مجلة جامعة طرطوس للبحوث والدراسات العلمية _ سلسلة العلوم الهندسية المجلد (5) العدد(9) 2021

Tartous University Journal for Research and Scientific Studies - engineering Sciences Series Vol. (5) No. (9) 2021

تخفيض التعقيد الحسابي في الشبكات العصبية الالتفافية التي تعتمد في بنيتها على اتصالات التخطي

أ.د.يعرب ديوب*

د. لينا مقديسيان**

م. كلودا ابراهيم***

(تاريخ الإيداع 12 /5 /2021 . قُبِل للنشر في 25 /10 /2021)

□ ملخّص □

تم في هذا البحث دراسة التعقيد الحسابي للشبكات العصبية الالتفافية Convolutional neural networks (CNN)التي تعتمد في بنيتها على اتصالات التخطي والعمل على تقليل هذا التعقيد من خلال إجراء تغيير بسيط في تصميم هذه الشبكات لتقليل عدد الاتصالات وبالتالي ضغط بنى هذه الشبكات. تم تقييم أداء الخوارزمية من خلال إجراء سلسلة من التجارب على مجموعات البيانات المرجعية المستخدمة على نطاق واسع من أجل مهام تصنيف الصور ومقارنة النتائج مع الخوارزميات الحديثة التي أظهرت أداءً واعداً في هذا المجال. حققت هذه الخوارزمية أفضل دقة بين جميع الخوارزميات الحديثة التي تمت المقارنة معها وذلك على مجموعة البيانات Cifar10 ومجموعة البيانات Cifar100.

لكلمات المفتاحية:الشبكات العصبونية الالتفافية، CNN، الحوسبة التطورية، التعلم العميق، تصنيف الصور، التعرف على النماذج.

* أستاذ في جامعة طرطوس - كلية هندسة تكنولوجيا المعلومات والاتصالات - قسم هندسة تكنولوجيا المعلومات.

** أستاذ مساعد في جامعة طرطوس - كلية هندسة تكنولوجيا المعلومات والاتصالات - قسم العلوم الأساسية.

*** حاصلة على درجة الماجستير في جامعة طرطوس - كلية هندسة تكنولوجيا المعلومات والاتصالات -قسم هندسة تكنولوجيا المعلومات.

مجلة جامعة طرطوس للبحوث والدراسات العلمية _ سلسلة العلوم الهندسية المجلد (5) العدد(9) 2021

Tartous University Journal for Research and Scientific Studies - engineering Sciences Series Vol. (5) No. (9) 2021

Reducing computational complexity in convolutional neural networks whose architecture dependent on skip connections

Prof. Yaroyb Dayoub*
Dr. Lina Makdessian**
Eng. Kloda Ibrahim***

(Received 12 /5 / 2021 . Accepted 25 / 10 / 2021)

□ ABSTRACT □

In this paper, we study the computational complexity of Convolutional neural networks (CNNs) that depend in their structure on skip connections and work to reduce this complexity by making a simple change in the design of these networks to reduce the number of connections and thus compress the structures of these networks. The performance of the algorithm was evaluated by running a series of experiments with widely used reference datasets for image classification tasks and comparing results with modern algorithms that have shown promising performance in this area. This algorithm achieved the best accuracy among all the modern algorithms compared with it on the Cifar10 dataset and the Cifar100 dataset.

Keywords: Convolutional Neural Networks, CNN, Evolutionary Computing, Deep Learning, Image Classification, Pattern Recognition.

* Professor in Information and Communication Technology Engineering Faculty,Tartous University, Syria.
** Assistant Professor in Information and Communication Technology Engineering Faculty, Tartous University, Syria.
*** Master in Information and Communication Technology Engineering Faculty, Tartous University, Syria.

مقدمة

تعد الرؤية الحاسوبيةمن مجالات علوم الحوسبة الحديثة وشكل من أشكال الذكاء الاصطناعي، تساعد هذه التقنية على رؤية العالم وتحليل البيانات المرئية لاستخدامها في اتخاذ القرارات أو اكتساب فهم حول البيئة والعالم بالإضافة إلى تحديد ومعالجة الأشياء مثل الصور ومقاطع الفيديو بنفس الطريقة التي يَفعلها البشر .

