معالجة صور الأقمار الصّناعيّة باستخدام شبكات التّعلّم العميق على منصّات البيانات الكبيرة

م. عروه احمد قصاب د.م. أحمد محمود أحمد د.م. عهد نصر البودي

كلية الهندسة المعلوماتية – جامعة تشرين

الملخص

مكّن التقدّم الكبير الحاصل في تقنيات الاستشعار عن بعد من جعل حسّاسات الأقمار الصّناعيّة قادرة على التقاط صور عالية الوضوح والدّقّة وذات حزم طيفيّة متعدّدة. تحوي هذه الصّور معلومات قيّمة يمكن استخدامها في تطبيقات معالجة الصّور في عدّة مجالات مختلفة، كتصنيف الغطاء اللأرضي واكتشاف التغيّرات والتحذير المبكّر من الكوارث. الّا أنّ هذه التّطبيقات وخصوصاً تلك التي تعتمد على تقنيات التّعلّم العميق تواجه مشكلة الحاجة الكبيرة لموارد الحوسبة أثناء معالجة بياناتها إضافةً الى الوقت الكبير المُستهلك عند تنفيذها على النّظم التّقليديّة، وهذه المشكلة تزداد سوءا للتطبيقات التي تتطلب سرعة استجابة عالية. الأكثر من ذلك وبسبب الازدياد الكبير في الصّور المُولّدة عن طريق الاقمار الصّنعيّة أصبحت هناك حاجة لوجود أنظمة قادرة على استيعاب وتخزين حجوم ضخمة من الصّور الموّلّدة وبنفس الوقت لها القدرة على التّوسّع عند زيادة حجمها. يدرس هذا البحث استخدام النّظم فائقة الأداء ومنصّات البيانات الكبيرة باستخدام سبارك وهادووب واستثمارها في معالجة صور الأقمار الصّناعيّة بمساعدة أُطر عمل تفرّعيّة مُخصّصّة (Horovod, TensorFlowOnSpark)، حيث تمّ العمل على نمّوذج تعلّم عميق للتّقطيع الدّلاليّ (U-Net) كمثالٍ عمليٍّ وقُيّم أداءُ النّظام في كلّ من برنامجي التّدريب والتّنبؤ. هذا وقد أظهرت النّتائج تحسيناً كبيراً في زمن التّنفيذ للمنهجيّة

المُقترحة بالمقارنة مع منهجيّة الجهاز المنفرد وقدرةً على التّوسّع لحجوم مختلفة من عناقيد البيانات الكبيرة.

الكلمات المفتاحيّة

التّعلّم العميق – معالجة صور الأقمار الصّناعيّة – نمّوذج U-Net للتّقطيع الدّلاليّ – المُعالجة التّفرّعيّة – النّظم فائقة الأداء – منصّات البيانات الكبيرة

Satellite Image Processing Using Deep Learning Networks on Big Data Platforms

Eng. Orwa Ahmad Kassab Dr. Ahmad Mahmoud Ahmad Dr. Ahed Alboody

Faculty of Informatics Engineering, Tishreen University

Abstract

The recent advancement in remote sensing technologies made satellite sensors capture high-resolution and hyper-spectral images. These images contain valuable information and can be utilized in many applications and different disciplines like land cover classification, change detection and early warning of disasters. These applications especially that depend on deep learning technologies face the problem of high demands of computation resources and take much time to execute on traditional systems, and this problem got worse for applications that require rapid response. Moreover, the increasing amount of images generated by satellites needs a system that can store massive volumes of data and has the ability to scale when their size increases. This paper investigates the use of high-performance systems and big data platforms where Spark and Hadoop can be utilized in processing images with the help of specialized distributed computing frameworks like Horovod and TensorFlowOnSpark. A segmentation deep learning model

was taken as an example application (U-Net) and the system performance was evaluated in training and prediction programs. Results showed a significant improvement in execution time of the proposed approach compared to the single machine approach and an ability to scale for different big data cluster sizes.

Key words: Deep Learning, Satellite Image Processing, U-Net Semantic Segmentation, Distributed Processing, High-Performance Systems, Big Data Platforms,

1- مقدّمة

تتوافر اليوم العديد من الشركات العاملة في مجال الاستشعار عن بعد والتي تؤمّن صوراً مُلتقطة عن طريق أقمار صنعيّة وتوظفها في العديد من المجالات المختلفة كاكتشاف التغيرات وتصنيف المناطق والإنذار المبكّر من الكوارث وجمع احصائيّات عن مناطق واسعة من الأرض كالمحاصيل او توزّع الغابات والجفاف وغيرها. ومع دخول عصر المراقبة الأرضية عالية الدقة أصبحت بيانات الاستشعار عن بعد ذات حجوم كبيرة ومتوفّرة بشكل يومي وتمرّ بمرحلة تضخم هائل، ويُوصف نُموّها على أنّه انفجاريّ، كما انّ انتشار هذا النّوع من البيانات يدفع بقوّة نحو زيادة التعقيد في التّعامل معها نتيجة تنوّعها وتعدّد خصائصها وأبعادها [1]. وقد وصلت مرحلة وضوح الصور المقدّمة درجة أن يصل حجمها الى ما يقارب ال 50 GB ، وهذا يُعتبر فائق الحجم بالمقارنة مع الصّور الملتقطة بالكاميرات الرّقميّة الحديثة التي تتزّود بها الأجهزة الذكيّة المنتشرة بكثرة.

يتم استثمار هذه البيانات في العديد من التّطبيقات الخدميّة المختلفة والمتنوّعة من ناحية الاغراض الّا أنّ الاستجابة الزمنية لتطبيقات معالجة الصور الفضائية وفق التقنيات الحديثة وخصوصا تلك التي تعتمد على تقنيات الذكاء الصّنعي بطيئة بسبب الزمن الكبير المستهلك للقيام بذلك نتيجة تعقيد خوارزمياتها ومتطلباتها العالية جدّا في الحوسبة، اضافة لذلك يوجد طلب متزايد على تقديم سرعات عالية تلبيةً لرغبات الزّبائن اكثر من أي وقت مضى. الاكثر من ذلك مشكلة التعامل مع هذه البيانات تزيد بشكل كبير في حال كنا نتعامل مع الفيديو الفضائي الحديث او التطبيقات التي تتطلب معالجة في الزمن الحقيقي.

إنّ استخدام الطرق والانظمة التقليدية في معالجة بيانات الصور والفيديو الكبيرة في تطبيقات ذات طبيعة مشابهة لما تمّ التّطرّق اليها سابقا يصطدم بمشكلتين رئيسيتين هما بطئ التنفيذ الناتج عن الحجم الكبير للبيانات من جهة وعدم قابلية التّوسّع من جهة أخرى. من هنا نجد الحاجة الكبيرة الى موارد وعتاد مكلف ومُخصّص لأداء هذه المهمّة

بسبب السّعة العالية والسّمات عالية الوضوح للبيانات وهذه يفرض عبئاً على البنية التحتية المُستثمرة في أثناء المعالجة [2].
وليست الكلفة هي المشكلة الوحيدة فقط في هذه الانظمة وانّما نجد أيضا الصعوبة العالية في قابلية توسعة الموارد عند الحاجة لأنّ طريقة توسّعها يعتمد المنهج العمودي المُرتكز اساساً على تحديث مواصفات الاجهزة بزيادة ذواكرها ومعالجاتها الحسابية وهذا يتطلب ايضا اعادة تشغيل الخدمات العاملة من جديد عند تعديل العتاد المُستخدم، اضافةً الى ذلك نجد مشكلة نقطة الفشل الواحدة(Single point of failure) التي تعني حدوث فشل ما أثناء تنفيذ ومعالجة التطبيق على الجهاز العامل من دون وجود بديل او اجهزة احتياطية تستأنف تنفيذ التطبيق، وهذا يجعل العمل ينهار بشكل كامل حيث يتمّ اللّجوء الى اعادة تشغيل التّطبيق مرّة أخرى أو الانتظار ريثما يتم اصلاح سبب المشكلة وبالتالي المزيد من التكاليف الاضافية.

مما سبق نرى أنّه اصبحنا بحاجة للانتقال الى استخدام نظم ذات أداء عالٍ قادرة على استيعاب الحجوم الكبيرة للبيانات ومعالجتها بما يتوافق مع حاجات المستثمرين لناحية السّرعة في ايصال النّتائج المرغوبة وخصوصاً في التطبيقات التي تتطلب استجابة زمنيّة سريعة جدّا او تطبيقات الزّمن الحقيقيّ، الأكثر من ذلك يجب أن تكون هذه النّظم قابلة للتوسّع بمعنى قادرة على التكيّف مع امكانيّة تضخّم البيانات مستقبلاً وتتكيّف الى حدّ ما مع حالات الفشل.

تُعتبر تقنيات البيانات الكبيرة ومنصّاتها من أهم الأدوات المُستخدمة في تلبية الاحتياجات السابقة من ناحية قدرتها على استيعاب التضخم الكبير في حجوم البيانات وفي استثمارها للنّظم المُوزّعة من خلال المعالجة التّفرّعيّة لتحقيق الأداء المرغوب، وعلى الرّغم من التّحديات الموجودة لتحقيق زمن استجابة جيدة بالنسبة للمستثمرين عند معالجة هذه البيانات باستخدام التّقنيات المذكورة فإنّه يمكن تجهيزها بطرق تساعد في تخفيض الزمن المستهلك أثناء تنفيذ التطبيقات بشكل كبير اذا ما تم إعدادها بالشكل المطلوب.

2- مشكلة البحث

أصبحت تطبيقات معالجة الصّور الرّقميّة المُلتقطة عبر أجهزة الاستشعار عن بعد ذات تعقيد كبير، كما أنّ البيانات المُولّدة عن طريق هذه الأجهزة تزداد من ناحية الحجم وتتنوّع أيضاً لتشمل الفيديو الفضائي اضافة للصور، وعلى الرّغم من التّطوّر الكبير في تقنيات التّخزين والحوسبة التقليديّة التي قدّمت امكانيّات واسعة من ناحية سرعة معالجة البيانات وتخزينها لا تزال هذه التقنيات تعاني من صعوبة كبيرة في تلبية حاجات المستثمرين لناحية قابلية اضافة بيانات جديدة باستمرار ومن ناحية تخفيض زمن التنفيذ ايضاً، وتحديداً في التطبيقات التي تطلب قدرات حوسبة عالية جدّاً كتقنيات الذكاء الاصطناعي التي تستثمر خوارزميات تعلّم الآلة والتعلّم العميق أو تطبيقات الزمن الحقيقي.

انّ المشاكل المذكورة آنفاً تمثل تحدّيات حقيقية للنظم والتقنيات التقليديّة و يحتّم علينا الانتقال الى أدوات وتقنيات أخرى تستطيع تلبية حاجات المستثمرين بمعالجة البيانات بسرعة كافية لايصال النتائج بزمن مقبول يتلائم مع طبيعة التطبيقات المُستثمرة و تكون قادرة ايضاً على استيعاب الحجوم الكبيرة والمتزايدة لهذه البيانات.

3- هدف البحث

تلقى ابحاث استخدام تقنيات البيانات الكبيرة ونظم الحوسبة فائقة الأداء في معالجة تطبيقات صور الأقمار الصناعية رواجا مطردا واهتماما بالغاً بحثياً وعلمياً خصوصا ان مستقبلاً كبيراً ينتظر هذه التقنية عند استخدام تقنيات التّعلّم العميق أو في حال توفّر الفيديو الفضائي.

