دراسة وتطوير خوارزميّة لتحسين أنظمة التخزين عن طريق كشف الصور شبه المكرّرة باستخدام DCT

م. حسن علي حسن

قسم هندسة التحكم الآلي والحواسيب

كلية الهندسة الميكانيكيّة والكهربائيّة

جامعة البعث

إشراف: أ.د. عمار زقزوق

ملخّص البحث

نقدّم في هذا البحث تقنيّة لكشف التشابه بين الصور الرقميّة، والذي يتمثّل بضغط الصورة، تغيير سطوعها، تباينها، إشباع الألوان فيها، الخ. ومن ثمّ حذف الصور الأقل دقّة وذلك لتوفير مساحة تخزينيّة في النظام. يتم ذلك بالاعتماد على تحويل جيب التمام المتقطّع DCT(Discrete Cosine Transform) ، والذي ينتج مصفوفة تمثّل تردّدات الصورة، إذ تمثّل التردّدات المنخفضة التفاصيل العامّة للصورة والتردّدات المرتفعة تمثّل حواف الصورة. إنّ أي عمليّة تغيير في الصورة توثّر على التردّدات المرتفعة أي حواف الصورة (عناصر الصورة التي يتغيّر اللون عندها)، وقد تتأثّر تفاصيل الصورة في حال كان التغيير الحاصل كبيراً.

سنعتمد في عمليّة الكشف على إنشاء مفتاح اختزال لكل صورة. يعتمد هذا المفتاح على 64 عنصر من مصفوفة DCT، والتي تمثّل التردّدات الأقل في الصورة. ومن ثم تطبيق شيفرة هامينغ على مفاتيح الاختزال لزيادة نسبة التشابه. أظهرت هذه الخوارزميّة فعاليّة في كشف تشابه الصور ما دام التعديل الحاصل على الصور لا يؤثّر على 63 عنصر من الصورة التي تجسّد أهم تفاصيل الصورة.

كلمات مفتاحيّة: معالجة الصورة، حذف البيانات المكرّرة، الصور شبه المكرّرة، التخزين السحابي، الأنظمة العنقوديّة (Hadoop)،DCT.

Study and development an algorithm to improve storage systems by detecting near-duplicate images using DCT

Eng. Hasan Ali Hasan
Automatic Control and Computers Engineering Department
Mechanical and Electrical Engineering Faculty
Albaath University
Supervision: Dr. Eng. Ammar Zakzouk

Abstract

In this paper, we present a technique for detecting the similarity between digital images, which is represented by compressing the image, changing its brightness, its contrast, its saturation. etc., and then deleting the less accurate images in order to save storage space in the system. Our technique is based on (Discrete Cosine Transform) DCT, which produces an array that represents the image frequencies. Lower frequencies represent the general details of the image, while high frequencies represent the edges of the image. Any change in the image affects mainly the higher frequencies, i.e. the edges of the image (pixels at which the color changes). Image details may be affected if the change is significant.
In the detection process, we will rely on creating a hash key for each image. This key is based on 64 elements of the DCT array, which represent the lowest frequencies in the image. Then apply Hamming code to hash keys to increase similarity. This algorithm has shown to be effective in detecting similarity of images as long as the

modification made to the images does not affect the 63 pixels of the image that embody the most important details of the image.

Keywords: Image processing, Data deduplication, similarities images, Cloud storage, Cluster systems (Hadoop), DCT.

1- المقدّمة

تعتبر الملفّات المكرّرة من الأسباب التي تؤدّي إلى هدر في المساحة التخزينيّة، ويسمّى الحيّز الذي تشغله هذه الملفّات بالمساحة الضائعة، ولذلك تعتبر البيانات المكرّرة بالبيانات غير المفيدة، وتزداد هذه المساحة بزيادة الملفّات المكرّرة ممّا يصعّب من إدارة مساحة التخزين وتنظيم البيانات. استخدمت خوارزميّات لضغط الملفّات وذلك لتقليل حجمها وتوفير مساحة تخزينيّة. كما تم تطوير خوارزميّات تساعد في كشف التطابق بين البيانات المكرّرة. من أهم هذه الخوارزميّات هي خوارزميّات التجزئة والتي تعتمد على البيانات الثنائيّة للملفّات، ومن ثم توليد مفتاح مختزل لكل ملفّ، بعدها تتم المقارنة بين الملفّات لكشف التطابق. لكن أي تغيير طفيف في الملفّ يؤدّي إلى توليد مفتاح مختزل مختلف، فعلى سبيل المثال إذا كان لدينا صورة وقمنا بتغيير عنصر واحد فقط، فيؤدّي ذلك إلى الحصول على ملفّ جديد مشابه للملفّ الأصليّ وليس مماثل له، وبالتالي توليد مفتاح مختزل مغاير. وبذلك تفشل جميع خوارزميّات التجزئة عند أقل تغيير في الملفّ. إذاً سنطلق اسم الصور شبه المكرّرة على الصور المعدّلة (كتغيير عدد عناصر الصورة وتعديل سطوعها وتباينها وغيرها...).

1.1- نظام التخزين السحابي (Cloud Storage)

أحد نماذج الحوسبة السحابيّة التي يتم من خلالها تخزين البيانات والملفّات على الإنترنت، والذي يتمثّل بخوادم ذات مواصفات كبيرة تحتوي على مساحة تخزينيّة ضخمة،

بحيث تمكّن المستخدمين من الاحتفاظ بملفّاتهم على الإنترنت بدلاً من محرّكات الأقراص محدودة السعة.

تستخدم خدمة التخزين السحابي كجهاز تخزين إضافي في حال استخدام محرّكات أقراص ذات سعات تخزينيّة صغيرة، كما توفّر إمكانيّة الرجوع إلى البيانات التي تم تخزينها في أي مكان وأي وقت من خلال تحميلها من الإنترنت. كما ويمكن مشاركة هذه الملفّات مع الآخرين، بالإضافة إلى إمكانيّة الاحتفاظ بالملفّات المهمّة آمنةً بوساطة كلمة سر ونظام تشفير [1].

2.1- نظام التخزين العنقودي (Hadoop)

هو منصّة لتخزين ومعالجة البيانات الضخمة (Big Data)، يعتمد هذا النظام على التوزّع في التخزين والمعالجة، إذ يتم تخزين البيانات على أجهزة حاسوبيّة عدّة، وتوزّع عمليّة معالجة البيانات على هذه الأجهزة لتسريع عمليّة المعالجة. فمع ازدياد كميّة البيانات وتنوّع مصادرها وسرعة تدفّقها تفشل قواعد البيانات التقليديّة في هيكلة وتخزين وتحليل هذه البيانات، وبالتالي لا بدّ من نظام يمكّننا من التعامل مع هذه البيانات. يتكوّن نظام Hadoop من عقدة رئيسيّة (Name node) وعقد ثانويّة (Data nodes). مهمّة العقدة الرئيسيّة تتمثّل في إدارة تخزين البيانات مثل معرفة السعة التخزينيّة المتوفّرة في العقد الثانويّة، أسماء الملفّات والمجلّدات المخزّنة في نظام التخزين الموزّع وغيرها...، أمّا تخزين البيانات ومعالجتها يتم في العقد الثانويّة [2].