أظهرت الشبكات العصبية الالتفافية (CNN) في السنوات القليلة الماضية نجاحًا كبيرًا في مجال الرؤية الحاسوبية وتحديداً في مهام التعرف على النماذج، حيث وفّرت درجة عالية جدًا من الدقة مقارنة بطرق التعلم الآلي الأخرى وذلك نظراً لقدرتها على استخراج السمات تلقائياً دون تدخل بشري واستخدامها معالجة أولية قليلة نسبياً مقارنة بخوارزميات تصنيف الصور الأخرى[2,1].من ناحية أخرى، في الوقت الحاضر يتم استخدام الحوسبة التطورية (الخوارزميات الجينية، خوارزميات الأسراب وغيرها) بشكل متزايد لحل مشاكل الأمثلة ومنها تحديد البارامترات المثلى للدالة [3, 4, 5]، وبما أن تصميم بنية CNN يرتبط ارتباطاً وثيقاً باختيار العديد من البارامترات لذلك بدأ الاهتمام في تطبيق استراتيجيات الحوسبة التطورية لتحديد البارامترات المثلى لتشكيل بنية CNN.

في السنوات الأخيرة، اكتسبت الشبكات العصبية التلافيفية أهمية كبيرة حيث تعد ResNet [6] وDenseNet [7] من أحدث شبكات CNN المصممة يدوياً والتي حققت نجاحاً كبيراً في مجال التعرف على النماذج، ومن ثم تم اقتراح خوارزميات متعددة لتصميم بنى CNNأهمها [8] Genetic CNN،-Large Scale Evolution [9]، [10] CGP-CNN.

تواجه الشبكات العصبية الالتفافية (CNN) الخاصة بمهام الرؤية الحاسوبية عدة تحديات تعيق نشرها في تطبيقات العالم الحقيقي وهي[11 ,12]:

1. حجم النموذج: تأتي قوة التمثيل القوية لشبكات CNN من الملايين من البارامترات القابلة للتدريب حيث يجب تخزين هذه البارامترات بالإضافة إلى معلومات بنية الشبكة على القرص وتحميلها في الذاكرة أثناء وقت الاستدلال.
2. ذاكرة وقت التشغيل: أثناء وقت الاستدلال قد تستغرق عمليات التنشيط/الاستجابات الوسيطة لشبكات CNN مساحة ذاكرة أكبر من تخزين بارامترات النموذج حتى مع حجم الدفعة 1، وهذه ليست مشكلة بالنسبة لوحدات معالجة الرسومات المتطورة، ولكن هناك العديد من التطبيقات ذات القدرة الحسابية المنخفضة التي لا تستطيع تحملها.
3. عدد عمليات الحوسبة: عمليات الالتفاف مكثفة حسابياً على الصور عالية الدقة وبالتالي قد تستغرق شبكة CNN الكبيرة عدة دقائق لمعالجة صورة واحدة على جهاز محمول، مما يجعل اعتمادها للتطبيقات الحقيقية غير واقعي.

لذلك سيتم في هذا البحث دراسة التعقيد الحسابي للشبكات التي تم توليدها من خلال الخوارزمية التي تمت دراستها واختبارها[13]، والعمل على تقليل هذا التعقيد من خلال إجراء تغيير بسيط في تصميم هذه الشبكات عن طريق الانتقال من طوبولوجيا التجميع الكامل في وحدة الشبكة كثيفة الاتصال UnitDenseNet إلى طوبولوجيا يكون فيها دخل كل كتلة ضمن هذه الوحدة عبارة عن مجموعة فرعية من خرج الكتل السابقة ضمن نفس الوحدة أي العمل على ضغط بنى الشبكات التي يتم توليدها.