تحقّق هذه الدّراسة في استخدام تقنية/تقنيات أو نظم ذات أداءٍ عالٍ تستطيع تنفيذ تطبيقات معالجة الصّور المُلتقطة بالأقمار الصّنعيّة بزمن يلبّي حاجات المستثمرين وتتمتّع بالقدرة على استيعاب حجوم بيانات كبيرة ومتزايدة، وفي نفس الوقت ستقيّم الدّراسة أداء عمل هذا النظام/النظم وامكانية استثماره على أرض الواقع. وقد اعتُمد تطبيق تصنيف الصّور

المُرتكز على تقنيات التعلّم العميق باستخدام الشّبكات العصبونية الالتفافيّة (CNN) على مستوى الكائن (object) كمثال عمليّ وتجريبيّ.

للبحث تطبيقات واسعة في تصنيف الغطاء الأرضي واكتشاف التغيرات ويمكن ان يكون له دور كبير في تصنيف المناطق المتضررة سكنيا وعلى مستو البيئة والمحاصيل الزّراعيّة في سوريا عند توفّر البيانات المطلوبة حيث يمكن استثمار هذه الدّراسة في الحصول على نتائج سريعة لو تمّ تطبيق منهجيّاتها على بنية مناسبة للعمل. وسوف نرى أنّ المنهجيّة المُطبّقة في هذا البحث لا تقتصر على صور الأقمار الصّناعية فحسب وانّما يمكن أن تُنجز على تطبيقات و صور ذات طبيعة مختلفة، كما ينسحب الأمر على بيانات كبيرة أخرى كالنّصوص والسّجلّات المتنوّعة.

سيكون من ضمن أهداف هذا البحث تقييم أداء عمل منصّات البيانات الكبيرة Hadoop و Spark في تطبيقات معالجة الصور المذكورة في عمليتيّ التّدريب والتنبّؤ حيث سيكون معامل زمن التّنفيذ معياراً أساسياً في مراقبة الأداء.

4- مواد وطرق البحث

استُخدم نمّوذج الU-net الشّهير في هذا البحث، وهو أحد نماذج التّقطيع الدّلالي التي تعتمد على الشّبكات العصبونيّة العميقة في تصنيف الصّور، حيث استعرض البحث توصيفاً لطبقات ومعمارية هذا النّموذج وآليّة عمله بحيث نحصل من خلاله على خرائط التّقطيع من صورة دخل عند القيام بعمليّة التّنبؤ. كما تناول البحث أهمّ منصّات البيانات الكبيرة وناقش اختيار المنهج الأفضل لمعالجة التطبيق المطلوب. ومن بين ما تمّت دراسته كان استخدام أُطر عمل تفرّعيّة مُخصّصة لأداء المعالجة التّفرّعية للبيانات على منصّات البيانات الكبيرة. أيضاً تطرّق البحث الى مجموعات البيانات المُستخدمة وطبيعتها وكيفيّة معالجتها.

مرّ هذا البحث بالمراحل التالية:

- تحديد الأداة الأفضل لمعالجة البيانات، واختيار النّموذج البرمجّي التّفرّعيّ الملائم
- اختيار وتركيب منصة بيانات كبيرة مناسبة للمعالجة التّفرعيّة لتطبيقات الصّور
- استخدام آلية للاستحواذ على هذه البيانات واجراء معالجة أوّلية لتوليد مجموعة جديدة وكبيرة من الصّور الجاهزة للمعالجة الفعليّة في التّطبيق
- استخدام النّظام المُقترح لتصنيف صور الأقمار الصّنعيّة على بيانات منطقة جغرافيّة مُحدّدة متاحة بشكل مجّاني باستخدام منصّتي البيانات الكبيرة هادووب وسبارك
- مقارنة وتحليل النتائج التي تمّ الحصول عليها باستخدام النّظام المُقترح

4-1- الدّراسة المرجعيّة

في الآونة الأخيرة، وُجدت عدّة دراسات تناولت كيفيّة استخدام النّظم فائقة الأداء في معالجة صور الأقمار الصّناعيّة باستخدام منصّات وتقنيات البيانات الكبيرة من خلال استثمار إمكانيّات أجهزة متعدّدة واستثمار كل من الحوسبة المُوزّعة وقدرات التّخزين لكل منها في تنفيذ التّطبيقات المختلفة بسبب قدرتها العالية على استيعاب الكمّيات الكبيرة والمتزايدة من البيانات و بغيّة الحصول على أداء أفضل من ناحية زمن التّنفيذ وسنستعرض فيما يلي بعضاً من أهمّها.

ناقشت الدّراسة [3] مشكلة الحجوم الكبيرة والمتزايدة لصور الاقمار الصّناعيّة الكبيرة وصعوبة الوصول اليها ومعالجتها على مخدّمات مركزيّة، واقترحت استخدام النّموذج البرمجي MapReduce[4] لأجل هذه الغرض مع أخذ تطبيق اكتشاف الحواف كمثال عمليّ، حيث تمّ تقسيم الصّورة الى مُقتطعات وأجزاء ويُعطى كل جزء الى جهاز مُعيّن لتتم معالجته وفق التّطبيق المطلوب تنفيذه. ولأجل ذلك يُقسَّم التّطبيق الى مهمّات تُسند الى العقد العاملة بحيث تُنجز كل مهمّة جزءاً من العمل الكلّيّ بشكل تفرّعيّ على التّوازي، بعد ذلك تُجمع النّتائج مع بعضها البعض وتدمج لتشكيل الخرج النّهائيّ. هذا وقد تناولت الدّراسة آليّة تقسيم الأدوار بين كل من عمليّتيّ Map و Reduce وتنسيق تبادل البيانات فيما بينهما للحصول على النّتائج المرغوبة.

أظهرت النّتائج تحسيناً كبيراً في زمن التّنفيذ للنظام المُقترح بالمقارنة مع منهجيّة الجهاز الواحد وكفائة منهجيّة النظام التّفرّعي في انجاز العمل المطلوب على التّوازي.

ولا تقتصر امكانيّة استخدام هادووب بنمّوذجه البرمجيّ التّفرعيّ على التطبيقات البسيطة، بل تتجاوز ذلك لتنفيذ تطبيقات تعلّم الآلة، وقد بيّن البحث [5] كيف يمكن استثمار هادووب في معالجة بيانات الاستشعار عن بعد لاستخراج السّمات وتصنيف الصّور واستخدموه في معالجة صور الأقمار الصّناعيّة الكبيرة لتحسين اداء وكفائة العمل. قام الباحثون بتحقيق خوارزميّة شجرة القرار لتصنيف الصّور اعتمادا على قيم المتوسّط والتّباين والمسافة الإقليديّة للبكسلات بشكل تفرّعيّ باستخدام MapReduce، حيث تمّ تقطيع الصّورة الواحدة الى كُتل متساوية الحجم ومعالجة كل منها بشكل منفرد ليتمّ في النّهاية تجميع الخرج وتكوين الصّورة النّهائيّة، وقد بيّنت النّتائج أداءً مُنخفضاً للنظام عند معالجة عدد قليل من الصّور وجيّداً عندما كان عددها كبيراً.

تناولت دراسات أخرى تطبيقات معالجة صور ذات طابع مُختلف، فقد تناول الباحثون في [6] تحقيقاً لخوارزميّة العنقدة الشّهيرة K-Means في تصنيف صور الأقمار الصّناعية باعتبارها خوارزميةً تتطلّب قدرات حوسبة عالية وخصوصا عندما يكون حجم الصور كبيراً أو أنّ التّطبيق يحتاج لنتائج خلال وقت محدود جدّاً، ولكون هذه الخوارزميّة ذات طبيعة تكراريّة فإنّ اتّباع المنهجيّة المُستخدمة في [3] و [5] اعتماداً على النّموذج البرمجيّ MapReduce لن يكون مناسباً أبداً بسبب الزّمن الكبير المُستهلك في عمليّات القراءة/الكتابة من/الى نظام الملفّات المُوزّع أثناء تسليك البيانات بين عمليّتيّ Map و Reduce، يُضاف الى ذلك الوقت الاضافي النّاتج من عبئ عمليّات النّسخ المتماثل على كتل البيانات في نظام الملفّات المُوزّع أثناء الكتابة على وسائط التّخزين. بسبب ما سبق فقد اختِيرَ Spark بدلاً عن Hadoop لتجنّب عمليّات القراءة والكتابة التّكراريّة ونُفّذ التّطبيق على عنقود في بيئة حوسبة سحابيّة واختُبر أداؤه بإجراء التجربة عدّة مرّات مع تغيير عدد تكرارات خوارزمية الK-Means وقُيّم أداء النّظام المُقترح مع ال K-Means العاديّة من ناحية زمن التّنفيذ وبيّنت النّتائج أداء عالٍ بشكل كبير لل K-Means التّفرعيّة بالمقارنة مع ال K-Means العاديّة لأجل تكرارات كبيرة.

ليس بعيداً عن الخوارزميّات التّكراريّة و عند معاينة تطبيقات تصنيف أخرى لا بدّ أن نصادف كل من خوارزميّة آلة أشعّة الدّعم SVM[7] و الشبكات العصبونية MLP[8] اللّتين تتميّزان بدقّة أكبر في تصنيف الصّور، انّ كلتا الخوارزميتين مستنزفتين للموارد بشكل كبير خصوصاً اذا كانت البيانات كبيرة الحجم ومتعدّدة الخصائص والسّمات، وهذا ما تطرق اليه البحث [9] حيث استُخدم نظام هادووب للملفّات المُوزّعة لتخزين بيانات الصّور وسبارك كمحرّك تنفيذ لمعالجة صور أقمار صناعيّة عالية الدّقّة باستخدام كل من SVM و MLP. وقد اعتمد منهج البحث على القيام بمعالجة أوّليّة للصّور باعتبار كل بكسل من بكسلات الصّورة المُستخدمة عيّنة تدريب مفردة ثمّ دُرّب النّموذج باستخدام الخوارزميّتين السّابقتين وأُجري تقييمه على بيانات اختبار. بيّنت النّتائج تحسّن في زمن التّنفيذ عند استخدام النّظام المُقترح.

بدراسة الأبحاث السّابقة [3]، [5]، [6] و [9] وجدنا أنّه تم تصنيف الصّور اعتمادا على قيمة البكسل بشكل منفرد، ومعنى ذلك أنّه تم تصنيف البكسل بشكل مستقلّ عن بقية البكسلات المجاورة اعتماداً على قيمه الطّيفيّة فقط دون الاستفادة من قيم البكسلات المجاورة له، وهذا الأسلوب يعطي أداءً جيّداً عند استخدامه على صور منخفضة أو متوسّطة الدّقّة لكنّه ليس مناسباً للصّور ذات الدّقة العالية. إضافة لذلك فإنّ تقنيات التّعلّم العميق هي أكثر تقدّماً من الطّرق السّابقة (بما فيها خوارزميّات تعلّم الآلة) لكونها تستخدم الشّبكات العصبونيّة الالتفافيّة CNN ذات الدّقّة العالية في مجال الرؤية عبر الحاسب والقادرة على استنباط سمات معقّدة للغاية تسهم الى حد كبير في نتائج أفضل لتطبيقات تصنيف الصّور.

