Journal of Engineering Sciences and Information Technology
Volume (5), Issue (1) : 30 Mar 2021
P: 1 - 11

ISSN: 2522-3321

مجلة العلوم الهندسية وتكنولوجيا المعلومات
المجلد (5)، العدد (1) : 30 مارس 2021م
ص: 1 - 11

Develop an algorithm to delete near-duplicate images in Hadoop

Hasan Ali Hasan

Ammar Ali Zakzouk

Faculty of Mechanical and Electrical Engineering || Albaath University || Syria

Abstract: The concept of near-duplicate images refers to images that are subjected to noise, that have been compressed, or whose resolution is reduced as a result of their transmission, and other images to which digital image operations are applied. The ideal storage system aims to optimize the storage space, by managing, structuring and organizing data in an efficient manner, so that the storage space is preserved including valuable and useful information, and we get rid of useless data. The space occupied by insignificant data is called wasted space, and this space increases with the increase of these files, resulting in a waste of storage space, which makes it difficult to manage storage space and organize data, which affects the overall system performance. Hadoop is used to store and process big data, and depends on branching in storing data, as the data is divided into parts (blocks), and these parts are distributed in computer devices, called these devices (Data Nodes). Researchers have developed techniques to get rid of fragments of duplicate data, in order to save storage space in the Hadoop system, but each node may contain unimportant files occupying part of this space, so we will present in this research a technique to delete the near-duplicate images stored within data nodes, using a Discrete Cosine Transform (DCT).

Keywords: Digital Image processing, Image compression, Near-duplicate images, Hadoop, DCT.

تطوير خوارزمية لحذف الصور شبه المكرّرة في Hadoop

حسن علي حسن

عمار علي زقزوق

كلية الهندسة الميكانيكية والكهربائية || جامعة البعث || سوريا

الملخص: يطلق مفهوم الصور شبه المكرّرة على الصور التي تتعرّض للضجيج، أو التي تمّ ضغطها، أو التي تنخفض دقّتها نتيجة إرسالها، وغير ذلك من الصور التي يطبّق عليها عمليّات الصورة الرقميّة. إنّ نظام التخزين المثالي يهدف للاستثمار الأمثل لمساحة التخزين، وذلك عن طريق إدارة وهيكلة وتنظيم البيانات بطريقة فعّالة، بحيث يتمّ المحافظة على مساحة التخزين متضمنةً معلومات قيّمة ومفيدة، والتخلّص من البيانات غير المفيدة. يسمّى الحيّز الذي تشغله البيانات غير المهمّة بالمساحة الضائعة، وتزداد هذه المساحة بزيادة هذه البيانات، فيحصل هدراً في مساحة التخزين، مما يصعّب من إدارة مساحة التخزين وتنظيم البيانات، الأمر الذي يؤثّر على أداء النظام بشكل عام. يستخدم Hadoop لتخزين ومعالجة البيانات الضخمة، ويعتمد التفرّع في تخزين البيانات، إذ يتم تقسيم البيانات إلى أجزاء (Blocks)، وتوزّع هذه الأجزاء في أجهزة حاسوبيّة، تسمّى هذه الأجهزة (Data Nodes). طوّر الباحثون تقنيّات للتخلّص من أجزاء البيانات المكرّرة، وذلك لتوفير مساحة تخزينيّة في نظام Hadoop، ولكن قد تحتوي كل عقدة حاسوبيّة على ملفّات غير مهمّة، فتشغل قسماً من هذه المساحة، لذلك سنقدّم في هذا البحث تقنيّة لحذف الصور شبه المكرّرة المخزّنة ضمن Data Nodes، وذلك باستخدام تحويل جيب التمام المتقطّع DCT(Discrete Cosine Transform).

الكلمات المفتاحية: معالجة الصورة الرقميّة، ضغط الصورة، الصور شبه المكرّرة، Hadoop، تحويل جيب التمام المتقطّع.

1- المقدمة:

تعتمد عمليّة كشف التكرار بين الملفّات بجميع أنواعها على البيانات الثنائيّة للملفّات سواء كانت هذه الملفّات نصوصاً آو صوراً أو مقاطع مرئيّة أو غيرها من الملفّات، ففي حال تماثل البيانات الثنائيّة لملفّين يكون الملفّان مكرّران. استخدمت تقنيّات عدّة لكشف التكرار بين الملفّات، وتعتبر خوارزميّات الاختزال أهم هذه التقنيّات. تعتمد خوارزميّات الاختزال على إنشاء مفتاح خاص لكل ملف يسمّى مفتاح الاختزال (hash key)، ومن ثمّ تتم عمليّة المقارنة بين مفاتيح الاختزال لكشف التطابق بين الملفّات. تختلف خوارزميّات الاختزال عن بعضها بشكل عام بطول مفتاح الاختزال، والذي بدوره يؤثّر على سرعة تنفيذ الخوارزميّة وأمانها وفعاليّتها في كشف التكرار.