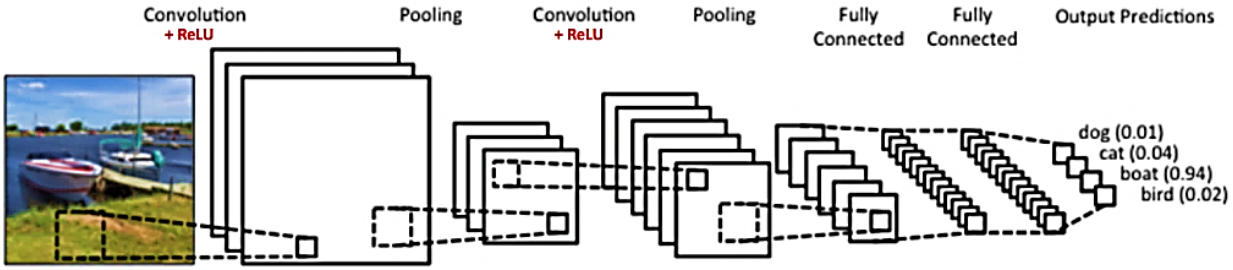
أهمية البحث وأهدافه

يستمد هذا البحث أهميته من ظهور الشبكات العصبية الالتفافية كنماذج رائدة لمهام التعرف على النماذج بالإضافة إلى فعاليةالخوارزميات الجينية في حل المشاكل المعقدة ذات فضاء البحث الكبير.يهدف هذا البحث إلى تحسين أداء الشبكات العصبونية الالتفافية المستخدمة في مهام التعرف على النماذج عن طريق دراسة التعقيد الحسابي للشبكات والعمل على تقليل هذا التعقيد.

طرائق البحث ومواده

1. الشبكات العصبية الالتفافية

الشبكة العصبية الالتفافية العميقة DCNN هي نوع خاص من الشبكات العصبية التي أظهرت أداءً مثالياً في العديد من المسابقات المتعلقة بالرؤية الحاسوبية ومعالجة الصور.يلعب ترتيب مكونات CNN دورًا أساسيًا في تصميم بنى جديدة وبالتالي تحقيق أداء محسَّن[14].



الشكل (1): بنية شبكة CNN

يوضح الشكل (1) مثالاً لبنية شبكة CNN حيث تتكون الشبكة العصبية الالتفافية بشكل عام من عدة طبقات مختلفة لكل منها وظيفتها الخاصة، يتم تصنيف الطبقات الرئيسية لأي شبكة عصبية التفافية إلى أربع طبقات وهي[15]:

1. **الطبقة الالتفافيةConvolutional layer**:هي العمود الفقري لـ CNN وتأتي تسميتها من عملية الطي أو الالتفاف الرياضية، حيث يكون ناتج هذه العملية هو خريطة السمات feature map التي تعكس استجابة المرشحات لنمط معين في الصورة من خلال أوزان كل مرشح ويتم تحديد أوزان المرشح أثناء عملية تدريب الشبكة.

تتكون خريطة السمات من عدة قنوات، ترتبط أبعاد هذه القنوات بأبعاد مصفوفة الإدخال وأبعاد المرشح بالإضافة إلى العاملين التاليين:

- الخطوةStride: تمثل عدد العناصر التي يتم إزاحة المرشح بمقدارها بعد كل عملية.
- الحشو Padding: هي عملية توسيع المصفوفة إما بإضافة أصفار إلى حدود المصفوفة أو تكرار قيم نهايات المصفوفة. وبالتالي، يتم استخدام المصفوفة بأكملها في عملية الترشيح ولا يتم فقد أي معلومات على نهايات المصفوفة.

2. **طبقة التفعيلActivation Layer**:بعد انتهاء عملية الالتفاف يتم إدخال خريطة السمات إلى طبقة التفعيل حيث يُطبّق تابع التفعيل على كل عصبون أي ما يكافئ عنصر من خريطة

السمات، ويُؤخذ بعين الاعتبار الحد من خرج العصبون وكذلك اللاخطية في عملية التفعيل حيث أنّ عمليات الالتفاف التي أجريت قبلها هي عمليات خطية. أهم توابع التفعيل المستخدمة في هذا النوع من الشبكات هو تابع الوحدة الخطية المصححة ReLU الذي أثبت فاعليته مقارنة بالتوابع الأخرى.