من خلال المسح الذي قمنا بإجرائه مؤخّراً على العديد من الدّراسات في مجال معالجة صور الأقمار الصّناعيّة على النّظم فائقة الأداء وتقنيات البيانات الكبيرة بما فيها الدّراسات المذكورة آنفاً لم نجد أبحاثاً تناولت بشكل واف تطبيق تقنيات التّعلّم العميق على هذه الصّور باستخدام هذه المنصّات، وخصوصاً تلك التّي تقدّم دقّات كبيرة للصور عالية الوضوح، لذلك سعينا في هذا البحث الى المساهمة في هذا المجال من خلال دراسة تحقيق تقنيات التّعلّم العميق والشّبكات العصبونيّة الالتفافيّة اعتماداً على مبدأ الكتلة أو

دفعة بكسلات معاً في معالجة صور الأقمار الصّناعيّة على هادووب و سبارك، كما ناقشنا النّظام المُقترح وآليّة تقييم العمل المُنجز مع النّتائج التّي تم الحصول عليها.

4-2- المعالجة التّفرّعيّة لتطبيق تصنيف الصّور

إنّ كل من عملية تدريب النّموذج أو التّصنيف تتطلب قدرات معالجة عالية خصوصا مع بيانات ذات حجوم كبيرة، الا أنّ الحاجة الى الموارد العالية في تدريب النّموذج هي أكبر بكثير من تلك المطلوبة عند القيام بعمليّات التّنبؤ، كما أنّ تحويل الرمّاز المصدري في عمليّة التدريب من الشّكل التسلسلي الى الشّكل التّفرّعيّ ليس بالأمر السّهل قياسا بعمليّة التّنبؤ على بيانات جديدة لذلك تمّت دراستها بشكل أوسع في هذا البحث.

4-2-1- التّدريب المُوزّع

يتمّ تدريب الشّبكة العصبونيّة تفرّعيّاً وفق نفس المبدأ المُتّبع تسلسيّاً عبر خوارزميّة الهبوط المتدرّج العشوائي(SGD) Stochastic Gradient Descent وهي الخوارزميّة الاكثر شيوعا أثناء التّدريب في مجال الذّكاء الصّناعي ومُشتقّة اساسا من خوارزمية الهبوط المتدرّج Gradient Descent .

ترتكز ال(Gradient Descent) في عملها على ايجاد قيمة المتحول او المتحولات (w) التى تجعل خرج تابع الخطأ المبُيّن في المعادلة (1) أقل ما يمكن. ويمثل تابع الخطأ الفرق بين قيمة الاجابة الفعلية لعينة دخل ما (Y) وقيمة الاجابة المُتنبّئ بها(Y^)

$$Q(w) = \frac{1}{n}\sum_{i=1}^{n} Q_i(w),$$

المعادلة (1) الشكل العام لتابع الخطأ

يشير المتحول (i) الى دليل عينة التدريب الواحدة وبالتالي فان المعادلة السابقة تشير الى مجموع الاخطاء بين القيم المُتنبئ بها والقيم الحقيقية لبيانات التدريب، ويمكن تلخيص آلية عمل الخوارزمية بالخطوات التالية:

1- إعطاء قيم أولية للمتحولات (w)

2- حساب قيمة التنبؤ لعينات التدريب بتعويض قيم الدخل في تابع التّنبؤ ($\hat{Y}$)

3- حساب تابع الخطأ للعينات من خلال المعادلة (1)

4- حساب قيمة مشتق تابع الخطأ $\nabla Q_i(w)$ بالنسبة للمعاملات

5- تحديث قيم معاملات (w) وفق المعادلة التالية(2):

$$w := w - \eta \nabla Q(w) = w - \frac{\eta}{n} \sum_{i=1}^{n} \nabla Q_i(w),$$

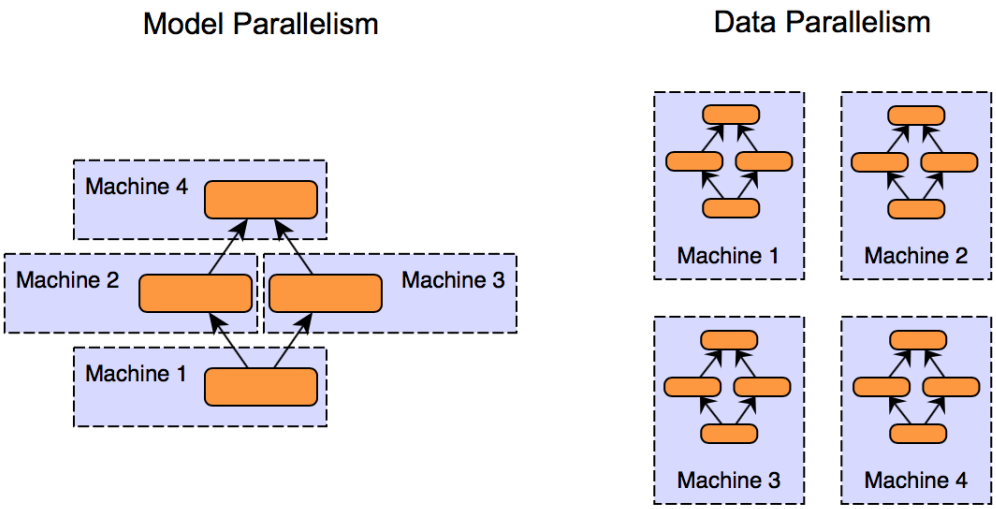
المعادلة (2) تحديث المعاملات

6- تكرار الخطوات من 2 الى 4 عددا من المرّات حتى الوصول الى التقارب (أقل قيمة ممكنة لتابع الخطأ)

تستخدم خوارزمية الهبوط المتدرّج العشوائي(SGD) نفس الآلية السابقة في ايجاد المعاملات المطلوبة (w) وبنفس الخطوات مع فرق أنه يتم أخذ عدد معيّن من عيّنات البيانات تُسمّى دُفعة في كل تكرار عوضاً عن أخذ كل العيّنات بحيث تُؤخذ بشكل عشوائي(Stochastic) وتُطبق الخطوات السابقة آنفاً ثم تتم متابعة العمل على الدفعة التالية.

ان الهدف النّهائي لهذه الخوارزمية هو ايجاد افضل قيم ممكنة للمعاملات التي تمثل أوزان الشبكة العصبونية، وكما رأينا فانّها وفق المبدأ التسلسليّ تتضمّن عدّة جولات من العمل حيث أنّ نتائج كل جولة من التّدريب تُضمّن في النّموذج المُدرّب لتُستخدم في الجولة التّالية، و تُعالج بيانات كل جولة تسلسلياً وفق منهجية الجهاز المنفرد.

أما بالنسبة للتدريب المُوزّع فإنه توجد منهجيّتين معروفتين لتدريب النّموذج بشكل تفرّعي على مجموعة من الأجهزة وهي كما موضّحة في الشكل(1) .



الشكل(1) منهجيّة تفرّعيّة النّموذج مقابل تفرّعيّة البيانات في التّدريب المُوزّع [10]

4-2-1-1- منهجيّة تفرّعيّة النّموذج

بعض نماذج الشّبكات العصبونية كبيرة لدرجة أنّه لا يمكن لذاكرة جهاز واحد (او ذاكرة معالج الرّسوميّات GPU Graphical Processing Unit) أن يستوعب حجمها، لذلك استخدام هذه المنهجيّة هنا هو الحلّ المناسب ومن الأمثلة على ذلك الشبكة العصبونيّة لنظام الترجمة الخاص بغوغل [10].

إنّ تدريب نماذج من هكذا أنواع يتطلّب تقسيم النّموذج الى عدة أقسام على عدّة أجهزة بحيث يتمّ التّدريب على التّوازي وكل جزء يتم تدريبه على عقدة واحدة باستخدام مجموعة البيانات بكاملها، معنى ذلك أنّ طبقات مختلفة من الشّبكة العصبونيّة يتمّ تدريبها على عدّة أجهزة او (GPUs) في نفس الوقت. عند انتهاء جميع العقد من تدريب الأجزاء المخصصة لها يجري تجميع النتائج وفق عملية معيّنة لنحصل في النهاية على النّموذج المُدرّب النّهائيّ. تُسمّى هذه الطّريقة أحيانا في إطار عمل TensorFlow بالنّسخ

المتماثل داخل البيان (in-graph replication) وتُعتبر من الأساليب الصّعبة التّطبيق للحصول على أداء جيّد .

4-2-1-2- منهجيّة تفرّعيّة البيانات

تُعرف هذه المنهجيّة في إطار عمل TensorFlow بالنّسح المتماثل بين البيان (between-graph replication) وفيه يُستخدم النّموذج على كل عقدة لتدريب بيانات مختلفة بخلاف الطريقة السابقة التي فيها يُدرّب جزء من النّموذج على كل البيانات.

تحسب كل عقدة المشتقّات "gradients" الناتجة من حساب الأخطاء في عملية الهبوط المتدرّج (Gradient Descent) لجزء من دفعة بيانات واحدة في كل مرّة، ثمّ يتم تجميعها مع نتائج العقد الباقية في كل تكرار من الخوارزمية كما لون أن العمل يجري على عقدة واحدة حيث أنّه يتعيّن على كل عقدة ان ترسل التغيّرات الى جميع العقد الأخرى. يتم بعد ذلك تحديث المعاملات (اوزان الشبكة العصبونية) عند كل عقدة عاملة وتُكرّر العملية على الدُفعات التالية من البيانات حتى الانتهاء منها.

4-2-1-3- معماريّات التّخاطب والاتّصال في عمليّة التّدريب

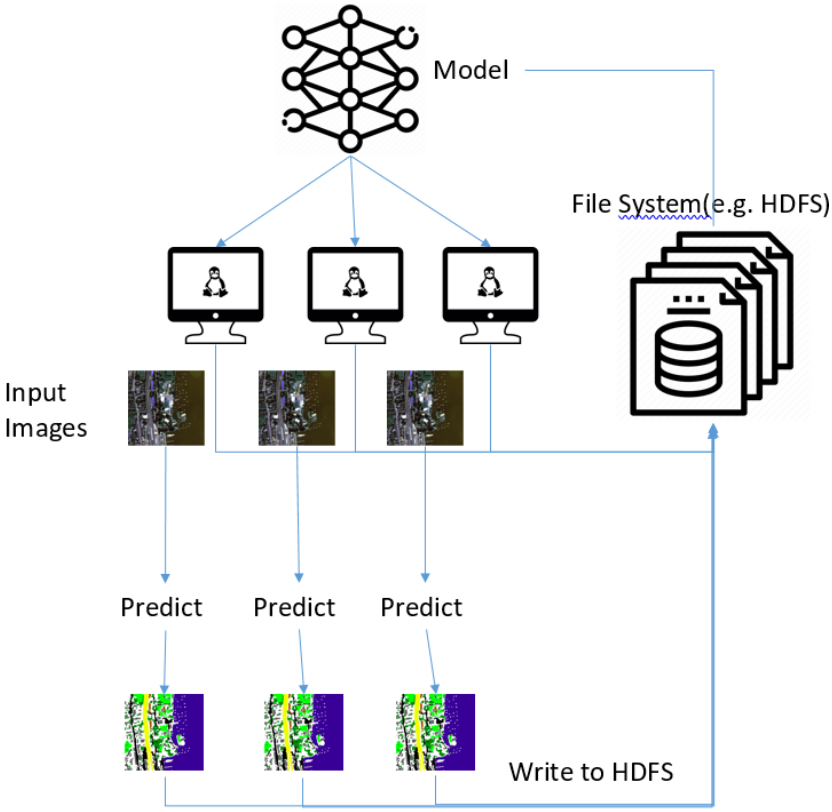
توجد معماريتان لتدريب نمّوذج التّعلّم العميق على التّفرّع الأولى تُدعى **مخدّم المعاملات المركزية** حيث توجد عقدة مركزيّة تعمل كمخدّم يتولّى تحديث معاملات النّموذج واجراء بثّ عام لها الى كافة العقد العاملة. في النّمط المتزامن منها ينتظر المخدّم انتهاء العقد العاملة من حساب المشتقّات وارسالها اليه ليتولّى بدوره حساب قيم متوسطاتها وتعديل أوزان الشبكة العصبونية ثم يرسلها من جديد الى جميع العقد الأخرى مع دفعة جديدة من البيانات لبدء تكرار جديد من خوارزمية ال(SGD)، أمّا في النّمط غير المتزامن فلا تنتظر الأجهزة تحديثات النّموذج من المخدّم في كل دفعة بيانات، بل حيث تعمل بشكل مستقل عن بعضها وتتشارك النّتائج فيما بينها.

