحصول على الترميز المقابل للـ Runlength/Category من خلال جداول الخاصة التالية .
و كذلك بالنسبة للحصول على الـ Category للمركبة فمن جدول خاص بها.

JPEG coefficient coding categories

Range	DC Difference Category	AC Category
0	0	N/A
-1, 1	1	1
-3, -2, 2, 3	2	2
-7, ..., -4, 4, ..., 7	3	3
-15, ..., -8, 8, ..., 15	4	4
-31, ..., -16, 16, ..., 31	5	5
-63, ..., -32, 32, ..., 63	6	6
-127, ..., -64, 64, ..., 127	7	7
-255, ..., -128, 128, ..., 255	8	8
-511, ..., -256, 256, ..., 511	9	9
-1023, ..., -512, 512, ..., 1023	A	A
-2047, ..., -1024, 1024, ..., 2047	B	B
-4095, ..., -2048, 2048, ..., 4095	C	C
-8191, ..., -4096, 4096, ..., 8191	D	D
-16383, ..., -8192, 8192, ..., 16383	E	E
-32767, ..., -16384, 16384, ..., 32767	F	N/A

الجدول 1.2 يبين الـ Category لمركبات AC و DC

JPEG default DC code

Category	Base Code	Length	Category	Base Code	Length
0	010	3	6	1110	10
1	011	4	7	11110	12
2	100	5	8	111110	14
3	001	5	9	1111110	16
4	101	7	A	11111110	18
5	110	8	B	111111110	20

الجدول 2.2 يبين ترميز مركبات الـ DC

JPEG default AC code

Run/ Category	Base Code	Length	Run/ Category	Base Code	Length
0/0	1010 (= EOB)	4			
0/1	00	3	8/1	11111010	9
0/2	01	4	8/2	11111111000000	17
0/3	100	5	8/3	11111111011011	19
0/4	1011	8	8/4	111111110111010	20
0/5	11010	10	8/5	111111110111011	21
0/6	111000	12	8/6	111111110111010	22
0/7	1111000	14	8/7	111111110111011	23
0/8	1111100110	18	8/8	111111110111100	24
0/9	111111110000010	25	8/9	111111110111101	25
0/A	1111111110000011	26	8/A	111111110111110	26
1/1	1100	5	9/1	111111000	10
1/2	111001	8	9/2	111111110111111	18
1/3	1111001	10	9/3	1111111111000000	19
1/4	111110110	13	9/4	111111111100001	20
1/5	11111110110	16	9/5	1111111111000010	21
1/6	1111111110000100	22	9/6	1111111111000011	22
1/7	11111111110000101	23	9/7	11111111110000100	23
1/8	11111111110000110	24	9/8	11111111110000101	24
1/9	11111111110000111	25	9/9	11111111110000110	25
1/A	111111111100001000	26	9/A	11111111110000111	26
2/1	11011	6	A/1	111111001	10
2/2	11111000	10	A/2	1111111111001000	18
2/3	11111100111	13	A/3	1111111111001001	19
2/4	1111111110001001	20	A/4	1111111111001010	20
2/5	1111111110001010	21	A/5	1111111111001011	21
2/6	1111111110001011	22	A/6	1111111111001100	22
2/7	1111111110001100	23	A/7	1111111111001101	23

الجدول 3.2 يبين ترميز مركبات الـ AC

JPEG default AC code (continued...)

2/8	111111110001101	24	A/8	111111111001110	24
2/9	1111111110001110	25	A/9	1111111111001111	25
2/A	1111111110001111	26	A/A	111111111010000	26
3/1	111010	7	B/1	11111010	10
3/2	111110111	11	B/2	111111111010001	18
3/3	1111110111	14	B/3	111111111010019	19
3/4	111111110010000	20	B/4	111111111010011	20
3/5	1111111110010001	21	B/5	111111111010100	21
3/6	1111111110010010	22	B/6	111111111010101	22
3/7	1111111110010011	23	B/7	111111111010110	23
3/8	1111111110010100	24	B/8	111111111010111	24
3/9	1111111110010101	25	B/9	111111111011000	25
3/A	1111111110010110	26	B/A	111111111011001	26
4/1	111011	7	C/1	111111010	11
4/2	111111000	12	C/2	111111111011010	18
4/3	111111110010111	19	C/3	111111111011011	19
4/4	1111111110011000	20	C/4	111111111011100	20
4/5	1111111110011001	21	C/5	111111111011101	21
4/6	1111111110011010	22	C/6	111111111011110	22
4/7	1111111110011011	23	C/7	111111111011111	23
4/8	1111111110011100	24	C/8	111111111100000	24
4/9	1111111110011101	25	C/9	111111111100001	25
4/A	1111111110011110	26	C/A	111111111100010	26