ظهرت في السنوات السابقة تقنيّات متعددة لكشف تطابق الملفات والكتل في الأنظمة العنقوديّة التي تستخدم Hadoop كنظام تخزين.

قام Parth Shah وآخرون بتقديم تقنيّة لإلغاء البيانات المكررة في Hadoop على مستوى الملف باستخدام خوارزمية SHA. حيث يتم تشكيل جدول في قاعدة بيانات Hadoop يسمى (Hbase) يحوي مفاتيح الاختزال للملفّات المخزّنة ضمن النظام. أي ملف يراد تخزينه في Hadoop يتم توليد مفتاح SHA له، فإذا لم يتطابق المفتاح الخاص بالملف مع المفاتيح الموجودة في Hbase، يتم تخزين الملف في نظام التخزين، وتخزين المفتاح الخاص بالملف في Hbase. وإذا حصل تطابق لا يتم تخزين الملف [4]. تمتاز هذه التقنية بالسرعة في التنفيذ، لكن بفعاليّة قليلة، لأنه إذا كان لدينا ملف ذو حجم كبير يراد تخزينه في Hadoop، وهناك اختلاف بت واحد فقط مع الملفات الموجودة في Hadoop سيتم تخزين الملف، وبالتالي سيحصل هدر في مساحة التخزين، وينعكس ذلك سلباً على أداء النظام.

قام Naresh Kumar وآخرون بتقديم تقنيّة لإلغاء البيانات المكرّرة المخزنة في Hadoop على مستوى الكتلة باستخدام خوارزميّة MD5، يتم تخزين مفاتيح الاختزال لكل الكتل المخزنة فيHadoop ضمن buckets. فعندما يتم تخزين ملفّ في نظام التخزين الموزع يتم تقسيم الملفّ إلى كتل، وتوليد 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 وقيمة الهاش الخاصّة بالكتلة في buckets [7]. تعتبر التقنيّة المستخدمة في هذه الدراسة فعّالة بدرجة أكبر في التعامل مع البيانات المكرّرة بالمقارنة مع الدراسة السابقة، لأن عملية المطابقة تتم على مستوى أقل حجماً من البيانات، كما أنّ خوارزمية MD5 المستخدمة أسرع في التنفيذ من خوارزميّة SHA.

قام Rui Hou وآخرون بتقديم تقنيّة لإلغاء البيانات المكررة في نظام Hadoop على مستوى الكتلة، تختلف هذه التقنيّة عن الدراسة السابقة في جدول فهرسة مفاتيح الاختزال (Hbase)، إذ تقوم هذه التقنيّة بتقسيم جدول الفهرسة إلى أقسام، ووضع كل قسم في جهاز حاسوبي، أي لكل عقدة حاسوبية ضمن الشبكة جدول فهرسة خاص بها يتضمن مفاتيح الاختزال للكتل المخزّنة ضمنها، وبالتالي كل عقدة تبحث عن التكرار ضمن الجدول المخزّن ضمنها [6]. لذلك هذه التقنيّة أسرع في عملية البحث والمقارنة من التقنيّة السابقة.

فيما يخص الصور الرقميّة، تتكوّن الصورة من مجموعة من العناصر(Pixels)، فعندما تكون الصور مكرّرة تكون عناصر هذه الصور متماثلة، لكن في حال تغيّر عنصر واحد من عناصر هذه الصور تكون الصور غير متماثلة، وبالتالي تفشل جميع خوارزميّات الاختزال عند تغيير عنصر واحد فقط في الصورة، ويطلق على هذه الصور بالصور شبه المكرّرة. ظهرت في السنوات السابقة دراسات حول كشف تشابه الصور في أنظمة التخزين وكان أهمها الدراسة التالية:

قام Shardha Kadam واخرون بتقديم تقنيّة لكشف الصور المتشابهة في أنظمة التخزين، تعتمد هذه التقنيّة على توليد مفتاح خاص لكل صورة بالاعتماد على عناصر الصورة، ومن ثمّ تتم المقارنة بين المفاتيح لكشف التشابه. ويتم توليد المفتاح 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 يتم وضع القيمة 1 في المفتاح، وإلا يتم وضع القيمة 0، وهكذا حتى يتم توليد المفتاح [8]. بالمقارنة مع الدراسات السابقة، في حال كانت البيانات تحتوي صوراً مكرّرةً فقط، فهذه التقنية تستغرق وقتاً أطولاً في عملية كشف التكرار. أما في حال كانت البيانات تحتوي صوراً متشابهة، فتظهر هذه الخوارزمية فعالية أكبر من جميع خوارزميات الاختزال المستخدمة سابقاً والتي تفشل عند أقل تغيير قد يحصل على الصورة.