يستخدم Hadoop على نطاق واسع من قبل الشركات الكبرى، وذلك لتخزين البيانات الضخمة وتحليلها وتشغيل العمليّات الحسابيّة الكبيرة الموزّعة، ومن هذه الشركات: Facebook, Yahoo, Amazoon, Google.

3.1- ضغط الصورة

يعرّف ضغط الصورة على أنه عمليّة لتقليل بيانات الصورة (تقليل حجم الصورة) بغرض توفير مساحة تخزينيّة أو تحميل الصورة. فمواقع الويب التي تحتوي صوراً غير مضغوطة تتطلّب وقتاً أكبر للتحميل منها، مما يؤثّر على زيارة المستخدمين لهذه المواقع، أو إرسال الصورة. فلدى بعض خوادم البريد الالكتروني حدّاً لحجم الملفّ، ويتم ذلك إمّا بحذف عناصر من الصورة، أو تجميع عدد من العناصر معاً، أو تقليل عدد الخانات (bits) اللازمة لتمثيل كل عنصر في الصورة.

- حذف عناصر من الصورة أو تجميع عدد منها معاً: وبذلك يتم بتقليل عدد الخانات اللازمة لتمثيل الصورة.

- تقليل عدد الخانات اللازمة لتمثيل كل عنصر: أو ما يسمّى (bit depth)، فعدد الخانات اللازمة لمثيل العنصر في الصورة الملوّنة هو 24 bit، بينما في الصورة الرماديّة 8 bit، والثنائيّة 1 bit.

تقسم تقنيّات الضغط بشكل عام إلى قسمين: Lossless compression، Lossy compression ويبيّن الجدول (1) خصائص كل تقنيّة:

Lossless	Loosy
لا تؤدّي عمليّة الضغط إلى فقدان المعلومات	تؤدّي عمليّة الضغط إلى فقدان المعلومات
يتم تقليل حجم الملفّ مع الحفاظ على جودة الملفّ نفسها قبل ضغطه	يتم تقليل حجم الملفّ بشكل أكبر من lossless ولكن على حساب جودة الملفّ المضغوط
يمكن استرجاع الملفّات الأصليّة بدقة بعد ضغطها	لا يمكن إعادة بناء الملفّ الأصليّ بعد ضغطه

الجدول (1): تقنيّات الضغط

- **بارامترات عمليّة الضغط:** تقسم إلى قسمين:

1- معدل الضغط (Compression Ratio): يعطى بالعلاقة:

$$Compression\ Ratio = \frac{\text{Uncompressed file size (bytes)}}{\text{compressed file size (bytes)}} \qquad (1)$$

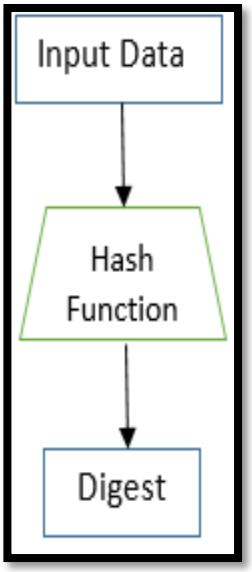
2- عدد الخانات لكل عنصر (Bits Per Pixel): يعطى بالعلاقة:

$$Bits\ Per\ Pixel = \frac{\text{Number of bits}}{\text{Number of pixels}} \qquad (2)$$

هذه العلاقة تعبّر عن عدد الخانات التي يتكوّن منها عنصر الصورة. يتم مقارنة هذا البارامتر قبل وبعد عمليّة الضغط. إذ تقلّ عدد الخانات الممثّلة لعنصر الصورة بعد عمليّة الضغط [3].

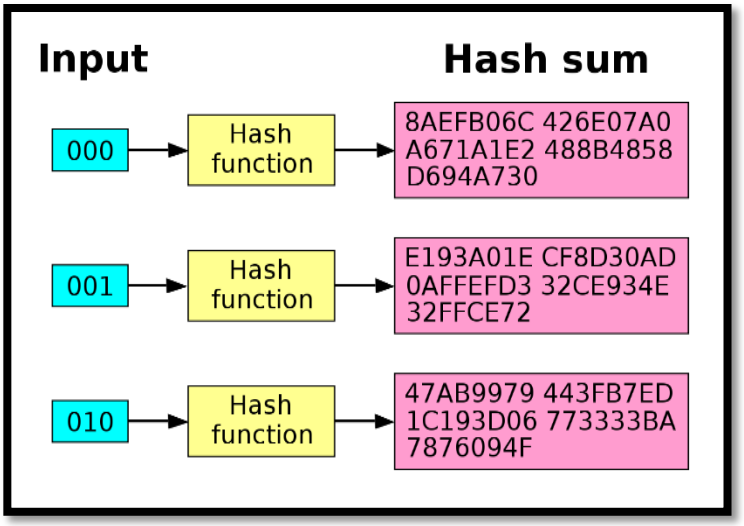
4.1- خوارزميّات التجزئة (Hash Algorithms)

خوارزميّات التجزئة هي خوارزميّات رياضيّة تحوّل البيانات ذات الحجم العشوائي إلى تجزئة بحجم ثابت، فهي تولّد قيمة ثابتة الطول من إدخال معين، تسمّى هذه القيمة قيمة التجزئة (hash value). قيمة التجزئة هي ملخّص للبيانات الأصليّة. والشكل (1) يبين عمل خوارزميّة التجزئة (تابع التجزئة):



الشكل (1): تابع التجزئة

إن عمليّة التجزئة هي عمليّة أحاديّة الاتّجاه أي غير قابلة للانعكاس، ممّا يعني أنه لا يمكن استرداد دفق البيانات الأصليّ من ملخّص الرسالة. جميع ملخّصات الرسالة أو قيم التجزئة التي تم إنشاؤها بوساطة دالة التجزئة المعيّنة لها الحجم نفسه بغضّ النظر عن حجم قيمة الإدخال. يعتمد حجم قيمة التجزئة على الخوارزميّة المستخدمة، فعندما يتم استخدام الخوارزميّة نفسها وبيانات الادخال نفسها يكون الناتج هو قيمة التجزئة نفسها على الدوام. ويبيّن الشكل (2) العلاقة بين بيانات الدخل وقيمة التجزئة:



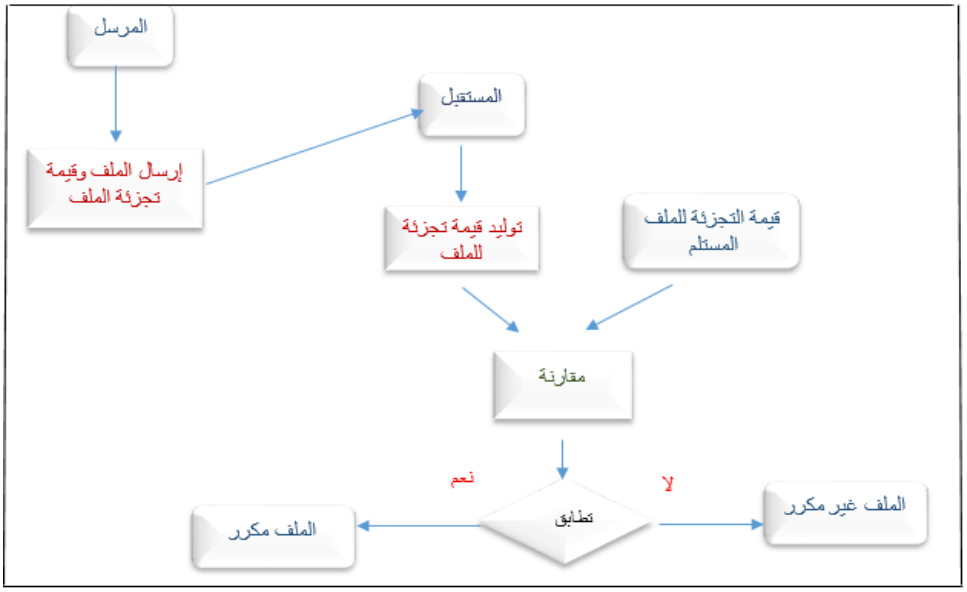
الشكل (2): العلاقة بين بيانات الدخل وقيمة التجزئة

1.4.1- خوارزميّات التجزئة الأكثر شيوعاً

تعتبر MD5(Message Digest 5)، SHA-1(Secure Hash Algorithm 1) خوارزميّات التجزئة الأكثر استخداماً. تختلف جميع خوارزميّات التجزئة بشكل عام عن بعضها بحجم بيانات الدخل وطول مفتاح الدخل، لكن MD5,SHA-1 تشتركان بحجم بيانات الدخل، إذ تتعامل هاتين الخوارزميّتين مع بيانات بحجم (512 bit) وفي حال كانت الرسائل أكبر تقسّم إلى رسائل صغيرة. إنّ الحد الأعظمي من البيانات التي يمكن للخوارزميّتين التعامل معها هو 2^{64} bit، ولكن تختلفان بطول مفتاح التجزئة الناتج، فخوارزميّة MD5 تنتج قيمة تجزئة بطول (128 bit)، بينما خوارزميّة SHA-1 تنتج قيمة تجزئة بطول (160 bit). أمان الخوارزميّة وسرعتها تتعلّقان بشكل أساسي بطول قيمة التجزئة، لذلك SHA-1 أكثر أماناً من MD5، بينما MD5 أكثر سرعة [4].

2.4.1- استخدامات خوارزميّات التجزئة

تُستخدم وظائف تجزئة التشفير على نطاق واسع في تكنولوجيا المعلومات. يمكننا استخدامها للتوقيعات الرقميّة ورموز مصادقة الرسائل وأشكال أخرى من المصادقة. يمكننا أيضاً استخدامها لفهرسة البيانات في جداول التجزئة، وبصمات الأصابع، وتحديد الملفّات، والكشف عن التكرارات، وتخزين كلمات المرور. ويوضّح الشكل (3) أهميّة هذه الخوارزميّات في مصادقة الرسائل وكشف التكرار:



الشكل (3): دور خوارزميّات التجزئة في مصادقة الرسائل وكشف التكرار

وبالتالي باستخدام مفاتيح التجزئة للملفّات يمكن كشف المطابقة بينها، وتوفير مساحة تخزينيّة حسب نسبة الملفّات المكرّرة المكتشفة.

5.1- شيفرة هامينغ (Hamming Code)

في أواخر الأربعينيّات من القرن الماضي، أدرك ريتشارد هامينغ أن التطوّر الإضافي لأجهزة الكمبيوتر يتطلّب موثوقيّة أكبر، لا سيّما القدرة على اكتشاف الأخطاء وتصحيحها. (ففي ذلك الوقت، كان التحقّق من التكافؤ مستخدماً لاكتشاف الأخطاء،

ولكنّه لم يكن قادراً على تصحيح أي أخطاء). فأوجد هامينغ مجموعة الرموز كان بمثابة بداية لنظريّة الترميز. إذ استخدمت هذه الرموز لإدراج معلومات تصحيح الخطأ في تدفّقات البيانات، بحيث يتم اكتشاف الخطأ وتصحيحه. تؤدّي إضافة هذه المعلومات إلى زيادة كميّة البيانات، إلّا أنها تزيد من موثوقية الاتّصالات في وسائط الاتصال ذات معدّلات الخطأ العالية.

1.5.1- مبدأ عمل شيفرة هامينغ

طريقة هامينغ تعتمد على إضافة خانة إضافية تسمّى الخانة المساعدة على الشيفرة المرسلة في المواقع من رتبة 2^n حيث n=0,1,2,... وذلك حسب طول البيانات المرسلة، فإذا قام المرسل بإرسال بيانات بطول (4 bit) فيتم إضافة الخانة المساعدة في المواقع 1،2،4، ويصبح طول البيانات المرسلة (7 bit). وإذا كان طول البيانات المرسلة (8 bit) يتم إضافة الخانة المساعدة في المواقع 1،2،4،8، وبالتالي يصبح طول البيانات المرسلة (12 bit). قيمة الخانة المساعدة تحسب بإنجاز عمليّة (XOR) بين أول موقع يحمل القيمة (1 bit) مع البيانات المرسلة، ثم تتم إضافة الخانة المساعدة إلى الرسالة، بعدها يتم الإرسال إلى المستقبل والذي بدوره يقوم بعمل بوابة (XOR) بين أول خانتين تحملان القيمة (1 bit) بحيث لا تكون الخانة من رتبة 2^n، ومن ثم عمليّة (XOR) بين الناتج والخانة المساعدة فإذا كان الجواب (0) لا يوجد خطأ في استقبال البيانات [11].

6.1- تحويل جيب التمام المتقطّع (Discrete Cosine Transform ((DCT)

هو أحد مراحل ضغط الصور باستخدام خوارزميّة JPEG. يعتبر هذه التحويل مشابه جدّاً لتحويل فوريه، إذ يقومان بنقل الصورة من المجال الفراغي(المكاني) إلى المجال التردّدي. لكن تحويل فوريه يستخدم علاقات رياضيّة معقّدة تتطلّب مدّة زمنيّة كبيرة، فهو يتعامل

مع الأعداد المعقّدة (complex) أي الأعداد الحقيقيّة والتخيليّة. بينما تحويل جيب التمام المنفصل يستخدم علاقات رياضيّة بسيطة ذات فترة تنفيذ قصيرة، فهو يتعامل فقط مع الأعداد الحقيقيّة ممّا يجعله أكثر استخداماً في عمليّات ضغط الصورة. باستخدام خوارزميّة JPEG يتم فصل الصورة إلى أجزاء أو نطاقات فرعيّة مستقلّة تختلف هذه الأجزاء من حيث الأهميّة، إذ يتم تقسيم الصورة إلى كتل (blocks) بحجم 8*8، ويتم تطبيق تحويل DCT على كل كتلة. مصفوفة التحويل الناتجة تمتاز بأنها تمتلك القيمة الأعظم في الزاوية العليا اليسرى وتقل قيم المعاملات باتجاه اليمين والأسفل. تمثّل المعاملات القريبة من الزاوية العليا اليسرى مركّبات التردّد المنخفض، في حين تمثّل المعاملات الباقية مركّبات التردّد المرتفع [6].