3. **طبقة التجميعPooling layer**:بعد تطبيق تابع التفعيل على خريطة السمات، يتم العمل على تقليل أبعادها بطريقة تحافظ على المعلومات من خلال عملية التجميع، ويتم ذلك بعدة طرق أهمها التجميع بالحد الأقصىmax-pooling حيث يتم مقابلة كل نافذة (مجموعة من العناصر المجاورة) مع عنصر واحد يمثل أعلى قيمة داخل هذه النافذة.

4. **الطبقة المتصلةبالكامل Fully Connected layer**:هذه الطبقة هي الأخيرة في الشبكة الالتفافية وهي من نوع multi-layer perceptron وفيها ترتبط العصبونات بالكامل مع كُل عُقد الطبقة السابقة وتتم فيها عملية التصنيف النهائية حيث يكون دخلها شعاع مكوّن من خريطة السمات بعد إجراء عملية التجميع، وخرجها عبارة عن شعاع يعبّر عن الصف الذي تنتمي إليه خريطة السمات.

2. الخوارزمية المقترحة:

تم العمل على تحسين أداء الشبكات العصبونية الالتفافية المستخدمة في مهام التعرف على النماذج عن طريق اقتراح خوارزمية لتصميم بنى CNN باستخدام الخوارزميات الجينية Genetic Alghorithms (GAs) القادرة على تعلم أفضل بنية لشبكة CNN وفقاً للمهمة المعنية بطريقة تلقائية واستناداً إلى الموارد الحاسوبية المحدودة[13]. اعتمدت الخوارزمية على بنى الشبكات الحديثة التي تم تصميمها واختبارها سابقاً من قبل باحثين آخرين وهي الشبكات المتبقية أو ما يدعى بشبكات الرواسب Residual networks (ResNets)[6]والشبكات الكثيفة الاتصال Densely connected networks (DenseNets)[7]، حيث أحدثت هذه الشبكات ثورة في هذا المجال بسبب استخدامها اتصالات التخطي.

- **اتصال التخطي skip connection:**

يعد اتصال التخطي وحدة قياسية في العديد من بنى الشبكات الالتفافية حيث يقوم بتخطي بعض الطبقات في الشبكة العصبية عن طريق جعل خرج طبقة ما كدخل للطبقات التالية (بدلاً من الطبقة التالية مباشرة فقط) وبالتالي يوفر مسارًا بديلاً لمشتقات تابع الخطأ أثناء الانتشار الخلفي للإشارة (مع الانتشار الخلفي). تستخدم اتصالات التخطي بينالطبقات المختلفة غير المتسلسلة بإحدى الطريقتين التاليتين:

1. الإضافة (Add) كما في الشبكات المتبقية ResNets.
2. السلسلة (Concatination) كما في الشبكات الكثيفة الاتصال DenseNets.

- **مراحل عمل الخوارزمية:**

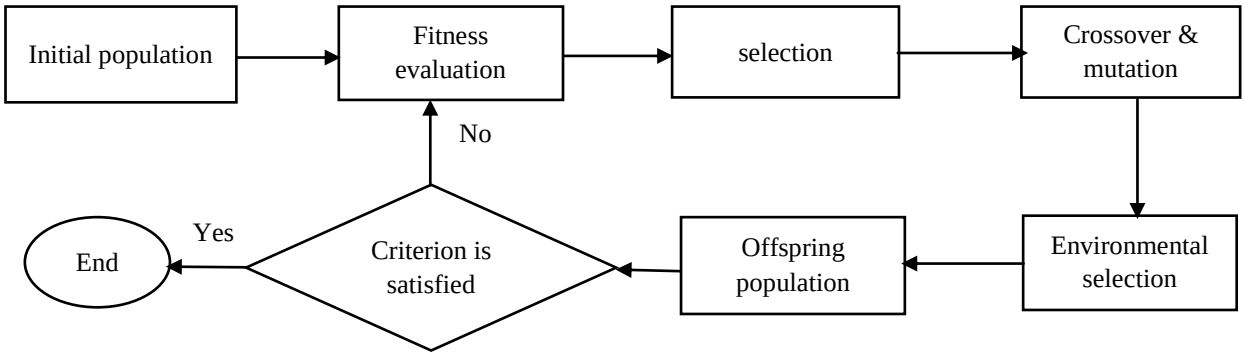
يلخص الشكل (2) مراحل تنفيذ الخوارزمية وهي كما يلي:

1. تمت تهيئة السكان استناداً إلى استراتيجية ترميز الفرد حيث يمثل الفرد بنية شبكة CNN مقترحة لحل مشكلة التصنيف.

2. تستمر عملية التطور حتى يتم تحقق معيار التوقفالمحدد مسبقاً وهو الحد الأقصى لعدد الأجيال في هذا العمل:

- تقييم جميع الأفراد أولاً بناءً على مقياس اللياقة وهو دقة تصنيف الفرد على قاعدة البيانات.
- اختيار الحلول الآباء باستخدام مبدأ المنافسة الثنائية Binary tournament selection.
- إنشاء نسلجديدoffspringباستخدام العوامل الوراثية التقاطعوالطفرة.
- اختيار أفضل الأفراد من مجتمع الآباء ومجتمع النسل الجديد لتشكيل السكان في الجيل التالي للمشاركة في تطور لاحق.

3. بعد انتهاء عملية التطور يتم اختيار أفضل فرد وفك تشفيره إلى شبكة CNN المقابلة.



الشكل (2): مراحل تنفيذ الخوارزمية

اعتمدت استراتيجية ترميز الفرد على ثلاثة أنواع مختلفة من الوحدات ومواقعها في شبكات CNN والمعلومات المشفرة في كل منها وهي:

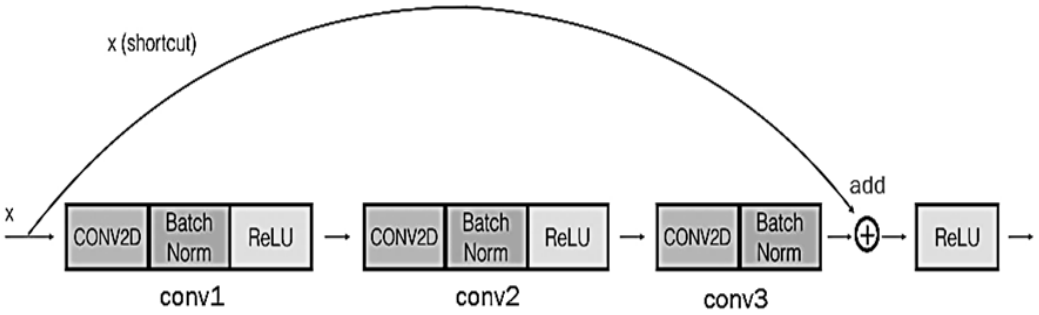
1. وحدة الشبكة المتبقية ResNet Unit (RU):يُظهر الشكل (3) مثالاً لوحدة شبكة متبقية RUتتكون من مجموعة من كتل الشبكات المتبقية RBs (ResNet Blocks) حيث يكون دخل كل كتلة هو خرج الكتلة السابقة لها.

الشكل (3): بنية RU مكونة من 5 كتل RBs

يُظهر الشكل (4) مثالاً لكتلة شبكة متبقية RB مكونة من تتابع ثلاث طبقات التفاف واتصال تخطي واحد، يتم في طبقة الالتفاف الأولى conv1 تقليل الحجم المكاني للمدخلات باستخدام عدد أقل من المرشحات بحجم 1×1 وذلك لتقليل التعقيد الحسابي في طبقة الالتفاف التالية conv2. في طبقة الالتفاف الثانية conv2، يتم استخدام المرشحات ذات الحجم الأكبر مثل 3×3 من أجل تعلم السمات ذات الحجم المكاني