سنتطرق في هذا البحث إلى التعامل مع بيانات الكتلة، والتي تضم مجموعة من الصور شبه المكرّرة، وذلك بعد تخزين الكتلة في Hadoop.

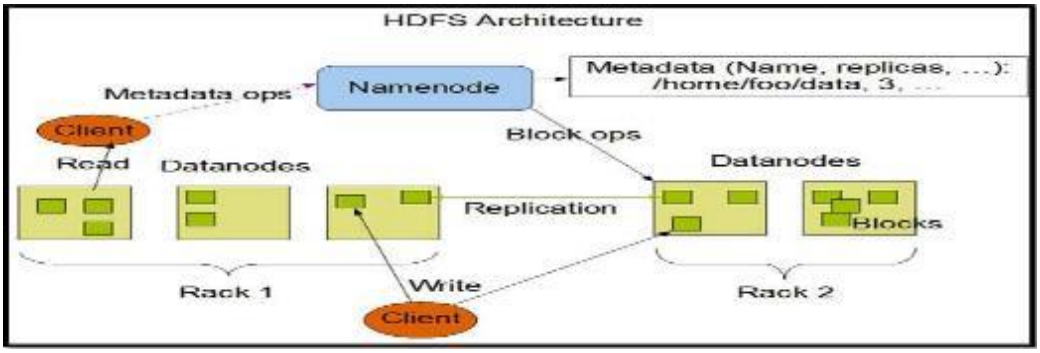
1.1- Hadoop:

هو منصة مفتوحة المصدر ومكتوبة بلغة جافا تستخدم لتخزين ومعالجة البيانات الضخمة (Big Data)، يعتمد Hadoop على التوزّع في التخزين والمعالجة، حيث 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 التقليدية في هيكلة وتخزين وتحليل هذه البيانات، وبالتالي لا بد من نظام يمكّننا من التعامل مع هذه البيانات.

يتكوّن نظام Hadoop من قسمين رئيسين:

1- Hadoop Distributed File System (HDFS): يؤمّن القدرة على تخزين كميّات كبيرة من البيانات.
2- MapReduce: يؤمّن القدرة على معالجة بيانات الضخمة.

يتكوّن نظام Hadoop الموزع من عقدة رئيسة (Name Node)، وعقد ثانويّة عدّة (Data Nodes). مهمة العقدة الرئيسة تتمثّل في إدارة تخزين البيانات مثل معرفة السعة التخزينيّة المتوفّرة في العقد الثانويّة، أسماء الملفّات والملفات المخزّنة في نظام التخزين الموزّع وغيرها...، أما تخزين البيانات ومعالجتها يتم في العقد الثانويّة. ويوضّح الشكل (1) بنية نظام تخزين Hadoop الموزّع:



الشكل (1) بنية HDFS

عندما يتم تخزين ملف في Hadoop، يتم تقسيم الملف إلى أجزاء، وتقوم العقدة الرئيسة بتوزيع هذه الأجزاء في العقد الثانويّة، وذلك بعد توافرها على معلومات عن المساحة المتوفّرة في كل عقدة ثانويّة. كل كتلة من البيانات يتم نسخها ثلاث مرّات، ووضع كل نسخة في عقدة ثانويّة، وذلك لحماية البيانات من الضياع [2].

2.1- خوارزميات الاختزال (التجزئة) (Hash Algorithms):

خوارزميّات الاختزال تقوم بتحويل سلسلة من الأحرف إلى قيمة أو مفتاح أقصر بطول ثابت يمثل السلسلة الأصليّة. يتم استخدام قيمة الاختزال لفهرسة العناصر واستردادها في قاعدة بيانات لأنه من الأسرع العثور على العنصر باستخدام المفتاح ذي التجزئة من العثور عليه باستخدام القيمة الأصليّة. كما أنها تستخدم في العديد من خوارزميات التشفير.

تعتبر خوارزميّتي 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، حيث تتعامل هاتين الخوارزميّتين مع بيانات بحجم 512بت وفي حال كانت الرسائل أكبر تقسم إلى رسائل صغيرة. إنّ الحد الأعظمي من البيانات التي يمكن للخوارزميّتين التعامل معها هو 264 بت، ولكن تختلفان بطول مفتاح التجزئة الناتج، فخوارزميّة MD5 تنتج قيمة تجزئة بطول 128بت، بينما خوارزميّة SHA-1 تنتج قيمة تجزئة بطول 160بت. أمان الخوارزميّة وسرعتها تتعلّقان بشكل أساسي بطول قيمة التجزئة، لذلك SHA-1 أكثر أماناً من MD5، بينما MD5 أكثر سرعة [3].