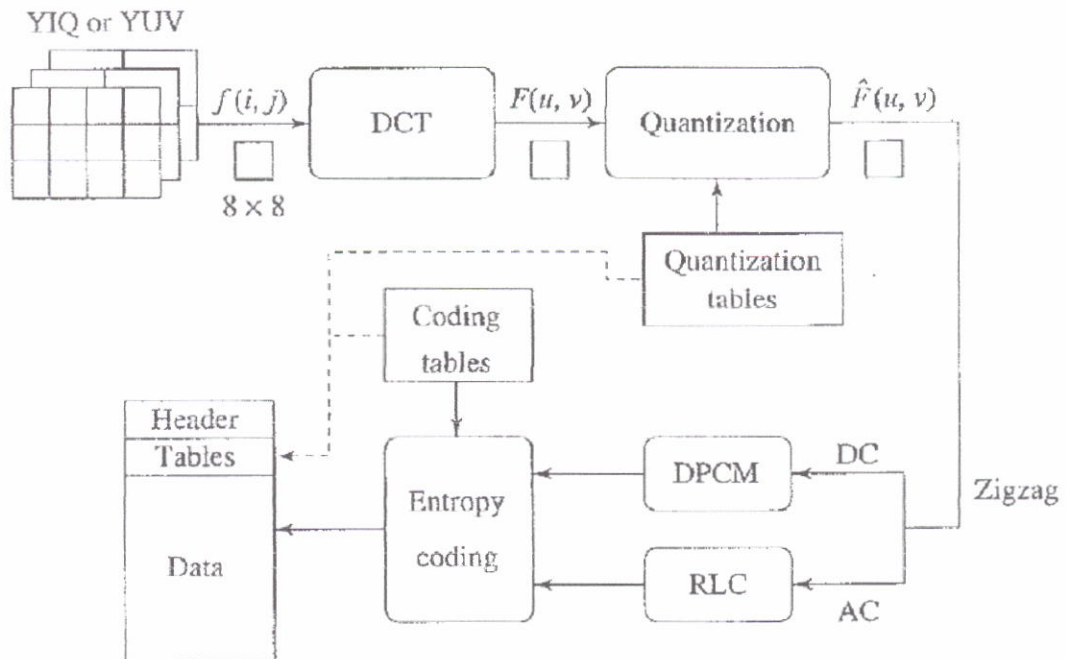
الجدول 4.2 يبين ترميز مركبات الـ AC

JPEG default AC code (continued...)

5/1	1111010	8	D/1	1111111010	12
5/2	1111111001	12	D/2	111111111100011	18
5/3	111111110011111	19	D/3	11111111111001000	19
5/4	1111111110100000	20	D/4	1111111111100101	20
5/5	1111111110100001	21	D/5	1111111111100110	21
5/6	1111111110100010	22	D/6	1111111111100111	22
5/7	1111111110100011	23	D/7	1111111111101000	23
5/8	1111111110100100	24	D/8	1111111111101001	24
5/9	1111111110100101	25	D/9	1111111111101010	25
5/A	1111111110100110	26	D/A	1111111111101011	26
6/1	1111011	8	E/1	11111110110	13
6/2	1111111000	13	E/2	111111111101100	18
6/3	111111110100111	19	E/3	1111111111101101	19
6/4	1111111110101000	20	E/4	1111111111101110	20
6/5	1111111110101001	21	E/5	1111111111101111	21
6/6	1111111110101010	22	E/6	1111111111110000	22
6/7	1111111110101011	23	E/7	1111111111110001	23
6/8	1111111110101010	24	E/8	1111111111110010	24
6/9	1111111110101011	25	E/9	1111111111110011	25
6/A	1111111110101110	26	E/A	1111111111110100	26
7/1	11111001	9	F/0	111111110111	12
7/2	1111111001	13	F/1	1111111111110101	17
7/3	1111111110101111	19	F/2	1111111111110110	18
7/4	1111111110110000	20	F/3	1111111111110111	19
7/5	1111111110110001	21	F/4	1111111111111000	20
7/6	1111111110110010	22	F/5	1111111111111001	21
7/7	1111111110110011	23	F/6	1111111111111010	22