المعمارية الثّانية هي الحلقة (Ring) التي تكون فيزيائية او منطقية ضمن عنقود العقد العاملة وفيها لا يوجد مخدّم مركزي لتحديث اوزان الشّبكة العصبونية وانّما تقوم كل عقدة بحساب مشتقات الجزء الخاص بها من دفعة البيانات وارسالها الى العقدة التّالية في

الحلقة واستقبال المشتقّات من العقدة السّابقة في الحلقة، ويجري بعد ذلك تحديث معاملات النموذج على كل عقدة. وبالتالي من أجل N عقدة عاملة ضمن الحلقة، ستكون جميع العقد قد استلمت ال gradients من جميع العقد الأخرى بعد N-1 عملية نقل بحيث تشمل العمليّة الواحدة ارسال واستقبال البيانات. تُعتبر هذه المعماريّة مثاليّة من ناحية استهلاك عرض الحزمة لكونها تضمن استخدام كامل لعرض حزمة الشبكة خلال رفع وتحميل البيانات على كل عقدة.

4-2-2- التّنبّؤ المُوزّع

لكي نتمكّن من إجراء التّنبّؤ المُوزّع بشكل تفرّعيّ فينبغي أن يتم تحميل النّموذج المُدرّب مُسبقاً الى ذاكرة كل عقدة في العنقود سواء من نظام الملفّات المحلّي للعقدة أو من نظام الملفّات المُوزّع كما في الشكل(2) وفيه تقرأ كل عقدة جزءاً من البيانات فقط تعالجها بشكل مستقلّ بحيث تدخل كل صورة الى النّموذج لتُعالج وينتج من ذلك خريطة التّقطيع الموافقة لصورة الدّخل لتتم كتابتها الى نظام الملفّات المُوزّع.



الشكل(2) آليّة التّنبّؤ الموزّع أثناء تصنيف الصّور

4-3- بنية نمّوذج U-Net

تتألّف معمارية شبكة ال U-Net كما يبيّن الشكل(3) من جانبين، أيسر يُسمّى مسار التّقليص وأيمن يسمّى مسار التّوسيع. يتبع مسار التقليص المعمارية النموذجيّة لشبكات الطيّ او الالتفاف (convolution) ويتألّف من التّطبيق المتكرّر للالتفاف بمصفوفات ذات أبعاد 3*3 من دون حشو، ويعني ذلك انّ أبعاد مصفوفات الخرج مختلفة عن الدّخل، متبوعاً بتابع تفعيل من النّوع Rectified Linear Unit(RLU) أو تابع التصحيح الخطّي الذي يعطي خرجاً مساوياً للدخل اذا كان موجباً، و صفرا اذا كان سالباً. كما يتضمّن النموذج عمليّة تجميع أعظمي بمصفوفة من قياس 2*2 وخطوة 2 لأجل عمليّة الاختزال [11].

في كل خطوت الاختزال (Down sampling) نقوم باختزال عرض وارتفاع المصفوفات وبمضاعفة عدد القنوات.

كل خطوة من مسار التّوسيع تتالّف من:

1- تضخيم خرائط السّمات (feature maps) متبوعة ب بطبقة التفاف (convolution) 2*2 تخفّض عدد قنوات السّمات الى النّصف
2- دمج الخرج الناتج من الخطوة 1 بخريطة السّمات المقابلة لها من مسار التّقليص وذلك بعد تعديل مقاسات ابعادها من خلال القطع لتوافق عملية الدّمج
3- طبقتي التفاف 3*3 متتاليتين متبوعا بتابع التنشيط (RELU) لكل منهما

في آخر طبقة توجد عملية التفاف 1*1 تُستخدم لمقابلة خرائط السّمات مع عدد الاصناف المُختارة سابقا ولكي تكون خرائط السّمات في الخرج النهائي للنموذج المُدرّب ملتحمة من دون تشقّقات في الالتحامات فإنّه من المهم اختيار ابعاد قياسات صور الدخل بحيث تُطبّق عمليّات التجميع الاعظمي 2*2 على مصفوفات بابعاد زوجية [11].

نظراً لأنّ عملية الالتفاف(Convolution) المُستخدمة في استخراج السّمات أثناء معالجة البيانات في الشبكة العصبونيّة لهذا النّموذج هي من النوع الذي لا يحوي حشو فإنّ صورة الخرج تكون ذات أبعاد أقل من صورة الدّخل بمقدار ثابت بالحالة المعيارية لتطبيق النّموذج، الّا أنّه في حالتنا هذه استخدمنا نمّوذج مُعدّل من U-Net بحيث استُخدمت طبقة التفاف مع حشو للمحافظة على أبعاد صورة الخرج، مما يعني ان صورة الدّخل وصورة الخرج النهائية لهما نفس الأبعاد.

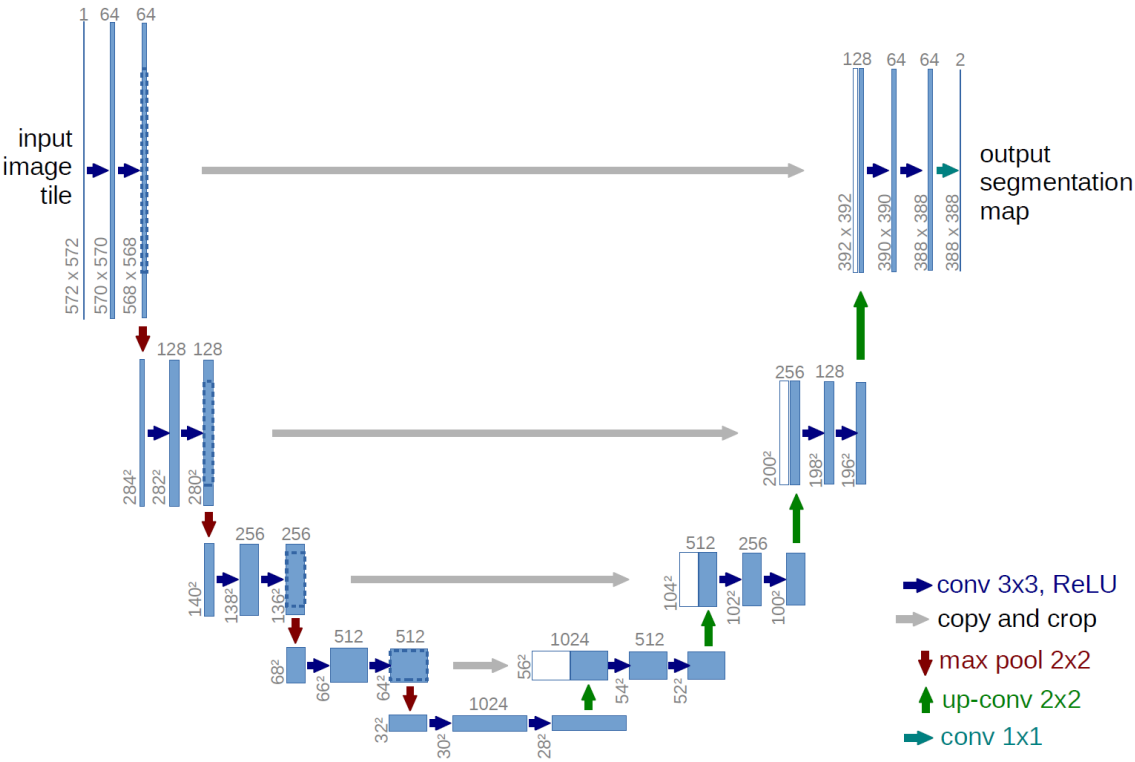
بالمجمل احتوت معمارية نموذج الU-Net المُستخدم في هذه البحث على **23 طبقة** من طبقات الالتفاف، وتتضمّن هذه الطّبقات ما يبلغ **31،053،225** معامل (بارامتر) . وهو من النّماذج الكبيرة نسبيّاً التّي تتطلب وقتا طويلا للقيام بعمليّة التّدريب خصوصا اذا كانت بيانات التدريب كبيرة الحجم.

تمّ اقتراح هذا النمّوذج للعمل عليه لعدّة اسباب وهي:

1- أحد نماذج تقنيات التّعلّم العميق ذات الأداء الجيّد للقيام بعملية تصنيف الصّور وفق الخرج المطلوب بدقّة تصل الى 95.40 لبيانات التّدريب و 83.354 لبيانات الاختبار

2- إمكانية التّطبيق على صوركبيرة الحجم من خلال تقطيع الصورة الواحدة الى أجزاء بحيث كل جزء يدخل الى النّموذج ليتم تدريبه بشكل منفصل وهذه الطّريقة مناسبة جدّا اذا كانت الصورة كبيرة جدّا (من فئة الغيغا بايت)

3- لا توجد حاجة لمجموعة كبيرة من الصّور للقيام بالتدرّب عليها لانّه يمكن استخدام صورة واحدة وتقطيعها الى قطع متداخلة مع بعضها البعض لتوليد مجموعة كبيرة من أمثلة التدريب التّي تدخل نموذج التّدريب



الشكل(3) معمارية ال U-Net[11]

4-4- استخدام سبارك بدلا عن هادووب كمحرّك أساسيٍّ للتّنفيذ

على الرّغم من أنّ هادووب هو المنصّة الأشهر في مجال معالجة البيانات الكبيرة بسبب تقديمه كل من نظام الملفّات المُوزّع ومعمارية MapReduce للمعالجة التفرّعيّة المُخصّصين أساساً للعمل عليه واللّذين يعملان بكفائة مع العديد من المهام الّا أنّه يعاني من مشكلة الأداء البطيئ والسيّى أحيانا من أجل التطبيقات التكراريّة أو التفاعليّة، وهي تطبيقات تحتاج للوصول الى البيانات بشكل متكرّر أثناء عمليّات المعالجة سواء كانت البيانات نفسها التي تمّت قرائتها أول مرّة او بيانات نتجت من عمليّات معالجة في خطوة سابقة مثلاً.

تُعرف هذه المشكلة أحياناً بمشكلة البطئ في تشاركيّة البيانات في نمّوذج MapReduce على هادووب بسبب الوقت المُستهلك في عمليّات القراءة والكتابة من/الى نظام الملفّات بالإضافة الى عمليّات تسليك البيانات (Serialization) و التّضاعف المتماثل في نظام الملفّات المُوزّع، ووفقا لقيم الأداء للعديد من تطبيقات MapReduce فإنّ 90% من وقتها مُستهلك في عمليّات القراءة و الكتابة من/الى HDFS [12] [13].