3.1- الصورة الرقمية:

يرمز للصورة الرقمية بمصفوفة ثنائية البعد f(x, y)، وهي عبارة عن صورة يتم تمثيلها بشكل رقمي، أي مجموعة من البتات (0،1)، من ناحية أخرى، تتكون الصورة من قطع مربعة صغيرة تدعى عناصر(نقاط) الصورة (Pixels)، مواقع هذه العناصر في المصفوفة تناظر مواقع نقاط الصورة الأصليّة المتمثّلة بالإحداثيّات (x, y)، و قيم تلك العناصر تتناسب مع قيمة الشدّة الضوئيّة عند تلك النقاط.

- معالجة الصورة الرقمية:

يشير مصطلح "معالجة الصورة" إلى الإجراءات التي يتم بموجبها تغيير المعلومات الموجودة في الصورة لاستعادة الصورة أو تحسينها بشكل مرئي. الأمثلة النموذجيّة هي تصحيح شحذ الصورة الناتج عن ضعف التركيز، وتصحيح الأخطاء البصريّة للعدسات، وتصحيح التباين، والشدة أو السطوع، وتصحيح الألوان، وتحسين بنية الصورة للتأكيد على العناصر التي لا يمكن رؤيتها بسهولة في الصورة الأصلية، والتخلص من الضجيج في الصورة. فالصورة الرقميّة تتميّز بمجموع من الخصائص كالسطوع والتباين واللون وغيرها...، ويؤدّي التغيّر في هذه الخصائص إلى إحداث تغيّرات في الصورة الرقميّة، وتعتمد درجة التغيّر على خصائص الصورة. ومن هذه الخصائص:

1- السطوع (Brightness): يمكن تعريف السطوع على أنه مقدار الطاقة الناتجة من مصدر الضوء بالنسبة للمصدر الذي نقارنه به. في بعض الحالات يمكننا القول بسهولة أن الصورة مشرقة، وفي بعض الحالات، ليس من السهل إدراكها. فالإضاءة القويّة تعطي درجة سطوع عالية للصورة، في حين تعطى الإضاءة الخافتة درجة سطوع منخفضة للصورة.
2- التباين (Contrast): ويمثّل الفرق بين أكبر وأصغر قيمة في عناصر الصورة، فإذا كانت قيمة الفرق عالية يكون التباين عالي، وعندما تكون قيمة الفرق صغيرة يكون التباين منخفض.

3- الإشباع (Saturation): يمثّل مدى كثافة الألوان في الصورة، تحتوي الصورة عالية التشبع على ألوان عالية الكثافة، والصورة منخفضة التشبع تكون قريبة جدًا من الصورة ذات القيمة الرمادية [1].

1.3.1- تحويل جيب التمام المتقطع Discrete Cosine Transform (DCT):

هو أحد أساليب الضغط التي يتم تطبيقها في مجال الصور. يعتبر هذه التحويل مشابه جداً لتحويل فوريه، حيث يقومان بنقل الصورة من المجال الفراغي(المكاني) إلى المجال الترددي. لكن تحويل فوريه يستخدم علاقات رياضية معقدة تتطلب مدة زمنية كبيرة، فهو يتعامل مع الأعداد المعقدة (complex) أي الأعداد الحقيقية والتخيلية. بينما تحويل جيب التمام المنفصل يستخدم علاقات رياضية بسيطة ذات فترة تنفيذ قصيرة، فهو يتعامل فقط مع الأعداد الحقيقية مما يجعله أكثر استخداماً في عمليات ضغط الصورة. يعمل DCT على فصل الصورة إلى أجزاء أو نطاقات فرعيّة مستقلة تختلف هذه الأجزاء من حيث الأهمية، حيث يتم تقسيم الصورة إلى كتل (blocks) بحجم 8*8، ويتم تطبيق تحويل DCT على كل بلوك. 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 (الصورة المرخمة smoothed)، في حين تمثّل المعاملات الباقية مركّبات التردّد المرتفع (الحواف edges) [5].

رياضياً، يعرّف تحويل جيب التمام المتقطع بالعلاقة (1):

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

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- مشكلة الدراسة:

تعتبر مساحة التخزين من الأمور الهامّة في مجال قواعد البيانات، خاصّةً عند التعامل مع بيئة البيانات الضخمة. فالمؤسسات التي تستخدم Hadoop لتخزين البيانات الضخمة، تستقبل البيانات بكميّة كبيرة ومن مصادر مختلفة، فلا بد من الاستثمار الأمثل لهذه البيانات والحصول على المعلومات المفيدة منها. إنّ الصور الرقميّة هي أحد موارد أنظمة التخزين، فعند استقبال أنظمة التخزين لهذا النوع من البيانات، فقد تتكوّن مجموعات هذه الصور من أنواع عدّة، إحدى هذه الأنواع هو الصور شبه المكرّرة، والتي يمكن أن نطلق عليها بالصور المتشابهة، فلا بدّ من كشف التشابه بينها وتخزين الصور الأكثر دقّة بينها. وهذا الأمر يساهم في توفير مساحة تخزينية في النظام، وتسهيل تنظيم البيانات وإدارتها، وبالتالي تحسين أداء النظام.