رياضياً، يعرّف تحويل جيب التمام المتقطّع بالعلاقة (3):

$$F(u,v) = \frac{c(u)\,c(v)}{4}\sum_{i=0}^{7}\sum_{j=0}^{7}\cos\frac{(2i+1)u\pi}{16}\cos\frac{(2j+1)v\pi}{16}\,f(i,j) \qquad (3)$$

حيث:

f : المصفوفة الجزئيّة والتي أبعادها 8*8.

F : *المصفوفة الناتجة عن تطبيق* DCT *على المصفوفة* f.

$f(i,j)$: العنصر الموجود في السطر i والعمود j.

$F(u,v)$: العنصر الموجود في السطر u والعمود v.

I,j,u,v=0,1,...,7

$c(u), c(v)$: ثوابت تأخذ القيم التالية:

$$c(u), c(v) = 1\sqrt{2} \quad for\ u,v = 0$$
$$c(u), c(v) = 1 \quad for\ u,v \neq 0$$

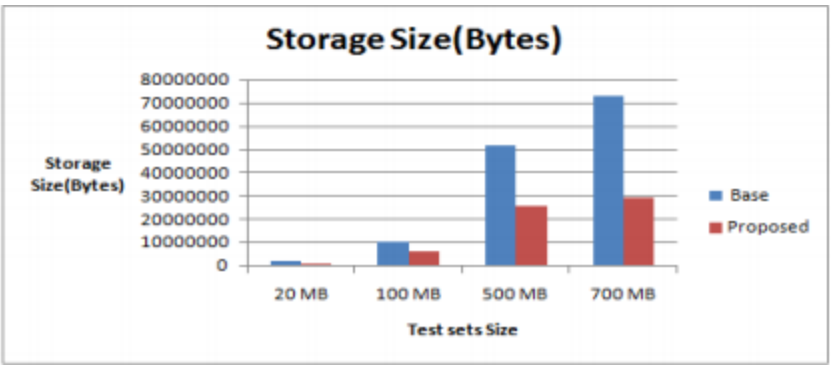
2- هدف البحث

يهدف هذا البحث إلى كشف التكرار والتشابه بين مجموعات الصور الموجودة في نظام التخزين. فالصور الرقميّة تعتبر إحدى موارد أنظمة التخزين، فعند استقبال أنظمة التخزين لهذا النوع من البيانات قد تتكوّن مجموعات هذه الصور من أنواع عدّة، فقد تكون هذه الصور مكرّرة أو قد تتعرّض للضجيج أو للضغط أو تنخفض دقّتها بسبب إرسالها. بالتالي لا بدّ من كشف التكرار والتشابه بين هذه الصور لتخزين البيانات الفريدة والمفيدة منها، وذلك من أجل الاستثمار الأمثل لمساحة التخزين، والذي بدوره يساعد في سهولة تنظيم البيانات وإدارتها، وبالتالي تحسين أداء النظام ككل.

3- الدراسات السابقة

سنتناول في الدراسات المرجعية التقنيّات المستخدمة في توفير المساحة التخزينيّة:

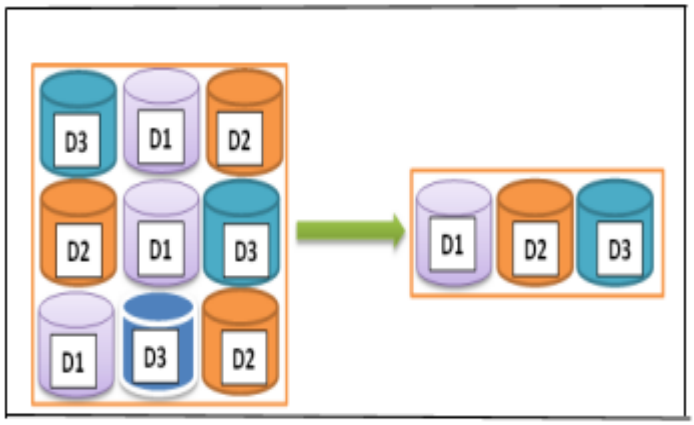
📖 **Karambi،Tannu** : قدّم هذان الباحثان تقنيّة فعّالة وآمنة للتخزين السحابي، تعتمد بفعاليّتها على استخدام خوارزميّة SHA-512 في كشف التطابق بين البيانات. فعندما يقوم مستخدم ما بتحميل البيانات إلى الخادم، يتم حساب قيمة التجزئة للملفّ باستخدام خوارزميّة التجزئة SHA-512، فإذا حصل تطابق مع أحد مفاتيح الاختزال الخاصّة بالملفّات المخزنة في الخادم لا يتم تخزين الملفّ. وإذا لم يحصل تطابق يتم تخزين الملفّ في الخادم، والاحتفاظ بقيمة التجزئة الخاصّة به [5]. يبين الشكل (4) المساحة التخزينيّة التي تم توفيرها باستخدام الخوارزميّة المقترحة:



الشكل (4): تحسين مساحة التخزين

حيث يمثّل المخطّط باللون الأزرق طريقة التخزين التقليديّة في الخادم، أما المخطّط باللون الأحمر يمثّل تخزين البيانات بدون تكرارها مما سمح بالحفاظ على مساحة تخزينيّة كبيرة.