وعلى النقيض من ذلك يستخدم سبارك ما يُعرف بمجموعات البيانات المرنة (Resilient Distributed Dataset) التي المعروفة اختصاراً ب RDD التّي تقدّم خدمة وضع البيانات في الذاكرة مباشرة لتتم معالجتها وتُسمّى احياناً بالحساب – في– الذّاكرة. يعني ذلك أنّه يتم تخزين حالة الذّاكرة ككائن (object) تتم مشاركته بين الأعمال أو التّطبيقات الأخرى العاملة. هذه التّقنية هي أسرع ب 10 الى 100 مرّة من تلك التّي تعمل اعتمادا كلّيّاً على التّناقل عبر الشّبكة ووسائط التّخزين كما في هادووب لذلك اعتُمد استخدام سبارك في عمليّات تنفيذ المعالجة الفعليّة للبيانات فيما بقي استخدام نظام الملفّات الموزّع الخاص بهادووب للتخزين.

4-5- التّدريب المُوزّع على سبارك

تدعم واجهات برمجة التطبيقات (API: Application Programming Interfaces) الخاصّة بسبارك العديد من خوارزميّات تعلّم الآلة بما في ذلك بناء وتدريب شبكات عصبونيّة بالإضافة الى API أخرى لعمليّة التنبّؤ، الّا أنّه يفتقر الى الأدوات والمكتبات المطلوبة التّي تمكّنه من التّعامل مع تقنيات التعلّم العميق لتلبية احتياجات التّطبيقات، من هنا قامت عدّة شركات بالاضافة الى مجتمع البيانات الكبيرة بملئ هذا الفراغ وتقديم عدّة مكتبات لحل هذه المكشلة.

من بين عدّة أدوات وبرمجيّات تدمج تقنيات التّعلّم العميق مع سبارك فأنّ عدداً قليلا منها فقط يدعم تدريب الشّبكات العصبونيّة العميقة على التّفرّع، هنا نجد مكتبات TensorFlowOnSpark[14] ، Horovod[15] ، BigDL[16] ، و Deeplearning4j[17] أمثلة عن هكذا أدوات تتيح استخدام تقنيات التّعلّم العميق على الانظمة التفرّعيّة بشكل سهل وعمليّ.
لأجل بحثنا هذا اخترنا إطاري عمل TensorFlowOnSpark و Horovod لأن كليهما يستطيعان التّعامل بشكل فعّال مع مكتبة TensorFlow الشّهيرة لبناء الشّبكات العصبونيّة وتدريبها باستخدام لغة بايثون مع امكانية استثمار العديد من المكتبات المساعدة الاخرى في العمليّات على المصفوفات، ايضا لا توجد تعقيدات كبيرة في اعداد بيئة العمل وتهيئتها.

4-5-1- إطار العمل تنسورفلو على سبارك (TensorFlowOnSpark)

إطار عمل مُطوّر أساساً من قبل شركة ياهوو للقيام بتدريب مُوزّع لنماذج التّعلم العميق بنطاق واسع على عناقيد هادووب ضمن البيئة السّحابية الخاصة بالشركة نفسها وتمكننا من تنفيذ تطبيقات TensorFlow على عناقيد Spark و Hadoop. ويعتمد إطار العمل هذا على خدمة غوغل للاتّصال عن بعد (gRPC: Google Remote Procedure Call) كتقنية لتبادل البيانات أثناء التّدريب وهي مشتقة من البروتوكل الاساسي في الاتّصال عن بعد (RPC: Remote Procedure Call) الذي يتكوّن من برنامجين هما المخدّم والزّبون بيحيث يعرض المخدّم خدماته على برنامج الزبون بشكل توابع برمجية

تُستدعى عن بعد [18]. وهذه التقنية هي الاساس الذي تعتمد عليه العقد العاملة ضمن العنقود في الاتصال وتبادل البانات فيما بينها.

4-5-2- إطار العمل هوروفود (Horovod)

إطار عمل مفتوح المصدر مُطوّر من شركة **Uber** و يُستخدم للتدريب المُوزّع لنماذج التّعلّم العميق ويعمل مع عدة تقنيات وأطر أخرى هي **TensorFlow** و **Keras** و **PyTorch** و **Apache MXNet**. اعتُمد في شركة Uber هذا الإطار على عدة تقنيات أثناء تطويره حيث اختير نمّوذج واجهة تمرير الرسائل المعروفة اختصارا باسم MPI لتبادل البيانات بين المعالجات الذي يُعتبر اسلوب مباشر وسهل التعامل معه [15].

4-6- تدريب النّموذج

في هذه العمليّة يكون لدينا صور الدّخل مع خرائط التّقطيع المُقابلة لها لتدريب الشبكة العصبونية بتطبيق الهبوط المتدرّج العشوائي.

يوجد معامل مهمّ ينبغي تحديده قبل البدء بعمليّة التّدريب وهو قياس دفعة البيانات الكلّيّ (Global Batch Size) سنرمز لها بGBS، ويُقصد به حجم الدّفعة التي ستتشارك العقد العاملة في معالجتها في وقت واحد ويمثّل مجموعة الدّفعات الجزئيّة التي تقوم كل عقدة بمعالجة واحدة منها فقط. يمكن أن نبقي قيمة الGBS عند التّدريب على التّفرّع مماثلا لقيمة حجم الدّفعة عند التّدريب على عقدة واحدة وهنا ستتقاسم العقد دفعة البيانات لتعالج كل منها جزء فقط، ويمكن أيضاً جعل قيمة الGBS تساوي قيمة حجم الدّفعة مضروبة بعدد العقد أي $GBS = N * (batch\_size)$، وهذه الطّريقة فعّالة للغاية لناحية أنّه يمكن استخدام حجم دفعة كلّيّة كبير جدّا ولا تستطيع معه العقدة الواحدة استيعابها عند المعالجة نظراً للحاجة الى ذواكر كبيرة جدّاً، ومن المعلوم ايضاً أنّ الدّقّة بهذه الحالة تصبح أفضل عادة مقارنة بالحالة الأولى لأنّ تحديث أوزان الشّبكة العصبونيّة سيجري بعد تدريبها على فضاء عيّنات أكبر.

إنّ هذه الميّزة تجعل من النّظم الموّزّعة تملك **قيمة مُضافة** الى جانب السّرعة في الأداء مقارنة بمنهج الجهاز الواحد نظراً لامكانيّة تدريب النّموذج باستخدام معاملات لم يمكن بالامكان استخدامها لو تمّ التّدريب بدونها.

4-7- العتاد والبيئة المُستخدمة

لبناء العنقود تم تجهيز مخدّم بالقدرات والموارد المُتاحة لانشاء عنقود من الاجهزة المتصلة مع بعضها البعض حيث تمّ انشاء 4 أجهزة افتراضية على مخدّم يعمل بذاكرة GB 64 ومعالج متعدّد النّوى يعمل بتردد GH 2.1 مُنصّب عليه بيئة افتراضية Xen Server 7.3، وتمّ تخصيص لكل آلة افتراضيّة معالج ثماني النوى وهذه الآت متصلة مع بعضها البعض بشبكة محلية افتراضية. المهم ذكره هنا أنّه تم تنصيب اتصال التشفير الآمن Secure-Shell(SSH) على جميع العقد العاملة لكي تتمكّن من الاتّصال وتبادل البيانات فيما بينها بشكل آمن لتجنّب القيام بعمليّة تسجيل الدّخول في كل عمليّة اتّصال، مع القيام بنسخ مفاتيح التشفير الى العقدة الرّئيسيّة، وهو أحد الاعدادات الاساسيّة اللّازمة لعمل كل من Spark و Hadoop.

لا يحوي المخدّم أي معالجات رسوميات Graphical Processing Unit(GPU) لذلك تم الاعتماد كلّيّاً على وحدة المعالجة المركزية للمخدّم، كما تمّ أيضاً تفعيل ميّزة تثبيت وحدة المعالجة المركزيّة (Virtual CPU Pinning) في البيئة الافتراضية للمخدّم وهي ميّزة تمكّن من مقابلة النّوى الفيزيائيّة للمخدّم مع نوى معالجات الآلات الافتراضية لتحقيق أكبر قدر ممكن من عزل الموارد بين تلك الآلات، وهو اعداد مهمّ في حالتنا هذه من أجل محاكاة نظم البيانات الكبيرة في بيئة افتراضيّة أُنشئت أساساً لتحقيق أكبر قدر من تشاركيّة الموارد.

بعدها تم تركيب منصة Spark على كل العقد وإعداداها بمتحولات البيئة اللازمة لعمل العنقود كما تمّ انشاء بيئة بايثون افتراضية نسخة 3.7 و تحميل المكتبات المطلوبة من أجل عملية التنّفيذ، ومن بين أهم ما تم تنصيبه مكتبة TensorFlow وإطاري عمل TensorFlowOnSpark و Horovod المستخدمين في عملية التّدريب الموزّع، فيما استُخدم

نظام الملفّات الموزع Hadoop بعد تركيبه على العقد جميعها من أجل تخزين بيانات التدريب وتخزين المودل وصور الخرج النهائية عند القيام بالتنبّؤ.

يوضّح **الجدول(1)** تفاصيل خصائص العقد العاملة في العنقود، وبعد الانتهاء من جميع الخطوات السّابقة تصبح البيئة جاهزة لتجريب البرنامج/البرامج المطلوب.

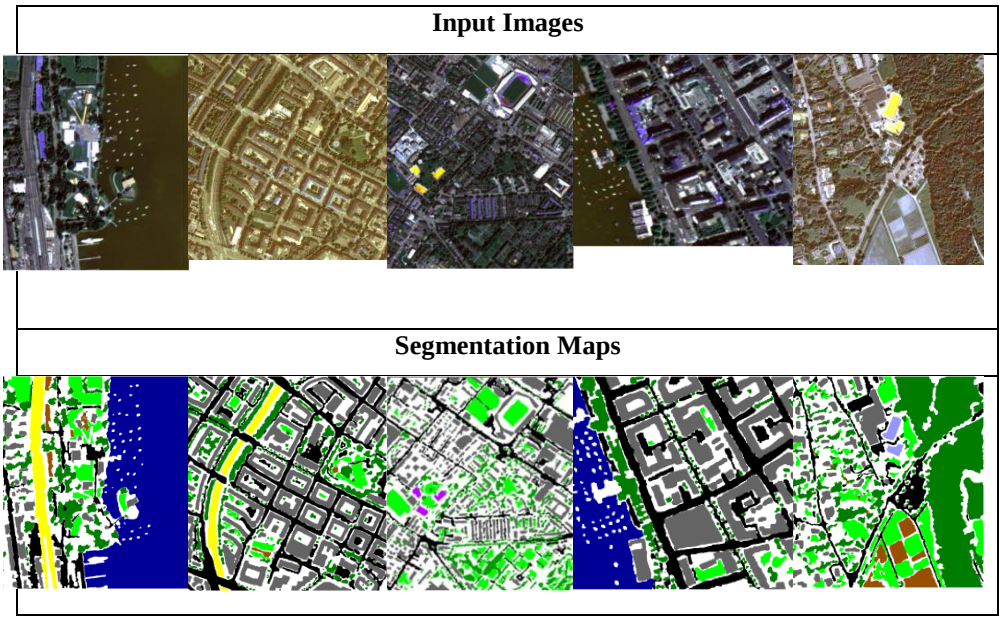
الجدول(1) خصائص العنقود والعقد العاملة ومواصفاتها الرّئيسيّة

عدد العقد	من 1 الى 4
نظام التشغيل في العقدة	Centos 7
ذواكر العقدة	16 غيغا بايت للعقدة الرئيسيّة و14 غيغا بايت لكل عقدة عاملة
عدد نوى المعالج في كل عقدة	8
هادووب	نسخة 3.1.1 (توجد عقدة واحدة لأسماء النطاق NameNode على العقدة المركزيّة أمّا بقية العقد فهي عقد بيانات DataNode)
سبارك	نسخة 3.0.0 ويعمل بالنمط المستقلّ
لغة بايثون	نسخة 3.7.7

تمّ التنفيذ العمليّ لكل من تطبيقي التّدريب والتّنبّؤ باستخدام اطاري العمل TensorFlowOnSpark و Horovod حيث أجريت التّجارب لكل إطار عمل على حدا على Spark و Hadoop وقُيّم أداء كليهما من ناحية زمن التّنفيذ كمعامل أساسي لقياس الأداء.