3- مواد البحث وطرائقه:

قبل البدء بالخوارزميّة المقترحة لا بدّ من التعريف بالبرمجيات المستخدمة وهي: نظام Ubuntu 16.04، برنامج Hadoop 2.7.3، برنامج Eclipse IDE.

- الخوارزمية المقترحة: نقوم بتوليد مفتاح خاص لكل صورة، مكوّن من 64 بت، نسمّي هذا المفتاح بالمفتاح المختزل (Hash Key)، ومن ثم تتم المقارنة بين الصور لكشف التشابه بينها. ويتم تشكيل المفتاح وفق الخطوات التالية:

- الخطوة الأولى 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. نقوم بتحويل الصورة الملوّنة إلى رماديّة وفق طريقة الأوزان حيث يتم تحويل قيمة كل عنصر حسب العلاقة (2):

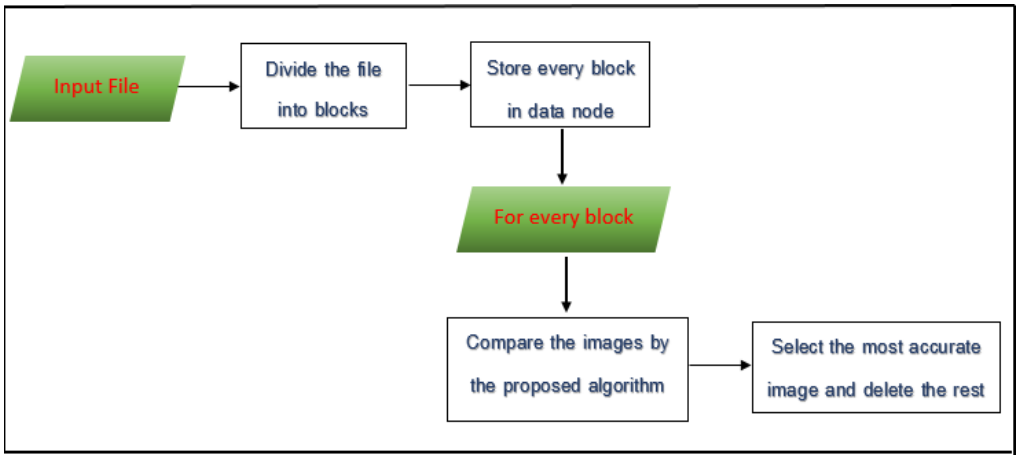
$$New\_Value = (0.3)R * (0.59)G * (0.11)B \quad (2)$$

واعتمدت هذه الطريقة بدلاً من أخذ القيمة المتوسّطة لقيمة الألوان الثلاثة وذلك لاختلاف الأطوال الموجية للألوان، وبالتالي حساسيّة العين البشرية للألوان، فالطول الموجي للون الأحمر أكبر من الطول الموجي للون الأخضر، فيتم تقليل مساهمة اللون الأخضر وزيادة مساهمة اللون الأخضر.

- الخطوة الثانية تعتمد على تقسيم الصورة إلى Blocks، كل بلوك بحجم 8*8 أي 64 عنصر في كل بلوك.
- في الخطوة الثالثة يتم حساب تحويل DCT لكل بلوك. وبالتالي تنتج مصفوفة تمثّل معاملات DCT. تكون قيمة المعاملات كبيرة في الزاوية العليا واليسرى من المصفوفة وتقل باتجاه الأسفل واليمين، والمعاملات العالية تمثّل التردّدات المنخفضة أي تفاصيل الصورة، والمعاملات المنخفضة تمثّل التردّدات المرتفعة أي حواف الصورة.
- في الخطوة الرابعة نقوم بأخذ القيم من العنصر (0،0) حتى العنصر (8،8) أي 64 عنصر الأولى من مصفوفة DCT والتي تمثّل التردّدات 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 بت والذي يمثّل المفتاح الخاص بالصورة.

- تطبيق الخوارزمية المقترحة في Hadoop:

يبين الشكل (2) تقنية إلغاء الصور شبه المكررة في Hadoop:



الشكل (2) إلغاء الصور شبه المكرّرة في Hadoop

- حيث يتم تقسيم ملف الإدخال في Hadoop إلى أجزاء(blocks)، كل بلوك بحجم 128MB. ثم تقوم العقدة الرئيسة بتخزين الكتل في العقد الثانويّة. تقوم كل عقدة بتطبيق الخوارزميّة المقترحة على الجزء المخزّن ضمنها لكشف الصور شبه المكرّرة. بعد ذلك يتم تحديد الصور الأكثر دقّة بينها وحذف الصور البقيّة.