الجدول 5.2 يبين ترميز مركبات الـ AC

الشكل التالي يبين خطوات خوارزمية الـ JPEG.



8.2 تطبيق لخوارزمية: (JPEG)

1- افرض لدينا المقطع التالي من صورة ما بحجم 8x8 بعد طرح 128 من كل قيمة.

124	125	122	120	122	119	117	118
121	121	120	119	119	120	120	118
126	124	123	122	121	121	120	120
124	124	125	125	126	125	124	124
127	127	128	129	130	128	127	125
143	142	143	142	140	139	139	139
150	148	152	152	152	152	150	151
156	159	158	155	158	158	157	156

قيم كبيرة

2- إن تطبيق التحويل DCT على المصفوفة السابقة يعطي المصفوفة التالية:

قيم صغيرة

Large values							
39.8756	6.5456	19.2222	1.2208	-0.3750	-1.0874	0.7834	1.1347
-103.4888	6.5675	0.2687	1.1206	0.3581	-0.6336	-1.0530	-0.4802
3.11906	0.8186	0.0740	0.2588	-1.5095	-2.2182	-0.1010	0.2329
-5.6746	0.8423	0.0060	-0.8132	1.4173	0.2212	-0.1393	0.1703
-3.3750	-1.7451	-1.7569	0.7764	-0.6250	-2.6597	1.5238	0.7620
5.9894	-1.1399	-0.4595	-0.7738	1.9994	0.2882	0.0529	0.1347
3.9733	6.5280	2.3990	-0.5589	-0.0518	0.0008	0.0000	0.0001
-3.4331	0.5198	-1.0721	0.8711	0.9634	0.0802	0.0006	0.0109
				Small values			

مصفوفة التكميم

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	50
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
40	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

$$\hat{F}(u, v) = \text{round} \left(\frac{F(u, v)}{Q(u, v)} \right)$$

3- مرحلة التكميم

مركبات الـ DCT

$$\text{round}\left(\frac{39.88}{16}\right) = 2$$

39.88	6.56	-2.24	1.22	-0.37	-1.08	0.79	1.13
-102.43	4.56	2.26	1.12	0.35	-0.63	-1.05	0.48
37.77	1.31	1.77	0.25	-1.50	-2.21	-0.10	0.23
-5.67	2.24	-1.32	-0.81	1.41	0.22	-0.13	0.17
-3.37	-0.74	-1.75	0.77	-0.62	-2.65	-1.30	0.76
5.98	-0.13	-0.45	-0.77	1.59	-0.26	1.46	0.00
3.97	5.52	2.39	-0.55	-0.051	-0.84	-0.52	-0.13
-3.43	0.51	-1.07	0.87	0.96	0.09	0.33	0.01

2	1	0	0	0	0	0	0
-9	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

المصفوفة بعد التكميم

4-تطبيق مسح Zigzag على المصفوفة بعد التكميم:

2	1	0	0	0	0	0	0
-9	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

و بالتالي تكون مركبة الـ DC هي DC=2 :

أما مركبات الـ AC فهي:

$$\text{AC} = (0, 1), (0, -9), (0, 3), \text{EOF}$$

5-ترميز مركبة الـ DC:

بفرض أن مركبة الـ DC للبلوك السابق كانت DC=-1 و بالتالي الفرق بينهما:

$$\text{Diff_DC} = 2 - (-1) = 3$$

DC=3 تقع في Category=2 بحسب الجدول السابق و بالتالي نرسل 100 و قيمة DC تحتاج

إلى 2 bits لترميزها أي 11 و بالتالي يكون الناتج النهائي لترميز المركبة هو:

100 11 = 5 bits.