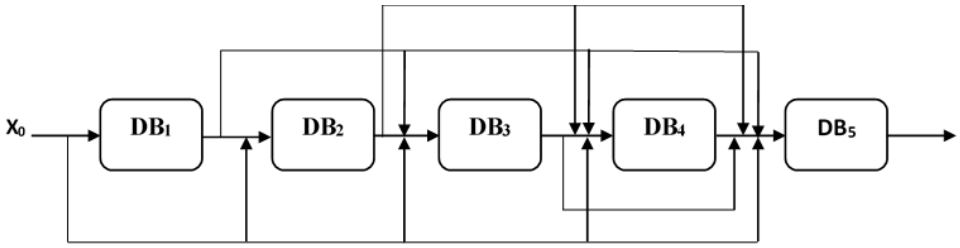
نفسه ومن ثم يتم استخدام مرشحات بحجم 1×1 مرة أخرى في conv3 وزيادة الحجم المكاني لتوليد المزيد من السمات.

يصبح الخرج النهائي للكتلة هو مجموع (add) الدخل x مع ناتج conv3، مع ملاحظة أنه إذا كانت أبعاد الدخل ومخرجات conv3 غير متساوية يتم تطبيق مجموعة من العمليات الالتفافية باستخدام مرشحات بحجم 1×1 على الدخل لتحقيق نفس الحجم المكاني لخرج conv3 وذلك حتى نتمكن من إجراء عملية الجمع (add). بالإضافة إلى ذلك يتم تطبيق تسوية دفعية batch normalization ودالة التفعيل relu بعد كل عملية التفاف.



الشكل (4): بنية ResNet Block

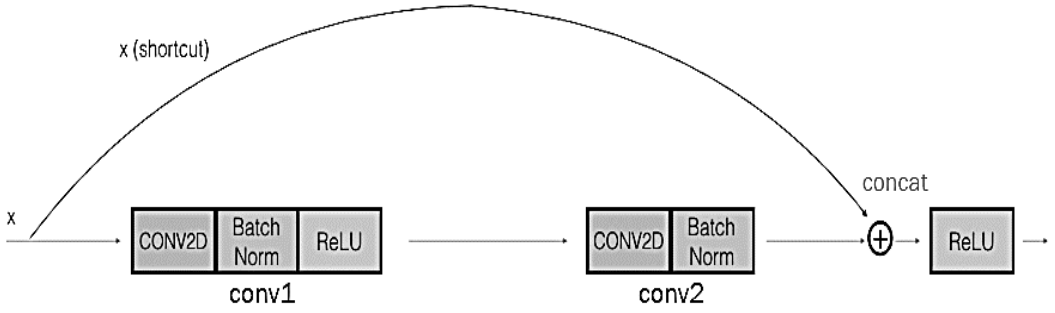
2. وحدة الشبكة الكثيفة الاتصال (DenseNet Unit (DU:يُظهر الشكل (5) مثالاً لوحدة شبكة كثيفة الاتصالDUتتكون من مجموعة من كتل الشبكات الكثيفة الاتصال DBs (DenseNet Blocks) بحيث يكون دخل كل كتلة هو خرج جميع الكتل السابقة.



الشكل (5): بنية DU مكونة من 5 كتل DBs

يُظهر الشكل (6) مثالاً لكتلة شبكة كثيفة الاتصالDB مكونة من طبقتي التفاف واتصال تخطي واحد. في الالتفاف الأول conv1 يتم تقليل الحجم المكاني للمدخلات بعدد أقل من المرشحات بحجم 1×1 لتقليل التعقيد الحسابي في طبقة الالتفاف التالية conv2. في الالتفاف الثاني conv2، يتم استخدام المرشحات ذات الحجم الأكبر مثل 3×3، لتعلم السمات ذات الحجم المكاني نفسه.

في هذه الوحدةDU، تتلقى كل كتلة مدخلات ليس فقط من بيانات خرج الكتلة السابقة ولكن أيضاً من خرج جميع الكتل السابقة. بالإضافة إلى ذلك، هناك بارامتر معدل النمو k للتحكم في الحجم المكاني للمدخلات والمخرجات للكتل حيث أنه إذا كان الحجم المكاني للمدخلات a، فإن الحجم المكاني للخرج هو a+k والذي يتحقق من خلال العملية الالتفافية باستخدام k من المرشحات.