4- النتائج ومناقشتها:

تم تشكيل ملف الإدخال من مجموعة من الصور شبه المكرّرة، نتجت هذه الصور عن إجراء بعض عمليّات الصورة الرقميّة على صورة من النوع JPEG ذات أبعاد1080*1920، ونتج عن هذه العمليّات نسخ معدّلة عن هذه الصورة، ويظهر الشكل (3) الصورة الأصليّة ونماذج من الصور المعدلة:

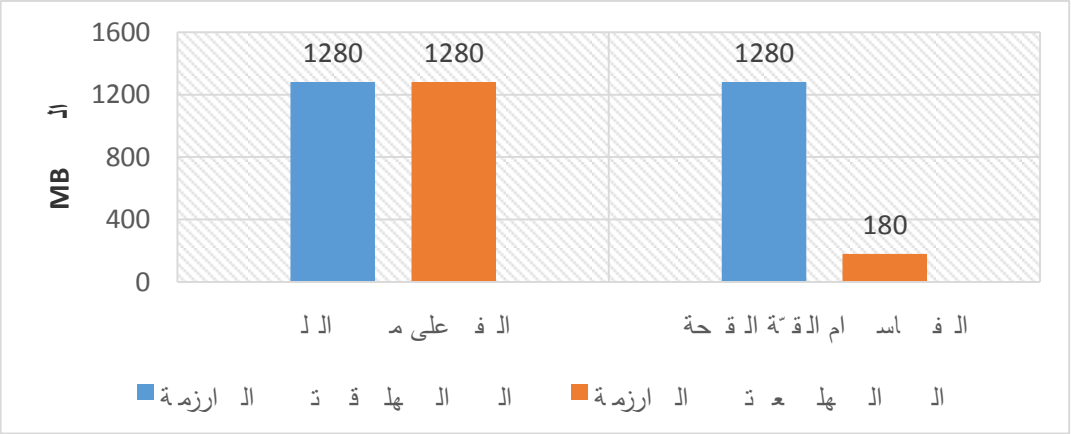
الشكل (3) نماذج من الصور شبه المكرّرة

وفيما يلي العمليّات الي تمّ تنفيذها على الصورة:

1- تغيير أبعاد الصورة: تمّ إنشاء نسخة من الصورة بزيادة عدد عناصر الصورة، حيث قمنا بتكبير الصورة 4 مرات لتصبح أبعاد الصورة 4320*7680، وبالتالي الحصول على صورة مغايرة تماماً ولكن بنفس تفاصيل الصورة، ونسخة أخرى بتصغير أبعاد الصورة، حيث قمنا بتقسيم العرض والارتفاع على القيمة 32 لتصبح أبعاد الصورة 33*60.

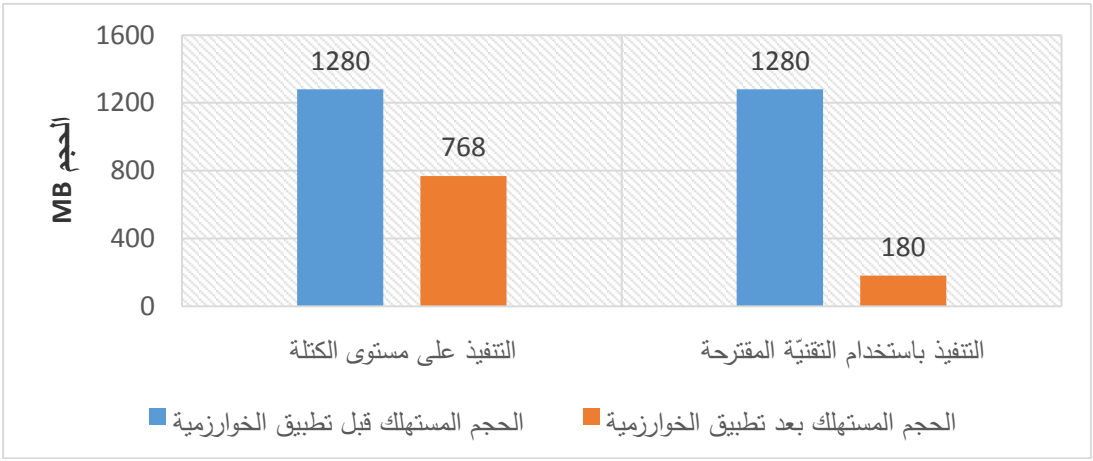
2- تغيير إشباع الصورة: أنشأنا نسخ عدّة من الصورة، منها ما تمّ تعديل إشباع مركّبة واحدة من مركّبات الألوان الرئيسيّة (الأحمر، الأخضر، الأزرق)، ومنها ما تمّ تعديل إشباع المركّبات الثلاثة معاً.
3- تغيير سطوع الصورة: حيث تم إنشاء نسخ ذات سطوع عالي، وصور ذات سطوع منخفض.
4- تغيير تباين الصورة: حيث تم إنشاء نسخ ذات تباين عالي، وصور ذات تباين منخفض.
5- تغيير نوع الصورة: حيث تم تحويل نوع الصورة من JPEG إلى PNG وGIF.
6- ضغط الصورة.
7- تغيير DPI (تغيير عدد العناصر في كل inch من الصورة).
8- تصحيح غاما للصورة.
9- تشويه الصورة وتطبيق ضجيج عليها.