6-ترميز مركبات ال:AC

AC = (0,1), (0,-9), (0,3), EOF

1 is in Category 1, Z/C = 0/1 (send 00), use 1 bit to specify "1" (001) 3 bits

-9 is in Category 4, Z/C = 0/4 (send 1011), use 4 bits to specify "-9" (1011 0110) 8 bits

3 is in Category 2, Z/C = 0/2 (send 01), use 2 bit to specify "3" (0111) 4 bits

EOF (send 1010) 4 bits

حيث أن Z هي عدد الأصفار و C هي الـ Category.
الترميز النهائي للبلوك يكون:

10011 001 1011 0110 0111 1010

معدل الضغط CR = 21.3

نلاحظ المعدل العالي للضغط.

ملاحظة: يتم تخزين جداول الترميز ضمن الصورة فيما يسمى بالـ Header من أجل استخدامها نفسها عند إعادة فك الضغط.

9.2 نماذج خوارزمية: (JPEG)

يدعم معيار الـ JPEG عدة أنواع من النماذج بعضها يستخدم بشكل واسع و أكثر انتشاراً من بعضها الآخر, أهم هذه النماذج هي:

- النموذج التسلسلي (Sequential Mode)
- النموذج التصاعدي (Progressive Mode)
- النموذج الهرمي (Hierarchical Mode)
- النموذج غير الفاقد للبيانات (Lossless Mode)

أولا النموذج التسلسلي:

و هو النموذج الافتراضي للخوارزمية, و فيه يتم ضغط كل الصور الرمادية أو الملونة عن طريق خط مسح واحد من اليسار لليمين أو من الأعلى للأسفل. و هو نفس النموذج الذي تم الحديث عنه في الفقرات السابقة.

ثانياً النموذج التصاعدي:

و هو مستخدم بشكل كبير في ضغط الصور في متصفحات الانترنت ,و يتم استخدام عدة خطوط مسح في هذا النموذج و هذا مفيد عندما تكون سرعة خط الاتصال بطيئة نوعاً ما .
إن عمليات المسح القليلة الأولى للصورة تؤدي إلى إعطاء صورة قريبة من الصورة الحقيقية نوعاً ما ,و لكن مع عمليات المسح الإضافية يتم استلام بيانات أكثر و بالتالي الحصول على جودة أفضل للصورة.

الشيء المفيد لهذه الطريقة هي أن المستخدم يمكنه التوقف عن عمليات المسح دون تحميل كامل بيانات الصورة أي دون الحصول على كامل الدقة الحقيقية للصورة.

ثانياً النموذج غير الفاقد:

و هو حالة خاصة من المعيار JPEG حيث أنه لا يعاني من أي ضياع في دقة الصورة .
يعتمد هذا النموذج على طريقة ترميز تفاضلية بسيطة .هذا النوع من الضغط نادراً ما يُستخدم لأنه لا يعطي أي معدل جيد للضغط.

هناك العديد من خوارزميات ضغط الصور منها JPEG 2000 التي تعطي معدل ضغط جيد و دقة عالية للصورة ,و الخوارزمية GIF و غيرها الكثير من الخوارزميات.

الفصل الثالث

جانب العملي

3.1 شرح الأدوات المستخدمة في البحث

1.1.3 واجهة دخول الى البرنامج كما في الشكل (1.3)