4-8- مجموعة البيانات

تحوي مجموعة البيانات المُستخدمة على 14 صورة دخل يقابلها 14 صورة من خرائط التّقطيع حيث يوضّح الشكل (4) بعضاً منها. تختلف الصّور الموجودة فيما بينها من ناحية الحجم ويبلغ أكبرها 17 ميغابايت وهي ذات أبعاد متنوّعة. تتالّف كل صورة من المجموعة من 4 حزم طيفيّة هي الأحمر Red والأخضر Green والأزرق Blue اضافة الى طيف الأشعة تحت الحمراء Near Infrared Radio ليصبح التشكيل يضم القنوات الاربعة معاً (**R, G, B, NIR**) علماً ان ال (NIR) لا يمكن رؤيته بالعين البشريّة المجرّدة لكنه يحوي معلومات مهمة يمكن الاستفادة منها، أمّا خرائط التّقطيع فتحوي على 9 قنوات يمثّل كلٌّ منها صنفاً من الأصناف ال9 التي تقابل أنواع الغطاء الأرضي المُستهدفة في تطبيقنا وهذه الأصناف هي الطّرق، الأبنية، الأشجار، العشب، التربة العارية، المياه، السّكك الحديدية، المسابح اضافةً الى الّلاصنف



الشكل (4) مجموعة بيانات التّدريب

4-9- عملية توليد الصّور

نظراً لانّ عدد الصّور المتاحة هي قليلة (14 صورة تدريب) تم اللّجوء الى طريقة تقطيع الصّورة الواحدة الى عدّة مُقتطعات (blocks) متداخلة فيما بينها بابعاد 4*128*128 مع قياس خطوة يبلغ 48 بكسل للانتقال من مُقتطع الى المُقتطع الذي يليه، وهذه الطريقة تمكننا من توليد عدد كبير من الصور الجزئية(مُقتطع) المُتداخلة التى يمكن استخدامها في عملية التّدريب حيث ينتج في حالتنا اكثر من 6700 مقتطع للتدريب في حال استخدام خطوة بحجم 48 بكسل واكثر من 15000 مقتطع في حال استخدام خطوة بحجم 32 بكسل، وهذه العدد الكبير هو من دون استخدام تقنيات تضاعف البيانات(Data Augmentation) الشائع استخدامها في مجال تعلّم الآلة عند الرّغبة بزيادة كمية بيانات التّدريب. بهذه الطريقة اصبح لدينا عدد كاف من بيانات التّدريب الجاهزة لملائمة المودل.

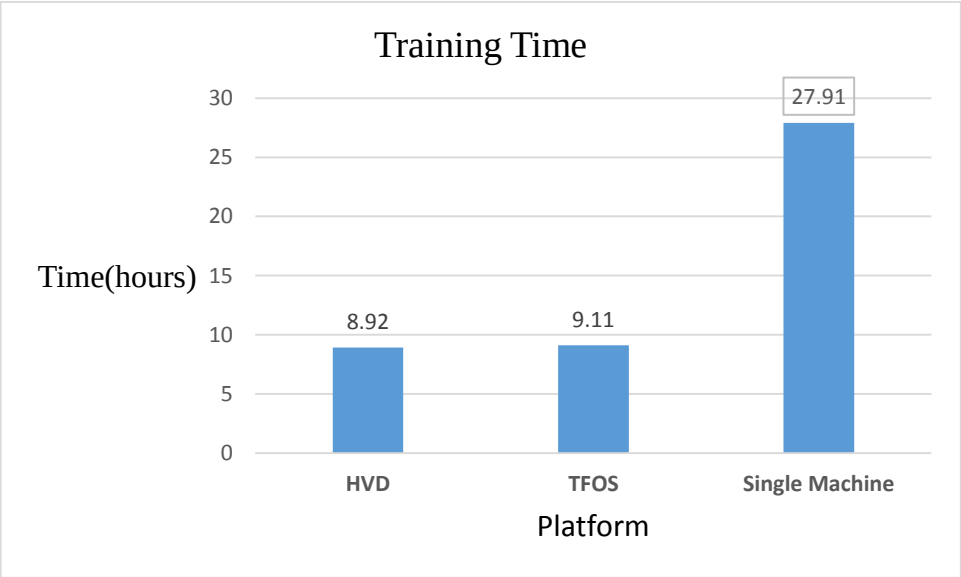
4-10- عمليّة التنبّؤ

في هذه العمليّة يجب أن يكون النّموذج موجودا في نظام الملفّات المحلّي لكل عقدة على حدى أو يكون موجوداً في نظام الملفّات المُوزّع ضمن العنقود او العناقيد ان وجدت وتبدأ العمليّة بتحميل النّموذج الى الذّاكرة الرّئيسيّة ويتم تقسيم البيانات على العقد بحيث تقوم كل واحدة منها بعملية التنبّؤ والحصول على الخرج وكتابته الى HDFS. نظراً لعدم توفّر صور كافية للاختبار في عمليّة التّنبؤ، تمّ مضاعفة إحدى الصّور من خلال نسخها عدّة مرّات لتكوين عدد كبير منها ومعالجتها في التّطبيق، وهذا مقبول في حالتنا لكوننا معنيين بقياس الزّمن المُستهلك لانجاز العمل المطلوب كمعامل أداء أساسي بغضّ النّظر عن محتوى الصّور التي تتم معالجتها.
الجدير بالذّكر هنا أنّه لا حاجة لتعديل حجم صورة الدّخل الى 4*128*128 كما كانت صور الدّخل في عمليّة التّدريب، بل يقبل النّموذج صوراً من أبعاد متعدّدة بشكل مُتكيّف مع إبقاء عدد القنوات مساوياً ل4.

5- النتائج والمناقشة

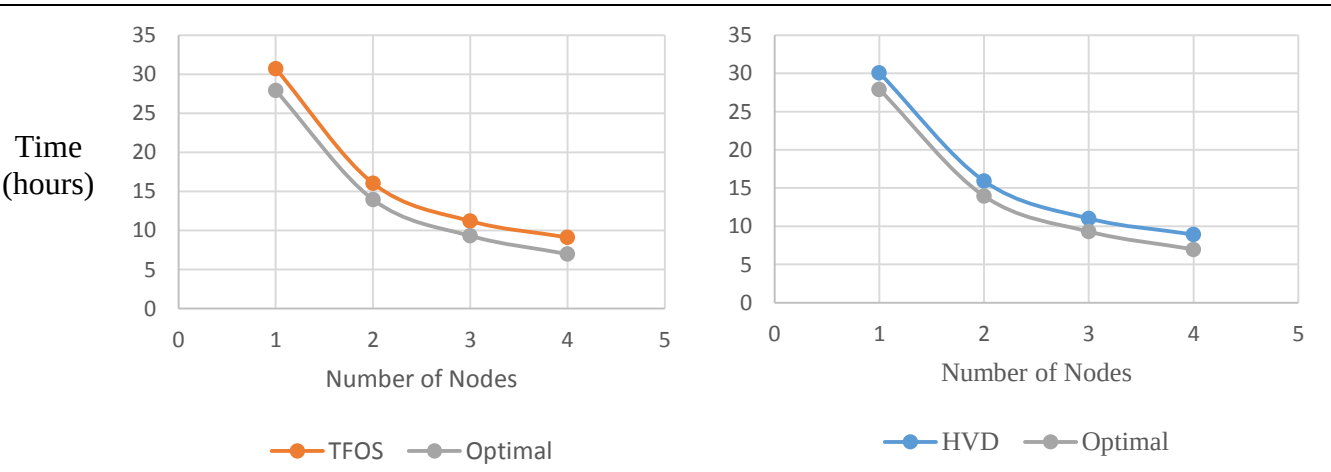
تمّ تنفيذ كل من عمليتي التّدريب والتنبّؤ على العنقود وفق مراحل لمعرفة وتقييم كفائة استخدام إطار العمل Spark في انجاز العمل بالإضافة الى قابلية التّوسّع للبيئة التي تم تهيئتها والعمل عليها حيث تمّ الاختبار على عنقود بحجم 1 و2 و3 و 4 عقد على التوالي وقياس الزمن المستهلك الكلّي الذي استغرقه العنقود لانجاز العمل المطلوب.

بدايةً تمّ تدريب النّموذج على عنقود بحجم 4 باستخدام كل من TensorFlowOnSpark و Horovod وتدريبه أيضاً وفق منهجيّة الجهاز المنفرد وقورنت قيم الأداء لكل منهم. أعطى تدريب النّموذج على منصّات البيانات الكبيرة و النّظم المّوزّعة نتائج جيّدة جدّا لناحية التّحسين الكبير في الزمّن المُستهلك حيث أنهى العنقود تدريب النّموذج خلال 8 ساعات و55 دقيقة باستخدام Horovod فيما استغرق 9 ساعات و7 دقائق باستخدام TensorFlowOnSpark، أمّا منهجيّة الجهاز المنفرد فقد استهلكت 27 ساعة و 55 دقيقة لاستكمال التّدريب كما في الشكل (5).



الشكل (5) زمن تنفيذ التّدريب باستخدام منهجيّة الجهاز المنفرد مقابل منهجيّة النّظم المُوزّعة على منصّات البيانات الكبيرة

يمكن القول أنّ هذه النّتيجة متوقّعة بالفعل لأنّ استثمار قدرات عدّة أجهزة (عقد) تعمل معاً لانجاز مهمّة واحدة مُشتركة ضمن نظام مُوزّع سيفضي الى أداءٍ أفضل بكثير عند المقارنة مع العمل على جهاز منفرد ذو مواصفات مطابقة لعقدة واحدة، كما أنّ الأداء سيتحسّن في كلّ مرّة يزداد حجم العنقود وفق نفس الشّروط الأوليّة للنظام. ولمعرفة كفاءة النّظام المُقترح في إنجاز هكذا أنواع من العمل خصوصاً عندما تكون حجوم البيانات كبيرة جدّا فإنّه ينبغي قياس أداء النّظام بشكل مستمرّ مع زيادة عدد العقد في العنقود لمعرفة اذا ما كان هناك أداء خطّي للنظّام أو ما يقاربه بزيادة عدد العقد من خلال متابعة الفرق بشكل مستمر. وتجري هذه المقارنة عادة مع الحالة المثاليّة للأداء التّي تُؤخذ قيمها اعتماداً على زمن تنفيذ العقدة الواحدة T ومن ثمّ تُحسب أزمنة التنفيذ على عقدتين بتقسيم T على 2 وتُحسب على 3 عقد بتقسيم T على 3 وهكذا دواليك حتى الانتهاء من التّجربة باستخدام حجم العنقود الأعظميّ. هذا وقد تمّ تسجيل النّتائج التّي تمّ الحصول عليها في كلّ مرّة وتمثيلها كما يبيّن الشكل(6).