- تم تطبيق الخوارزميّة المقترحة على ملف بحجم 640MB يحتوي 620 صورة. عند التخزين في Hadoop، تم تقسيم الملف إلى 5 أجزاء وكل جزء بحجم 128MB، وتخزين كل جزء في عقدة حاسوبيّة. كل عقدة قامت بتطبيق الخوارزمية المقترحة على الكتلة المخزّنة ضمنها. وأظهرت الخوارزميّة فعاليّة عالية في كشف الصور شبه المكرّرة. إذ إنّ عدد الصور التي تم كشفها هو 578 صورة أي بنسبة تقريبية 93.22%. إنّ التعديل على الصور البقية أثّر على العناصر 64 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 أن هذه الخوارزميّة فعّالة في حال كان التعديل على الصورة لا يؤثّر على العناصر 64 الأقل تردّداً في الصورة.
- بالمقارنة مع خوارزميّات الاختزال مثل MD5 فتظهر الخوارزميّة المقترحة نتائج أفضل، إذ إنّ خوارزميات الاختزال تفشل في كشف تشابه الصور، لأنّ تغيير قيمة عنصر واحد من الصورة يؤدّي إلى توليد مفتاح مغاير.
- بالمقارنة مع الدراسة [1] التي اعتمدت إلغاء البيانات المكرّرة على مستوى الملفّ، فإن المفتاح الخاص بالملفّ يتغيّر عند أقل تغيير في بيانات الملفّ. ففي حال قمنا بتخزين الملف ذو الحجم 640MB، والذي يحوي 620 صورة في Hadoop. ثم قمنا بتغيير صورة واحد فقط من الصور الموجودة في الملف. فلو أردنا تخزين الملف الجديد في Hadoop، سيتم توليد مفتاح مختزل للملف ومقارنته مع مفاتيح الاختزال في Hbase، فستتم مقارنة مفتاحه مع المفتاح للملف السابق ولا يحصل تطابق وسيتم تخزين الملف، مع العلم أن الاختلاف بين الملفين هو بصورة واحدة فقط. وهذا بدوره يؤثّر على مساحة التخزين حيث سيتم تخزين الملفّ المعدّل بمساحة التخزين ليصبح المساحة الإجمالية للملفّين 1280، وبالتالي سيحصل هدر في المساحة التخزينيّة، وبالتالي تفشل الخوارزميّة المستخدمة في الدراسة [1] في توفير أي مساحة تخزينية في حال تم التغيير على الملف. ويبين الشكل (4) الفرق بين التقنيّتين:



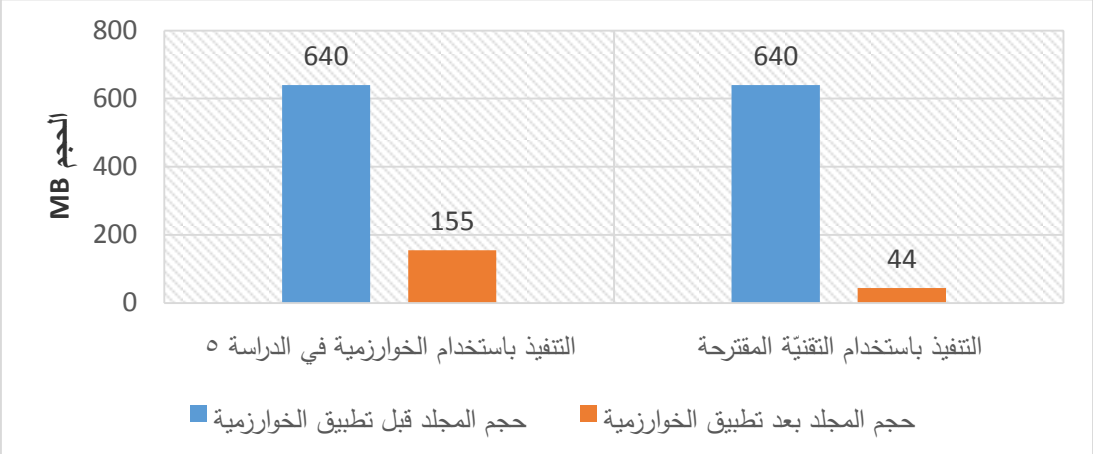
الشكل (4) مقارنة بين التنفيذ على مستوى الملفّ والتقنية المقترحة