📖 **Pronika, S.S.Tyagi:** قدّم هذا البحث طريقة لإلغاء البيانات المكرّرة في الأنظمة السحابيّة، تعتمد على تقسيم الملفّ إلى أجزاء ، ومن ثم توليد مفتاح اختزال لكل جزء باستخدام إحدى خوارزميّات التجزئة (MD5,SHA−1,Rabin). بعد ذلك يتم مقارنة مفتاح الاختزال لكل كتلة بالمفاتيح الخاصّة بالكتل المخزنة في نظام التخزين، فإذا لم يحصل تطابق يتم تخزين الكتلة الجديدة. وبالتالي تعتبر هذه الطريقة ذات فعاليّة أكبر من الخوارزميّة على مستوى الملفّ لكنها تستغرق وقتاً أطولاً [7]. يوضّح الشكل (5) التقنيّة المتّبعة في هذه الدراسة:



الشكل (5): إلغاء البيانات المكرّرة على مستوى الكتلة

📖**Shamsher Singh, Ravinder Singh**: قدّم هذان الباحثان تقنيّة لإلغاء البيانات المكرّرة عند القيام بالنسخ الاحتياطي للملفّات. إذ يتكوّن النظام السحابي من خادم النسخ الاحتياطي والذي مهمّته التواصل مع المستخدم للقيام بإضافة أو قراءة بيانات من السحابة، وعقدة تخزين رئيسية (primarynode)، وعقد تخزين ثانوية عدّة

(datanodes). تعتمد هذه التقنيّة بشكل أساسي على مكان إلغاء البيانات المكرّرة، فإذا كان حجم البيانات أقل من (1 GB) تتم العمليّة على مستوى العقدة الرئيسية، أي يتم مقارنة مفتاح الاختزال للكتلة قبل تخزينها في العقد الثانوية. وإذا كان حجم البيانات أكبر من (1 GB) تتم العمليّة على مستوى العقد الثانوية [8]. وممّا سبق نلاحظ أن هذه التقنيّة تعتمد على تسريع عمليّة المعالجة في حال كان حجم الملفّ أقل من (1 GB)، لأنّ عمليّة حذف البيانات المكرّرة تتم دون الحاجة لتخزين البيانات في العقد الثانويّة، ومن ثمّ حذف البيانات المكرّرة، مما يؤدّي لتقليل الزمن المستغرق لإنجاز العمليّة.

Naresh Kumar واخرون: قدّموا طريقة لإلغاء البيانات المكرّرة المخزنة في Hadoop على مستوى الكتل باستخدام خوارزميّة MD5، يتم تخزين مفاتيح الاختزال لكل الكتل المخزنة في Hadoop ضمن buckets. فعندما يتم تخزين ملفّ في نظام التخزين الموزع يتم تقسيم الملفّ إلى كتل، وتوليد قيمة مفتاح MD5 لكل كتلة، ومقارنة هذه القيمة مع مفاتيح الاختزال المخزنة في النظام، فإذا حصل تطابق لا يتم تخزين الكتلة، وإذا لم يحصل تطابق يتم تخزين الكتلة في نظام التخزين، وتخزين قيمة المفتاح الخاصّة بالكتلة فيbuckets [9]. تعتبر التقنيّة المستخدمة في هذه الدراسة فعّالة بدرجة أكبر في التعامل مع البيانات المكرّرة بالمقارنة مع الدراسات السابقة في حال كان التعامل يتم مع البيانات الضخمة، لأنّ Hadoop يمثل بيئة تخزينيّة ملائمة للتعامل مع البيانات الضخمة.

Shradha kadam، واخرون: قدّموا تقنيّة لكشف الصور شبه المكرّرة في أنظمة التخزين، تعتمد عمليّة المقارنة على حساب مسافة هامينغ بين المفتاحين المختزلين للصورتين المقارنتين، فإذا كانت نسبة المطابقة تتجاوز 97% فالصورتين متشابهتين، توليد المفتاح لكل صورة يتم عن طريق الخطوات التالية:

1- تقليل حجم الصورة.

2- تحويل الصورة الملوّنة إلى رماديّة.

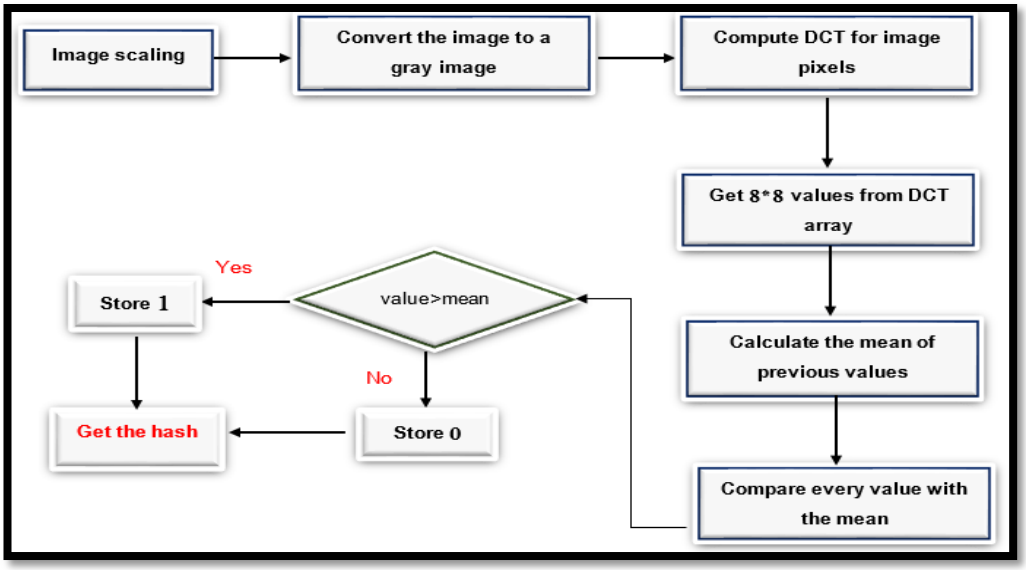
3- حساب متوسّط قيم العناصر للصورة الناتجة.

4- مقارنة قيمة كل عنصر في الصورة مع المتوسّط، فاذا كانت قيمة العنصر أكبر من المتوسّط يتم وضع القيمة 1 في المفتاح وإلّا يتم وضع القيمة 0، وهكذا حتى يتم توليد المفتاح [10].

بالمقارنة مع الدراسات السابقة، في حال كانت البيانات تحتوي صوراً مكرّرةً فقط، فهذه التقنيّة تستغرق وقتاً أطولاً في عملية كشف التكرار. أما في حال كانت البيانات تحتوي صوراً متشابهة، فتظهر هذه الخوارزميّة فعاليّة أكبر من جميع خوارزميات الاختزال المستخدمة سابقاً والتي تفشل عند أقل تغيير قد يحصل على الصورة.

4- طريقة البحث

يتم توليد مفتاح (hash) للصور، ومن ثم تتم عمليّة المقارنة بين الصور بالاعتماد على المفتاح لكشف تكرار وتشابه الصور. ويبيّن الشكل (6) الخوارزميّة المتّبعة في توليد المفتاح:



الشكل (6): خوارزميّة توليد مفتاح الاختزال للصور

– الخطوة الأولى تعتمد على تقليل عدد عناصر الصورة إلى 32*32 عنصر، بحيث تمثّل العناصر الناتجة التفاصيل الأهم في الصورة.

– الخطوة الثانية تعتمد على تحويل الصورة الملوّنة إلى رماديّة، وذلك لتقليل التعقيد لأنّ التعامل مع عنصر ذو بعد واحد أي يأخذ القيمة من 0 حتى 255 أسهل من التعامل مع عنصر ذو ثلاث أبعاد متمثّلة بالألوان الأحمر والأخضر والأزرق، وكل بعد يأخذ قيمة من 0 حتى 255. وبتحويل الصورة إلى رماديّة يمكننا التعامل مع تفاصيل الصورة وحوافها وسطوعها وإشباعها وغيرها من العمليّات ما عدا اللون.