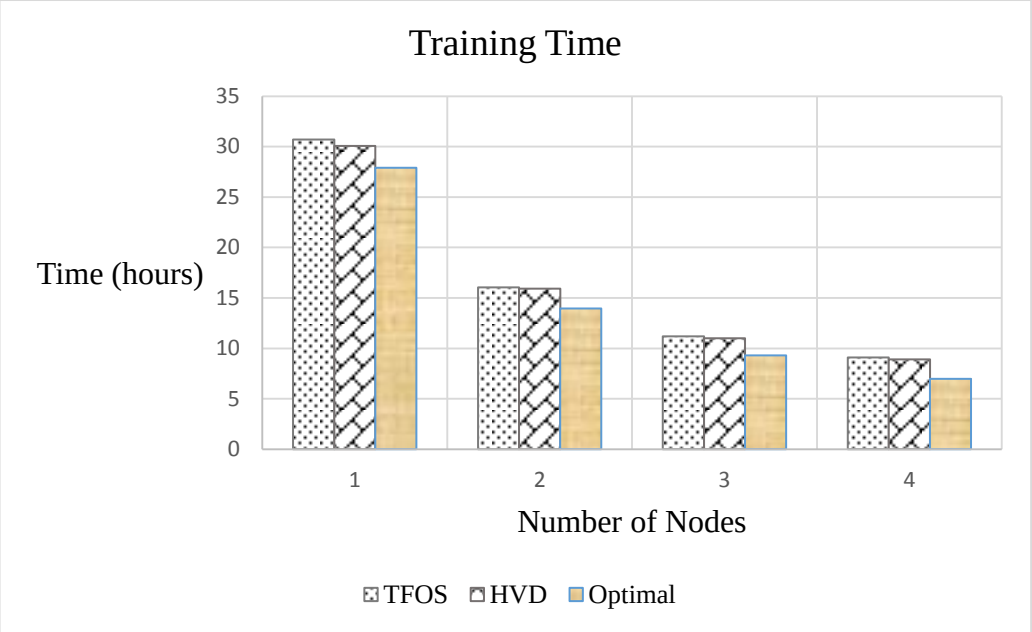


الشكل(6) زمن تدريب نمّوذج ال U-Net على العنقود بالمقارنة مع الحالة المثاليّة بزيادة عدد العقد

يوضّح الشكل(6) مقارنة بين أداء عمل TensorFlowOnSpark و Horovod مع الحالة المثالية من حيث زمن التّنفيذ عند تدريب النّموذج مع زيادة عدد العقد المستخدمة ضمن العنقود بحيث تمّت إعادة التجربة من أجل كل عدد جديد من العقد وحساب الزمّن في كلّ مرّة ليتبيّن لنا أن هناك فرق بأقل من 3 ساعات عند وجود عقدة واحدة فقط ليتوسع هذا الفارق قليلاً عند وجود عقدتين فيما يستقر الوضع نسبيا عند زيادة العقد. إنّ سبب زيادة الزمن يعود الى العبئ الإضافي الناتج من عميات الاتّصال بين العمليّات وتبادل البيانات فيما بينها بعد الانتهاء من حساب المشتقّات في كل دفعة بيانات (batch) لكي تحصل جميع العمليّات على نفس القيم اللّازمة لتحديث النّموذج.

نستنتج هنا أنّ هناك عبئاً اضافياً من تنفيذ التّدريب على Spark وخصوصا عندما يكون العمل يجري على عدد قليل من العقد العاملة في العنقود الّا أن التحسين في زمن تنفيذ يبدو جليّا عندما يزداد حجم العنقود وهو يعطي أداء جيّدا ومقبولاً لأنّ هامش الفرق بين الحالة المثاليّة بالمقارنة مع إطاري العمل تبقى ثابتة تقريباً حيث نلحظ ذلك من كون أنّ الخطين البيانيّين لكل من TFOS و HVD لا يتباعدان عن الخطّ البياني للحالة المثاليّة مع زيادة عدد العقد.
يعود الفرق بين الخطّ البياني للحالة المثاليّة والخطّين البيانيين الخاصين باطاري العمل الى العبئ الاضافي النّاتج من تهيئة واعداد كل من Horovod و TensorFlowOnSpark إضافة للوقت اللازم في إعداد العنقود بشكل عام، يُضاف اليه الزّمن المُستهلك في الاتّصالات وتبادل البيانات بين العقد العاملة.

يمكن مشاهدة الفرق بين إطاري العمل من خلال خلال إعادة تمثيل الشّكل السّابق باستخدام مُخطّط آخر كما يبيّن الشكل(7) ، حيث نلحظ التّقارب الكبير بين أدائي إطاري العمل مع فارق بسيط لصالح Horovod .

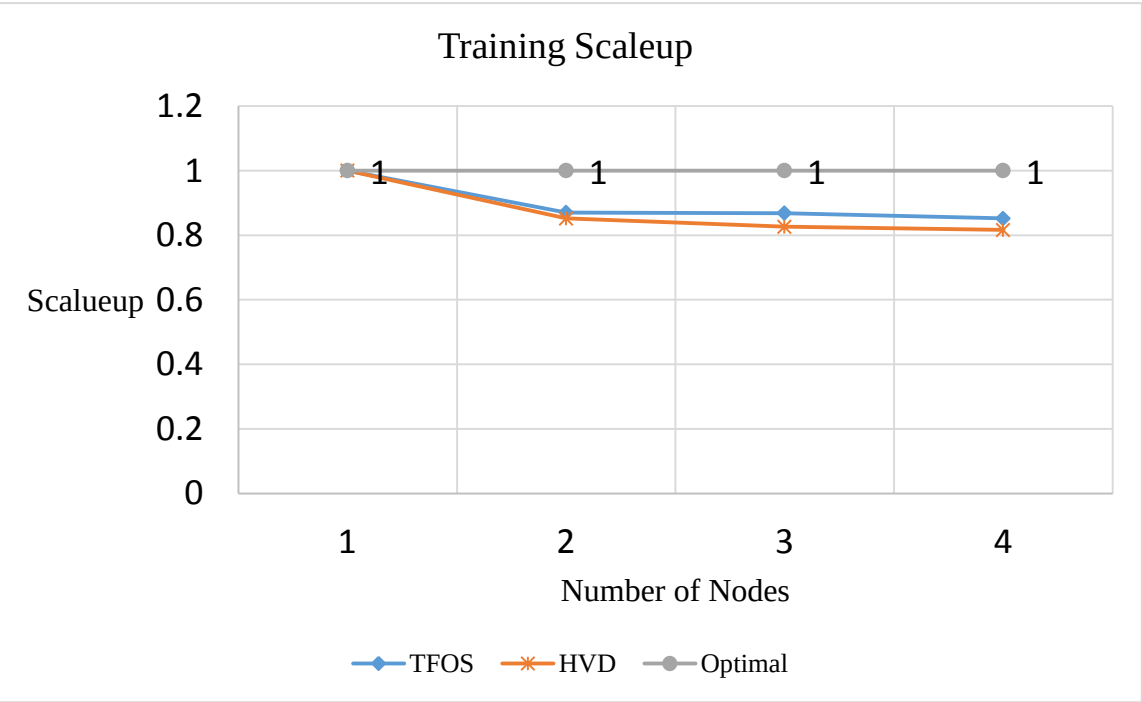


الشكل(7) مقارنة بين إطاري العمل TensorFlowOnSpark و Horovod من ناحية زمن تدريب النّموذج على العنقود مع الحالة المثاليّة

أحد المؤشّرات المهمّة التّي ينبغي النّظر اليها بما يتعلق بأداء هكذا اختبارات هو معامل التّوسع (Scaleup) ، ويُقصد به قياس الأداء حين يتمّ زيادة الموارد المتاحة للتنفيذ مع زيادة حجوم البيانات المّراد معالجتها بنفس النسبة، وتمّ ذلك في دراستنا من خلال معالجة 25% من البيانات بعقدة واحدة، ثمّ معالجة 50% بعقدتين، ثمّ 75% بثلاث عقد لننتهي بمعالجة كل البيانات بأربع عقد، ونحصل على قيم التّوسّع من خلال تقسيم زمن التّنفيذ بعد زيادة البيانت وحجم العنقود على زمن التّنفيذ قبل الزّيادة.

يبيّن الشكل(8) مقارنة بين الحالة المثالية للتّوسّع والتي يبقى فيها الزمن المُستهلك ثابتاً تماما بتغير الموارد المُتاحة وحجوم البيانات مع حالتنا باستخدام أُطر العمل وفق نفس القيم السّابقة للتجربة، حيث نلحظ انخفاض في معامل التّوسّع لكل من إطاري العمل عند استخدام عقدتين مع استمرار الانخفاض بنسبة بسيطة عند استخدام 3 عقد، فيما يستقرّ

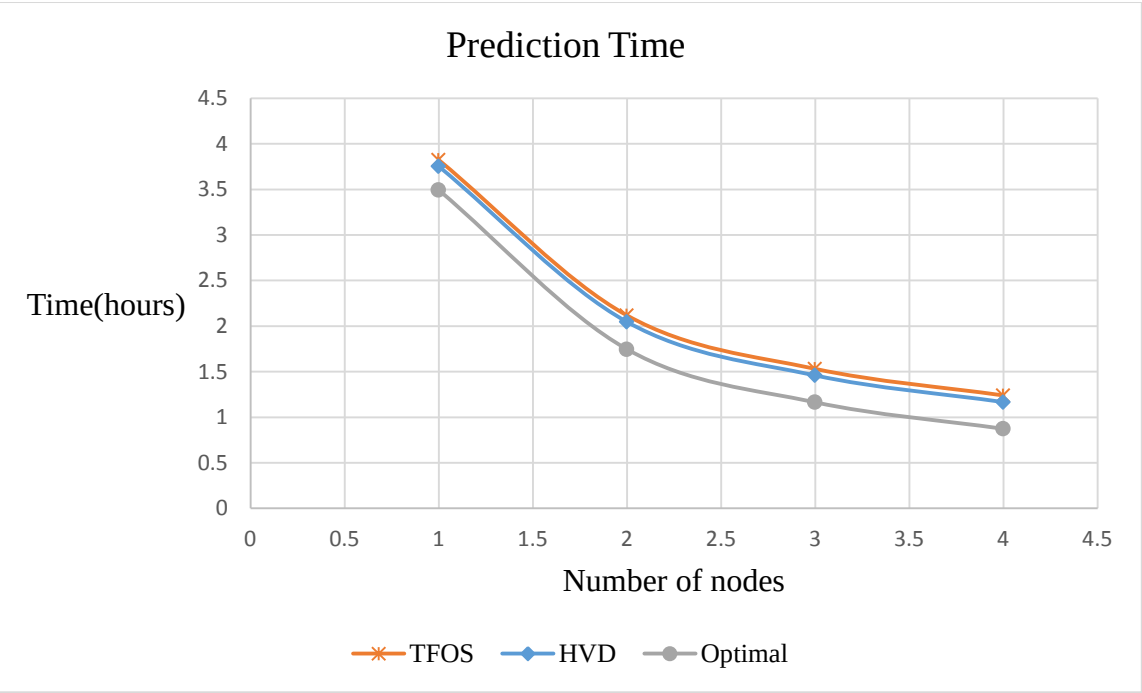
الوضع على معامل توسّع أكبر من 0.81 وهو نسبة مقبولة. أيضا نجد من الشّكل معامل توسّع أفضل ل TFOS بالمقارنة مع HVD ، ويعود السّبب الرّئيسي لذلك لكون زمن التّنفيذ على عقدة واحد باستخدام TFOS هو أقل من زمن التنفيذ على عقدة واحدة باستخدام HVD بمقدار كبير نسبيّاً لأنّ كلفة عمليّات الإعداد وتهيئة عنقود في HVD باستخدام عقدة واحدة أقل من نظيرتها في TFOS، وعند التّنفيذ على أكثر من عقدة يتقارب أداء إطاري العمل، ولكون مقياس التّوسّع يعتمد بالحساب على زمن تنفيذ عقدة واحدة في كل منهما فإنّه من المُتوقّع الحصول على نتائج كالتي حصلنا عليها كما يوضّح الشّكل المذكور آنفاً.