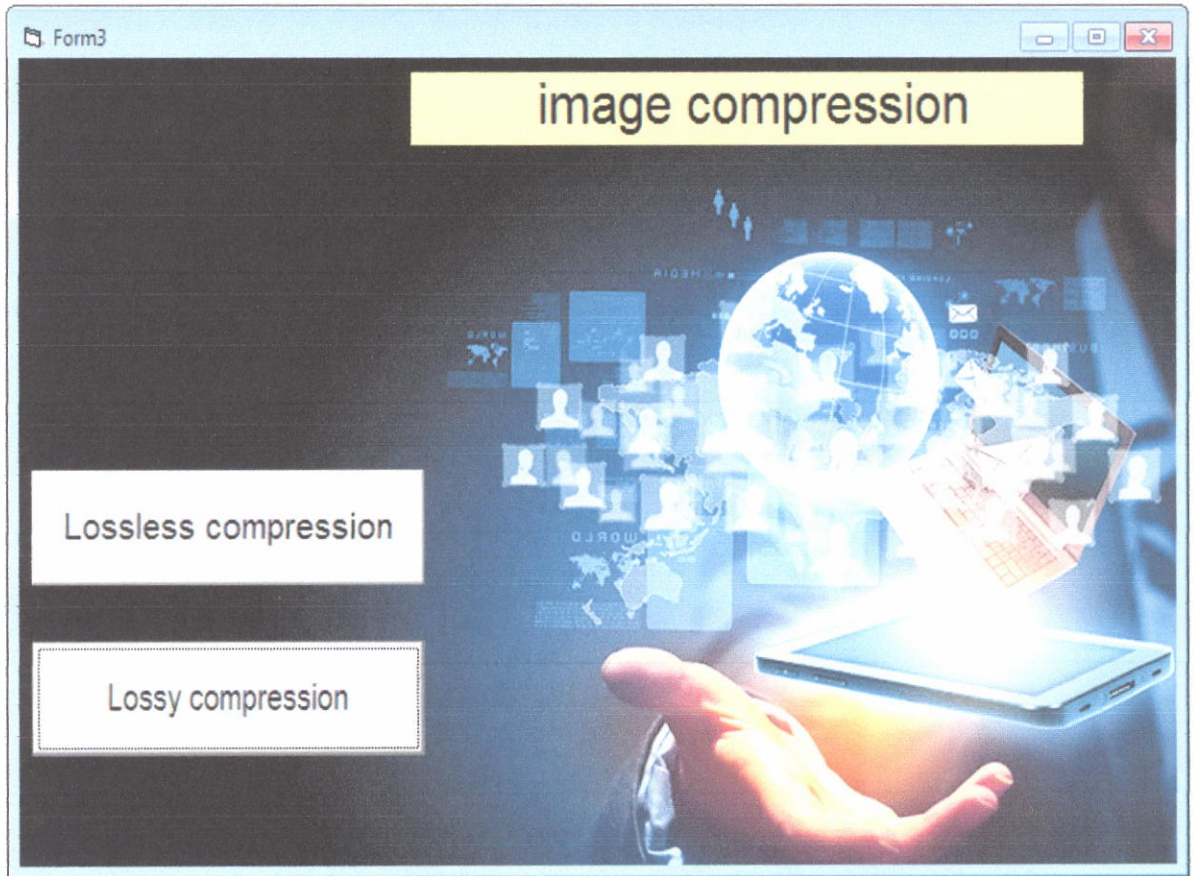
شكل(1.3)

في هذه الواجهة يتم حماية البرنامج من دخول الشخاص الغير مخولين بالدخول.

يقوم المستخدم بإدخال الاسم المستخدم و كلمة المرور الخاصة في البرنامج في حالة كتابة

خطا يقوم البرنامج بعرض رسالة تنبيه.

2.1.3 شاشة اختيار الطريقة الضغط كما في الشكل (2.3)