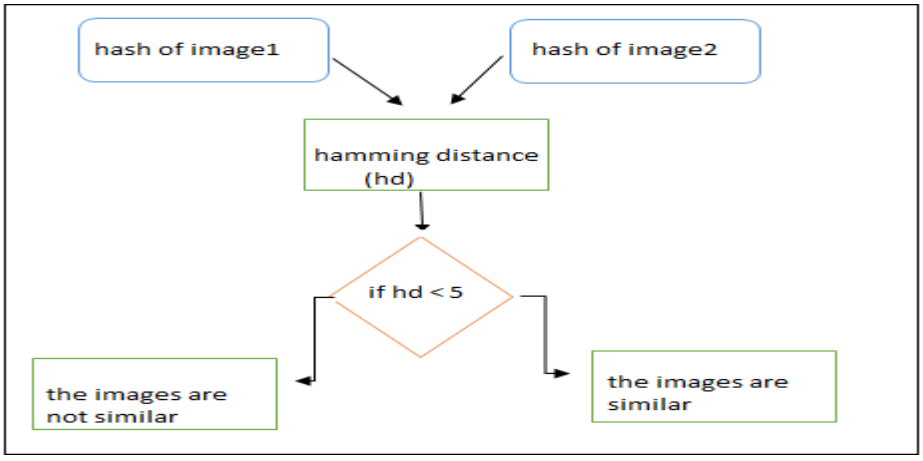
– في الخطوة الثالثة يتم حساب مصفوفة DCT للعناصر 32*32. وبالتالي تنتج مصفوفة تمثّل معاملات DCT. تكون قيمة المعاملات كبيرة في الزاوية العليا واليسرى من المصفوفة وتقل باتجاه الأسفل واليمين.

– في الخطوة الرابعة نقوم بأخذ القيم من العنصر (0،0) حتى العنصر (8،8) أي 64 عنصر الأولى من مصفوفة DCT والتي تمثّل الترددات الأقل في الصورة، وبالتالي

العناصر التي تمثّل التفاصيل الأدقّ في الصورة. بعد ذلك نقوم بالتعريف عن قيمتين الأولى نسمّيها (Dct_Average) والتي تمثّل متوسّط هذه القيم، والثانية نسمّيها (hash) تمثّل نتيجة المقارنة بين القيم المأخوذة وDct_Average. ثم نقوم بمقارنة كل قيمة من القيم مع المتوسّط، فإذا كانت قيمة معامل DCT أكبر من المتوسّط يوضع في hash القيمة 1 وإلّا توضع القيمة 0، وهكذا حتى تتم مقارنة جميع القيم مع Dct_Average. في نهاية عملية المقارنة ينتج مفتاح hash مكوّن من 64 رقم من الأصفار والواحدات، والذي يمثّل المفتاح الخاص بالصورة.

*** عمليّة المقارنة:**

بعد تشكيل المفتاح الخاص بكل صورة نقوم بمقارنة الصور باستخدام Hamming Distance فإذا كان الاختلاف في (1 byte) أي أقل من (5 bit) نقول أنّ الصورتان متشابهتان، وذلك من أجل زيادة نسبة التشابه بين الصور. فقد يتم التعديل على الصورة بحيث يختلف مفتاح الصورة الناتجة عن المفتاح الخاص للصورة الأصليّة في خانة واحدة فقط. ويبين الشكل (7) عملية المقارنة بين الصور شبه المكرّرة:



الشكل (7): كشف الصور شبه المكرّرة

5- النتائج ومناقشتها

تمّ إجراء بعض عمليّات الصورة الرقميّة على صورة ذات أبعاد 768*1024 عنصر، ونتج عن هذه العمليّات نسخ معدّلة عن هذه الصورة. إذ تتمثّل مجموعة البيانات من صورة محدّدة ومجموعة من الصور الناتجة عن إجراء تعديلات على هذه الصورة. ومن ثم تطبيق الخوارزميّة المقترحة على هذه المجموعة من الصور. وفيما يلي العمليّات الي تمّ تنفيذها على الصورة:

1. تغيير أبعاد الصورة: تمّ إنشاء نسخة من الصورة بزيادة عدد عناصر الصورة، إذ قمنا بتكبير الصورة 8 مرات لتصبح أبعاد الصورة 8192*6144، وبالتالي الحصول على صورة مغايرة تماماً ولكن تفاصيل الصورة نفسها، ونسخة أخرى بتصغير أبعاد الصورة، حيث قمنا بتقسيم العرض والارتفاع على القيمة 16 لتصبح أبعاد الصورة 48*64 عنصر.
2. تغيير إشباع الصورة: أي إشباع الألوان الرئيسيّة للصورة وهي الأحمر والأخضر والأزرق، فعلى سبيل المثال، ليكن لدينا صورة يوجد فيها لون أحمر يأخذ القيمة (R=250,G=0,B=0) فعمليّة زيادة الإشباع هي زيادة قيمة مركّبة اللون الأحمر إلى القيمة 255 للحصول على اللون مشبع. أمّا عمليّة تقليل الإشباع، فهي تقليل قيمة مركّبة اللون الأحمر حتى القيمة التي يتم فيها الحفاظ على اللون الأحمر (أقلّ قيمة للون الأحمر). فأنشأنا نسخ عدّة من الصورة، منها ما تمّ تعديل مركّبة واحدة من مركّبات الألوان الثلاثة، ومنها ما تمّ تعديل إشباع المركّبات الثلاثة معاً.
3. تغيير سطوع الصورة: السطوع هو عمليّة خطيّة على عناصر الصورة، فلزيادة سطوع الصورة نقوم بإضافة قيمة معينة على جميع عناصر الصورة، ولتقليل سطوع الصورة نقوم بطرح قيمة معينة من جميع عناصر الصورة.

4. تغيير تباين الصورة: إن زيادة تباين الصورة يعني توسّع قيم العناصر على كامل مجال الهيستوغرام للصورة، أي أن المناطق المضيئة تصبح أكثر إضاءة والمناطق المظلمة تصبح أظلم. وعمليّة تقليل تباين الصورة ينتج عنها منحني ذو مجال ضيّق من قيم العناصر.
5. تغيير الدقّة المكانيّة (spatial resolution) (DPI) (تغيير عدد العناصر في كل inch من الصورة).
6. عمليّات أخرى: تغيير نوع الصورة، تصحيح غاما للصورة، تشويه الصورة.