الشكل(8) قابلية توسّع إطاري العمل في تطبيق التّدريب على العنقود مع الحالة المثاليّة بزيادة عدد العقد

أُجريت عمليّة التّنبّؤ على 384 صورة كبيرة الحجم مع زيادة عدد العقد المستخدمة ضمن العنقود بحيث تمّت إعادة التجربة من أجل كل عدد جديد من العقد وحساب الزّمن المُستهلك.

يوضّح الشكل(9) مقارنة بين أداء عمل TensorFlowOnSpark و Horovod مع الحالة المثالية من حيث زمن التّنفيذ عند القيام بعملية التنبّؤ مع زيادة عدد العقد المستخدمة ضمن العنقود بحيث تمّت إعادة التجربة من أجل كل عدد جديد من العقد وحساب الزمّن المُستهلك. وقد استغرق النّظام المُقترح للتنبّؤ ب384 صورة 1 ساعة و10 دقائق باستخدام Horovod و 1 ساعة و14 دقيقة باستخدام TensorFlowOnSpark فيما كان الزّمن المستهلك لتنفيذ نفس التّطبيق وفق منهجيّة الجهاز المنفرد 3 ساعات و36 دقيقة مما يبيّن التّحسين الكبير في أداء النظام.
بملاحظة الشكل(9) نستطيع رؤية أداء مشابه لعمليّة التّدريب لناحية قابليّة التوسّع حيث نرى انخفاضاً في الأداء عند التّنفيذ على عقدة واحدة، ليتوسّع الفارق مع زيادة عدد العقد في العنقود ويستقرّ الوضع عند تكرار التّجربة على عقدتين أو أكثر كما تبيّن الخطوط البيانيّة لكل من الحالة المثاليّة وإطاري العمل المُستخدَمين.



الشكل(9) زمن تنفيذ تطبيق التّنبؤ

6- الخاتمة

تكمن أهمّيّة هذا النّوع من الأبحاث في تخفيض الوقت اللّازم لمعالجة البيانات في تطبيقات الاستشعار عن بعد وخصوصاً تلك التّي تعتمد على تقنيات الذّكاء الصّنعيّ وتخفيض التّكاليف المتعلّقة بتوفير عتاد مُخصّص وباهظ الثمن لمعالجة هكذا أنواع من التطبيقات، وتلك التكاليف كان من الممكن أن تبقى كبيرة بالمقارنة مع النّظم التقليديّة من دون استخدام تقنيات البيانات الكبيرة ونظم الحوسبة فائقة الأداء كالتّي تم استثمارها في هذا البحث لإنجاز العمل المطلوب.

بيّنا في هذا البحث كيف يمكن استثمار تقنيات البيانات الكبيرة ونظم الحوسبة فائقة الأداء في معالجة صور الأقمار الصّنعيّة في تطبيقات التّعلّم العميق المعروفة بحاجتها لموارد ذات قدرات حوسبة عالية أثناء التّنفيذ، هذا وقد واقترح البحث استخدام سبارك كمحرّك أساسي للتنّفيذ ومعالجة البيانات فيما فنّد استخدام

إطـــــاري عمـــــل لأجـــــل المعالجـــــة التّفرعيّـــــة همـــــا Horovod و TenosrFlowOnSpark، وتـــمّ ايضـــاً اســـتُخدام نظـــام الملفّـــات المُـــوزّع الخـــاص بهادووب لتخزين البيانات.

أظهـرت النّتـائج أداء جيّـداً جـدّا للمنهجيّـة المُقترحـة والـنظم الموزعـة علـى منصـات البيانـات الكبيـرة بالمقارنـة لناحيـة زمـن التنفيـذ مـع منهجيّـة الجهـاز المنفـرد فـي كـل مـن تطبيقـيّ التّـدريب والتّنبـؤ باسـتخدام إطـاري العمـل TensorFlowOnSpark و Horovod علـى سـبارك، والأهـمّ مـن ذلـك هـو أنّ الـزمن المُسـتهلك فـي تنفيـذ التطبيقـات باسـتخدام المنهجيّـة المتُبعـة لكـل مـن إطـاري العمـل يقـارب نظيـره فـي الحالـة المثاليـة حيـث أنّ الفـارق بينهمـا كـان مسـتقرا تقريبـا مـع زيـادة حجـم العنقـود. وعنـد مقارنـة إطـاري العمـل تبـيّن لنـا تقـارب الأداء بينهمـا لناحيـة زمـن التّنفيـذ مـع ملاحظـة تفـوّق Horovod فـي التّطبيقـين المـذكورين بفـارق قليـل نسـبيّاً علـى TensorFlowOnSpark بنسـبة تقـارب ال (2.1%) فيمـا كانـت قابليّـة توسّـع TensorFlowOnSpark أفضل بنسبة تقارب ال (5%).

7- التوصيات

إن من الجوانب المهمة والمساهمات الأساسية لدراستنا هو فتح الآفاق لدراسات مستقبلية يمكن العمل عليها من أجل تحسين النتائج وفتح باب اعتماد النموذج المقدم في الدراسة وتطويره ليلائم بيئات العمل المختلفة، وتطبيقه في قطاعات عمل أخرى. ومن هنا نعرض بعض الحالات التي نقترح العمل عليها مستقبلاً:

- اختبار نماذج تعلّم عميق كبيرة لا يمكن استيعابها على جهاز واحد باستخدام منهجيّة تفرّعيّة النّموذج وتقييم أدائها.
- استخدام المنهجيّة المُقترحة في هذا البحث في معالجة تطبيقات ذات طبيعة مختلفة كاكتشاف التغيّرات أو تطبيقات تتطلّب استجابة في الزّمن الحقيقيّ
- لتقنيات الاستشعار عن بعد في مجال تطبيقات معالجة الفيديو الفضائيّ مستقبل واعد للمستثمرين في عدّة مجالات حيث باتت عدّة شركات تقدّم هذه الخدمات،

الّا أنّ هذه التّطبيقات ذات متطلّبات وموارد أعلى بكثير من تلك التّي تعالج الصّور فحسب ومن المهمّ أن يكون المنهج المُقترح قابلاً للتعديل أو التّطوير وقادراً على التّكيّف لتلبية هذه النّوع من مجالات العمل لذلك نوصي باستخدام المنهجيّة المُقترحة في هذه الدّراسة كنواة عمل وتقييم أدائه.

- وضع نمّوذج جاهز للاستثمار كمنتج نهائيّ يمكن للمستثمرين استخدامه في أعمالهم بشكل آلي ويدعم تقديم خدمات المعالجة الأوليّة للبيانات وازالة أو تخفيف أعباء إعداد بيئة العمل المناسبة لتنفيذ التّطبيقات المطلوبة.

المراجع

[1] P. Swarnalatha and P. Sevugan, *Big Data Analytics for Satellite Image Processing and Remote Sensing*. IGI Global, 2018.

[2] H. M. Patel, K. Panchal, P. Chauhan, and M. B. Potdar, "Large scale image processing using distributed and parallel architecture," *Gujrat, India, Int. J. Comput. Sci. Inf. Technol.*, vol. 6, no. 6, pp. 5531–5535, 2015.

[3] R. Yadav and D. M. C. Padma, "Processing of Large Satellite Images using Hadoop Distributed Technology and Mapreduce: A Case of Edge Detection," *Int. J. Recent Innov. Trends Comput. Commun.*, vol. 3, no. 5, pp. 3456–3460, 2015.

[4] T. Sharma, V. Shokeen, and S. Mathur, "Distributed Approach to Process Satellite Image Edge Detection on Hadoop Using Artificial Bee Colony," *Int. J. Serv. Sci. Manag. Eng. Technol.*, vol. 11, no. 2, pp. 80–94, 2020.

[5] M. M. U. Rathore, A. Ahmad, A. Paul, and J. Wu, "Real-time continuous feature extraction in large size satellite images," *J. Syst. Archit.*, vol. 64, pp. 122–132, 2016.

[6] B. Shangguan and P. Yue, "SPARK processing of computing-intensive classification of remote sensing images: the case on K-means clustering algorithm," in *2018 26th International Conference on Geoinformatics*, 2018, pp. 1–4.

[7] I. Nurwauziyah, U. Sulistyah, I. Gede, I. G. B. Putra, and M. Firdaus, "Satellite Image Classification using Decision Tree, SVM and k-Nearest Neighbor," *no. July*, 2018.

[8] S. A. Manaf, N. Mustapha, M. Sulaiman, N. A. Husin, and M. R. A. Hamid, "Artificial neural networks for satellite image classification of shoreline extraction for land and water classes of the north west coast of peninsular Malaysia," *Adv. Sci. Lett.*, vol. 24, no. 2, pp. 1382–1387, 2018.

[9] Z. N. Absardi and R. Javidan, "Classification of big satellite images using hadoop clusters for land cover recognition," in *2017*

IEEE 4th International Conference on Knowledge-Based Engineering and Innovation (KBEI), 2017, pp. 600–603.

[10] J. Dowling, "Distributed TensorFlow," 2017. https://www.oreilly.com/people/jim-dowling/.

[11] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*, 2015, pp. 234–241.

[12] A. Foundation, "Apache Spark: Lightning-fast unified analytics engine," 2020. https://spark.apache.org/.

[13] B. Chambers and M. Zaharia, *Spark: The definitive guide: Big data processing made simple*. " O'Reilly Media, Inc.," 2018.

[14] L. Yang, J. Shi, B. Chern, A. Feng, and Y. B. M. Team, "Open Sourcing TensorFlowOnSpark: Distributed Deep Learning on Big-Data Clusters," 2017. https://www.oreilly.com/content/distributed-tensorflow/.

[15] A. Sergeev and M. Del Balso, "Horovod: fast and easy distributed deep learning in TensorFlow," *arXiv Prepr. arXiv1802.05799*, 2018.

[16] J. J. Dai *et al.*, "Bigdl: A distributed deep learning framework for big data," in *Proceedings of the ACM Symposium on Cloud Computing*, 2019, pp. 50–60.

[17] J. Patterson and A. Gibson, *Deep learning: A practitioner's approach*. " O'Reilly Media, Inc.," 2017.

[18] Google, "Introduction to gRPC," 2020. https://grpc.io/docs/what-is-grpc/introduction/.