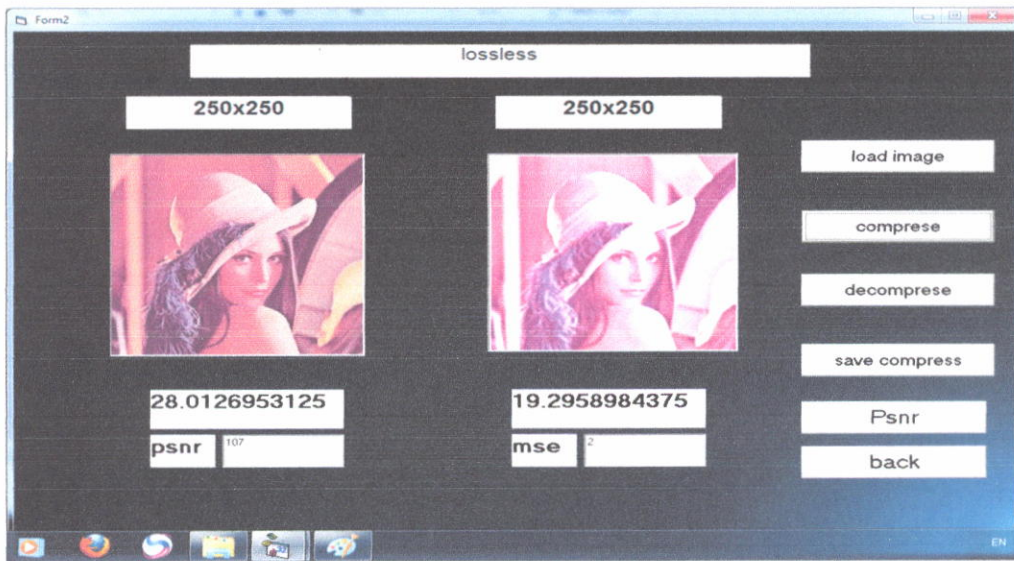


شكل (2.3)

في ذة الواجهه يتم اختيار الاحر الامرين اما ضغط الصورة بدون فقدان

اوضغطها بفقدان البيانات الصورة

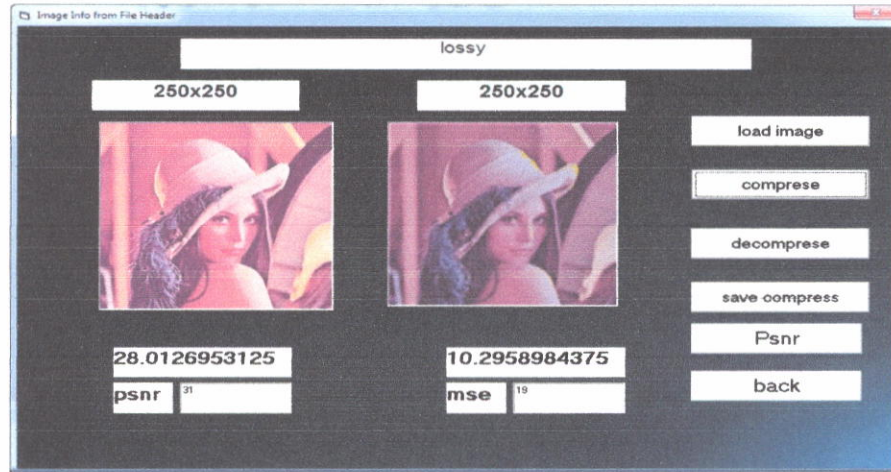
3.1.3 شاشة ضغط الصورة بدون فقدان البيانات



شكل (3.3)

في حال اختيار طريقة الضغط الصورة بدون فقدان يتم تحميل الصورة المراد ضغطها بعدها نضغط على ضغط الصورة يعرض الصورة الناتجة في المربع الثاني

4.1.3 شاشة طريقة ضغط الصورة بفقدان كما في الشكل (4.3)



شكل (4.3)

في حالة اختيار طريقة ضغط الصورة بفقدان نقوم بتحميل الصورة المراد ضغطها بعدها نضغط على ضغط الصورة يعرض الصورة الناتجة في المربع الثاني

2.3 المناقشة

عند استخدام طريقة الضغط بدون فقدان (lossless) فان حجم الصورة قبل الضغط هو (28kb) وبعد الضغط اصبح الناتج هو (kb19) اما نسبة الخطا بين الصورتين قبل الضغط وبعد الضغط هو $MSE = (2)$ ونسبة الإشارة للضجيج في الصورة هو $PSNR = (109)$ وكما مبين في الشكل ادناه



اما عند استخدام طريقة الضغط بفقدان (lossy) فان حجم الصورة قبل الضغط هو (28kb) وبعد الضغط اصبح الناتج هو (kb10) اما نسبة الخطا بين الصورتين

قبل الضغط وبعد الضغط هو $MSE = 19$ ونسبة الإشارة للضجيج في الصورة هو $(PSNR = 31)$ وكما مبين في الشكل ادناه