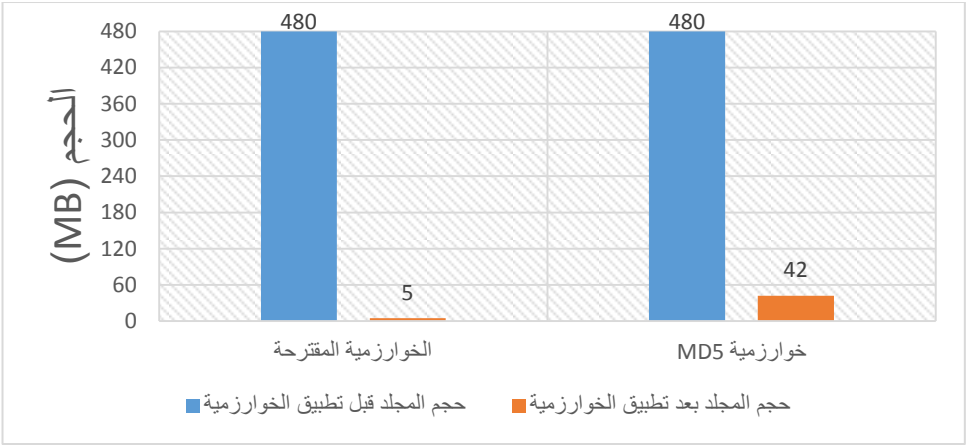
ويظهر الشكلين (8) و (9) الصورة الأصليّة ونماذج من الصور المعدّلة:

الشكل (8): الصورة الأصليّة

الشكل (9): نماذج من الصور شبه المكرّرة

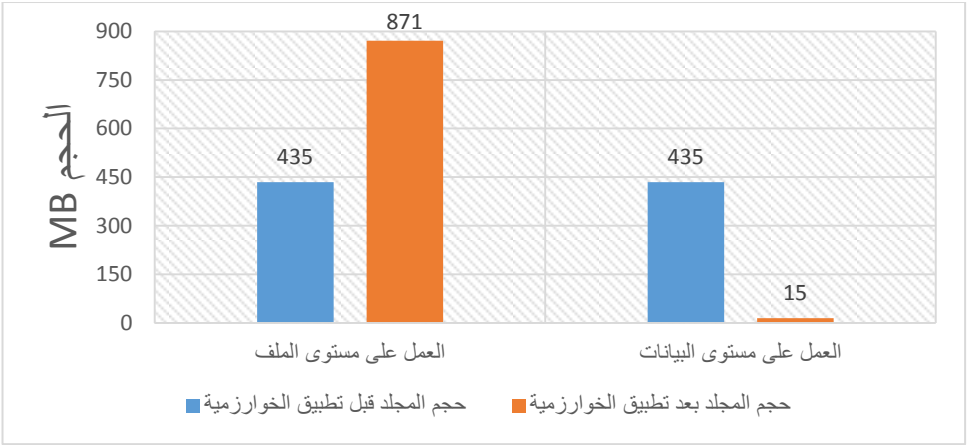
– تم تطبيق الخوارزميّة المقترحة على مجلّد بحجم 480MB يحتوي 435 ملفّ من الصور المكرّرة وشبه المكرّرة، وأظهرت الخوارزميّة فعاليّة عالية في كشف التكرار والتشابه، إذ إنّ عدد الصور المكرّرة وشبه المكرّرة المكتشفة هو 428 صورة أي بنسبة 98.39. إنّ التعديل على الصور السبعة البقية أثّر على 63 عنصر التي تمثّل العناصر الأقل تردّداً في الصورة، الأمر الذي أثّر على التفاصيل العامة للصورة بشكل كبير، كمثال على ذلك، إحدى الصور نتجت عن زيادة تباين الصورة الأصليّة بنسبة 33%. وإحداها نتجت عن تقليل سطوع الصورة بنسبة 81%، أي أنّ هذه الخوارزميّة فعّالة في حال كان التعديل على الصورة لا يؤثّر على 63 عنصر الأقل تردّداً في الصورة.

– بالمقارنة مع خوارزميّات الاختزال مثل MD5 فتظهر الخوارزميّة المقترحة نتائج أفضل، إذ إنّ عدد الصور المكتشفة باستخدام MD5 هو 384 صورة أي بنسبة 88.27%. ويبيّن الشكل (10) مقارنة بين فعاليّتي الخوارزميّتين:



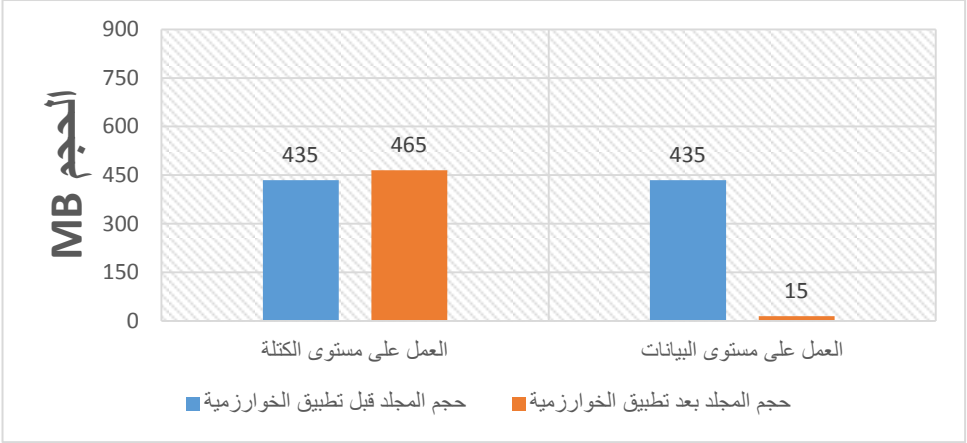
الشكل (10): مقارنة بين خوارزميّة MD5 والخوارزميّة المقترحة

– إن عمليّة كشف التكرار والتشابه المستخدمة اعتمدت على بيانات الملفّ(الصور). بالمقارنة مع الدراسات السابقة التي اعتمدت تقنيّات على مستوى الملفّ، فإن المفتاح الخاص بالملفّ يتغيّر عند أقل تغيير في بيانات الملفّ. ففي المجلد المستخدم الذي يحتوي 435 صورة إذ قمنا بتعديل صورة واحدة فقط من الملفّ، ثم قمنا بتوليد المفتاح المختزل للملف الجديد، فينتج مفتاح مختزل مغاير للمفتاح السابق. علماً أن الملفّ الجديد يحتوي على 434 صورة مماثلة للصور في الملفّ قبل تعديله. هذا بدوره يؤثّر على مساحة التخزين، حيث سيتم تخزين الملفّ المعدّل بمساحة التخزين لتصبح المساحة الإجماليّة للملفّين 871، وبالتالي سيحصل هدر في المساحة التخزينيّة. ويبين الشكل (11) الفرق بين التقنيّتين:



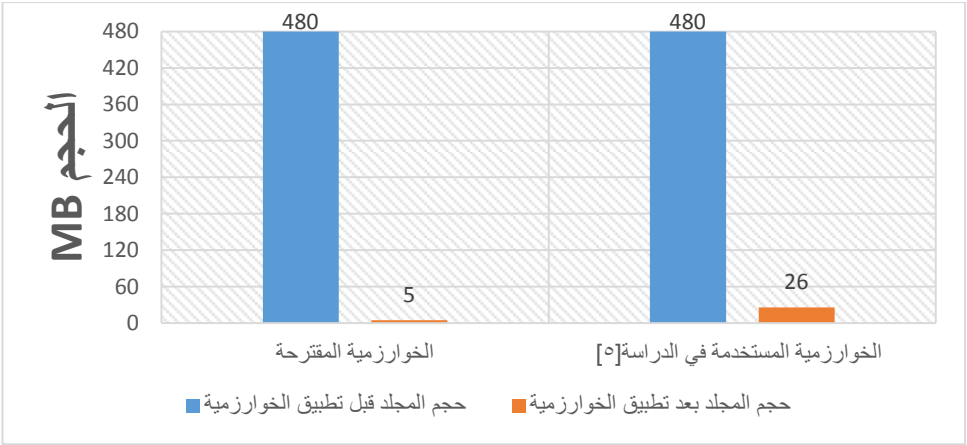
الشكل (11): مقارنة بين التنفيذ على مستوى الملفّ والتنفيذ على مستوى البيانات

–تعتبر تقنيّة حذف البيانات المكرّرة على مستوى الكتلة التي استخدمت سابقاً في الأنظمة السحابيّة والعنقوديّة ذات فعاليّة أكبر من الحذف على مستوى الملفّ، لأنه في حال تعديل صورة واحدة فقط من الملفّ واستخدام التقنيّة على مستوى الكتلة، فسيتم تقسيم الملفّ إلى كتل(أجزاء) وتوليد مفتاح مختزل لكل كتلة. عند عملية المقارنة يحصل تطابق بين أجزاء الملفّين ما عدا الجزء الذي يحوي الصورة المعدّلة، وبالتالي تعتبر هذه التقنيّة أكثر فعاليّة من التنفيذ على مستوى الملفّ، ولكنها أقل فعاليّة من التنفيذ على مستوى البيانات لأنه سيتم تخزين كتلة كاملة في مساحة التخزين. فلو تم تقسيم الملفّ الذي يحتوي 435 صورة بحجم 480، واعتماد التقسيم بحجم ثابت (64MB)، فينتج لدينا 7 أجزاء كل منها بحجم 64MB وكتلة بحجم 32MB. يبيّن الشكل (12) مقارنة بين المساحة التخزينيّة الموفّرة باستخدام تقنيّة التنفيذ على مستوى الكتل والتنفيذ على مستوى البيانات:



الشكل (12): مقارنة بين التنفيذ على مستوى الملفّ والتنفيذ على مستوى الكتلة

– بمقارنة الخوارزميّة المقترحة مع الخوارزميّة في الدراسة المرجعية [5]، تظهر الخوارزميّة المقترحة نتائج أفضل في كشف التشابه والتكرار، إذ إنّ عدد الصور المكرّرة وشبه المكرّرة التي تم كشفها باستخدام الخوارزميّة في الدراسة [5] هو 398 صورة بحجم 349MB، أي حقّقت نسبة فعاليّة تبلغ 91.5%. يبيّن الشكل (13) مقارنة المساحة التخزينيّة التي تم توفيرها باستخدام الخوارزميّتين:



الشكل (13): مقارنة بين الخوارزميّة المقترحة والخوارزميّة المستخدمة في الدراسة[5]

6- الاستنتاجات والاقتراحات

أثبتت النتائج الواردة أعلاه أنّ فعاليّة الخوارزميّة المقترحة في كشف الصور المكرّرة تماثل فعاليّة خوارزميّات الاختزال. ولكن الخوارزميّة المقترحة أثبتت فعاليّة أكبر عند التغيير في عناصر الصورة، إذ إنّ خوارزميّات الاختزال تفشل عند أقل تغيير في عناصر الصورة (تغيير خانة واحدة فقط). بينما الخوارزميّة المقترحة يمكنها كشف تشابه الصور ما دام التعديل الحاصل على الصور لا يؤثّر على 63 عنصر الأكثر أهميّة في الصورة. يمكن أن تهدف الأبحاث المستقبليّة إلى استخدام خوارزميّات أسرع في عمليّة كشف التكرار وأكثر فعاليّة في كشف الصور شبه المكرّرة بالمقارنة مع نتائج هذه الخوارزميّة. كما يمكن أيضاً تطوير خوارزميّات تطبّق عمليات كشف الصور المكرّرة وشبه المكرّرة على البيانات قبل تخزينها في أنظمة التخزين عامّةً والأنظمة السحابيّة والعنقوديّة خاصّة، وذلك لاستثمار موارد التخزين بشكل أفضل، وبالتالي الحصول على نظام تخزين مثالي.

7- المراجع

[1] Liu.A,Yu.T 2018 Overview of Cloud Storage, International Journal of Scientific & Technology Research, hal-02889947.

[2] Ashlesha.S,Tugnayat.R 2018 A Review of Hadoop Ecosystem for BigData, International Journal of Computer Applications, Volume 180 – No.14. 35-40.

[3] Muthuchamy.K, 2018 A STUDY ON VARIOUS DATA COMPRESSION TYPES AND TECHNIQUES, International Journal of Research and Analytical Reviews (IJRAR), Volume 5-No.3. 945-950.

[4] Gupta.P, Kumar.S 2014, A Comparative Analysis of SHA and MD5
Algorithm, International Journal of Computer Science and Information Technologies, VOL 5-No. 4492-4495.

[5] Tannu, Karambir 2017, Detection of De-Duplication Using SHA-512 and AES-256 in Cloud Storage, American International Journal of Research in Science, Technology, Engineering & Mathematics, AIJRSTEM 17- 323. 87-93.

[6] Raid.A, Khedr.W, El-dosuky.M, Ahmed.W 2014, Jpeg Image Compression Using Discrete Cosine Transform - A Survey, International Journal of Computer Science & Engineering Survey, Vol.5-No.2. 39-47.

[7] Pronika, S.S.Tyagi 2019, Deduplication in Cloud Storage, International Journal of Innovative Technology and Exploring Engineering, Vol.9-No.25. 364-368.

[8] Singh.S, Singh.R 2017, Next Level Approach of Data Deduplication in the Era of Big Data, International Journal of Advanced Research in Computer Science, Vol.8-No.4. 71-74.

[9] Kuma.N, Malik.P, Bhardwaj.S, S.C. Jain 2017, Enhancing Storage Efficiency Using Distributed Deduplication for Big Data Storage Systems, A UGC Recommended Journal, Vol.9-No.1. 96-108.

[10] kadam.S, Gupta.P, Veer.C, Loke.A 2016, Visual Based Image Search using Perceptual Hash Codes for Online Shopping, International Journal of Advanced Research in Computer and Communication Engineering, Vol.5-No.3. 1034-1035.

[11] Sudan.M, 2017-Hamming Codes, Distance, Examples, Limits, and Algorithms.CS 229r Essential Coding Theory, Lecture 1.