- تعتبر تقنية حذف البيانات المكرّرة على مستوى الكتلة التي استخدمت في الدراستين [2] و[3] ذات فعاليّة أكبر من الحذف على مستوى الملفّ، لأنه لو كان التعديل مثلاً على صورة واحدة من الملفّ كما ورد سابقا، فعند تقسيم الملفّ إلى كتل(أجزاء) وتوليد مفتاح مختزل لكل كتلة، 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. ويبين الشكل (5) مقارنة بين المساحة التخزينيّة الموفّرة باستخدام تقنيّة التنفيذ على مستوى الكتل والتنفيذ باستخدام التقنيّة المقترحة:



الشكل (5) مقارنة بين التنفيذ على مستوى الكتلة والتقنيّة المقترحة

- بمقارنة الخوارزميّة المقترحة مع الخوارزميّة في الدراسة المرجعية[5] ، تظهر الخوارزميّة المقترحة نتائج أفضل في كشف الصور شبه المكرّرة، إذ إنّ عدد الصور شبه المكرّرة التي تم كشفها باستخدام الخوارزميّة في الدراسة [4] هو 485 صورة، أي حققت نسبة فعاليّة تبلغ 75.48%. يبين الشكل (6) مقارنة المساحة التخزينيّة التي تم توفيرها باستخدام الخوارزميّتين:



الشكل (6) مقارنة بين الخوارزميّة المقترحة والخوارزميّة المستخدمة في الدراسة[5]

5- الخلاصة:

أثبتت النتائج الواردة أعلاه فعاليّة الخوارزميّة المقترحة في كشف الصور شبه المكررة ما دام التعديل الحاصل على الصور لا يؤثّر على 64 عنصر الأكثر أهميّة في الصورة، والتي تجسّد أهم تفاصيل الصورة. كما بينت النتائج فعالية أكبر للتقنية المقترحة في توفير مساحة تخزينية فيHadoop بالمقارنة مع الدراسات السابقة، وذلك في حال كان للصور شبه المكررة الحجم الأكبر من بيانات الإدخال.

6- التوصيات:

يمكن أن تهدف الأبحاث المستقبليّة إلى استخدام خوارزميّات تكون أكثر فعاليّة في كشف تشابه الصور، وتطوير تقنيّات تكون أكثر مرونة في التعامل مع البيانات، لتعالج مشكلة الحجم القليل للصور شبه المكرّرة في بيانات الإدخال. كما يمكن أيضاً تطوير تقنيّات تطبّق عمليات كشف الصور شبه المكرّرة على البيانات قبل تخزينها في Hadoop، وذلك لاستثمار موارد التخزين بشكل أفضل، وبالتالي الحصول على نظام تخزين مثالي.

7- المراجع:

1- J. Sachs, Digital Image Basics, Digital Light & Color, Cambridge, Massachusetts, 1999.

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

3- Piyush. G, Sandeep. K, "A Comparative Analysis of SHA and MD5 Algorithm", International Journal of Computer Science and Information Technologies, VOL 5-No, pp. 4492-4495, 2014.

4- Parth. S, Amit. G, Sandipkumar. P, Priteshkumar. P, " Efficient Cross User Client Side Data Deduplication in Hadoop", Journal of Computers, DOI: 10.17706/jcp.12.4, pp. 362-370, 2016.

5- Raid. A, Wael. K, Mohamed. E, Wesam. A, "Jpeg Image Compression Using Discrete Cosine Transform - A Survey", International Journal of Computer Science & Engineering Survey, Vol.5-No.2, pp. 39-47, 2014.

6- Rui. H, Guiqiang. N, Yinjin. F, Qing. L, " Hadoop Based Scalable Cluster Deduplication for Big Data", International Conference on Distributed Computing Systems Workshops (IEEE), DOI: 10.1109/ICDCSW.2016.17, pp. 98-105, 2016.

7- Naresh. K, Preeti. M, Sonam. B, S.C. Jain, "Enhancing Storage Efficiency Using Distributed Deduplication for Big Data Storage Systems", A UGC Recommended Journal, Vol.9-No.1, pp. 96-108, 2017.

8- Shradha. K, Priyanka. G, Charul. V, Akshay.L, "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, 2016.