الشكل (6): بنية DenseNet Block

3. وحدة التجميع Pooling Unit (PU): تتكون وحدة التجميع من طبقة تجميع واحدة من نوعالتجميع المتوسط average-poolingأو تجميع بالحد الأقصى max-pooling.

يوضح الشكل (7) مثالاً لبنية فرد مكون من 8 وحدات.

RU	DU	PU	RU	PU	PU	DU	PU

الشكل (7): بنية فرد مكون من 8 وحدات

3. التعقيد الحسابي

تُعد نظرية التعقيد إحدى أجزاء نظرية الحوسبة وتتعامل مع الموارد المطلوبة في عملية الحوسبة حيث يعتبر الزمن أكثر هذه الموارد شيوعاً (بمعنى كم من الخطوات أو ما يقابلها من الوقت يلزم لحل المسألة) بالإضافة إلى المكان (بمعنى ما حجم الذاكرة اللازمة لحل المسألة)، ويمكن أن يدخل بالاعتبار موارد أخرى مثل عدد المعالجات المتوازية اللازمة لإنجاز الحساب باستخدام برمجة متوازية[16].

في الوقت الحالي، أصبح تركيز المبرمجين ينصب بشكل خاص على حساب الوقت الذي تستغرقه الخوارزمية حتى تتنفذ وليس على المساحة التي تحتاجها لأن ذاكرات الأجهزة (RAMs) أصبحت كبيرة جداً ولا تشكل أي عائق بالنسبة للمبرمجين كما كانت قديماً، لذلك عند تقدير وقت تنفيذ الخوارزمية فإننا لا ننظر لأفضل احتمال ممكن أو للاحتمال الوسطي بل ننظر لأسوء احتمال ممكن وبما أننا نهتم فقط بمعرفة أسوء الاحتمالات الممكنة فإننا نركز على حساب الـ Big O Notation دون سواها وهي طريقة تحدد مدى سرعة البرنامج أو الخوارزمية نسبة الى عدد الخطوات فيه عندما يزيد حجم المشكلة أي وفقًا لكيفية نمو متطلبات زمن التشغيل أو حجم الذاكرة مع نمو حجم الإدخال[17].

فيما يلي سنقوم بدراسة التعقيد الحسابي في بنية شبكةCNNبسيطة مكونة من وحدة كثيفة الاتصال DU (تحتوي علىNمن الكتل كثيفة الاتصالDBs)، ووحدة متبقية RU (تحتوي على N من الكتل المتبقيةRBs)، ووحدة تجميعPU.

يتم حساب التعقيد الحسابي لهذه البنية باستخدام Big O Notation بناءً على عدد الاتصالات التي تتلقاها كل وحدة كدخل بحيث لن ندخل في تفاصيل عدد وحجم المرشحات وبالتالي عدد العمليات التي تتم في كل وحدة وإنما سيكون التركيز على عدد الاتصالات في هذه البنى وتأثيرها على زمن التنفيذ والذاكرة كما يلي:

أولاً: التعقيد الحسابي فيDenseNet Unit(DU):

تتلقى كل كتلة في هذه الوحدة دخل من خرج جميع الكتل السابقة في الوحدة، بمعنى آخر كل كتلة n تتلقى دخل من جميع الكتل التي تحقق المتراجحة التالية:

$$n - k \geq 0, \quad wherek = 0,1, \dots, n \qquad (1)$$

حيث n هو رقم الكتلة ضمن الوحدة.

وبالتالي يوضح الجدول (1) بنية وحدة DU مكونة من N كتلة كثيف الاتصال DB حيث يمثل كل سطر الدخل الذي تتلقاه الكتلة من خرج الكتل السابقة لها وتمثل القيمة X_0رقم الدخل الأولي للوحدة.

الجدول (1): دخل كل كتلة في الوحدة الكثيفة الاتصال DU

	X_0	DB_1-out	DB_2-out	DB_3-out	DB_4-out	…..	DB_{N-1}-out	DB_N-out
DB_1	×							
DB_2	×	×						
DB_3	×	×	×					
DB_4	×	×	×	×				
⋮								⋮
DB_{N-1}	×	×	×	×	×	…..		
DB_N	×	×	×	×	×	…..	×	

نلاحظ من الجدول (1) أن عدد الاتصالاتلكل كتلة DB_n هو n اتصال (أي تتلقى الدخل من خرج n كتلة سابقة لها) وعدد الاتصالات الكلي في الوحدة الكثيفة الاتصال هو $(\frac{n\,(n+1)}{2})$.

بتطبيق BigO على كل كتلة في الوحدة نجد ما يلي:

–بالنسبة للكتلة الأولى DB_1: دخل هذه الكتلة هو دخل الوحدة الأولي فقط أي X_0 كما هو موضح بالجدول وبالتالي سيكون التعقيد في الكتلة الأولى DB_1وفقاً لما سبق هو $O(1)$.

–الكتلة الثانية DB_2: دخل هذه الكتلة هو X_0 بالإضافة إلى خرج الكتلة الأولى DB_1–out وبالتالي سيكون التعقيد في الكتلة الثانية DB_2وفقاً لما سبق هو $O(2)$.

–وبنفس الطريقة يتم حساب التعقيد في الكتل التالية ليكون$O(3)$، $O(4)$، ... وهكذا ليصبح التعقيد في الكتلة الأخيرة DB_N هو $O(N)$.

وبالتالي يكون التعقيد الكلي لوحدة شبكة كثيفة DenseNet Unit ذات N كتلة هو مجموع التعقيد في كل كتلة، أي:

$$totalcomplexity = 1 + 2 + 3 + \cdots + (N-1) + N(2)$$

نلاحظ أن العلاقة السابقة (2) تمثل مجموع حدود متتالية حسابية منتهية، يُعطى الحد العام لها بالعلاقة التالية :[]

$$S_n = \frac{n}{2}(a_0 + a_n)(3)$$

حيث: n هو عدد الحدود، a_0 هو الحد الأول و a_n هو الحد الأخير.

وبالتالي يصبح التعقيد الكلي:

$$totalcomplexity = \frac{N}{2}(1 + N) = \frac{N^2 + N}{2}(4)$$

وبما أننا في التعقيد نأخذ الحالة الأسوأ، يصبح التعقيد الكلي لوحدة الشبكة الكثيفة الاتصال DU هو $O(N^2)$وهو قيمة كبيرة جداً وفقاً لنظرية التعقيد.

ثانياً: التعقيد الحسابي في ResNet Unit(RU):

تتلقى كل كتلة في هذه الوحدة دخل من خرج الكتلة السابقة في الوحدةوبالتالي يوضح الجدول (2) بنية وحدة RU مكونة من N كتلة RB حيث يمثل كل سطر الدخل الذي تتلقاه الكتلة من خرج الكتل السابقة لها وتمثل القيمة X_0 رقم الدخل الأولي للوحدة.

الجدول (2): دخل كل كتلة في وحدة الشبكة المتبقية RU

	X0	RB1-out	RB2-out	RB3-out	RB4-out	…..	RBN-1-out	RBN-out
RB1	×							
RB2		×						
RB3			×					
RB4				×				
⋮								⋮
RBN-1						…..		
RBN						…..	×	

نلاحظ من الجدول (2) أن عدد الاتصالاتلكل كتلة RBn هو اتصال واحد (أي تتلقى الدخل من خرج الكتلة سابقة لها فقط).

بتطبيق BigO على كل كتلة في الوحدة يكون التعقيد الكلي لوحدة شبكة متبقية ResNet Unit ذات N كتلة هو مجموع التعقيد في كل كتلة، أي:

$$totalcomplexity = 1 + 1 + 1 + \cdots + 1 + 1 = N(5)$$

وبالتالي يصبح التعقيد الكلي لوحدة الشبكة المتبقية RU هو O(N) وهو قيمة متوسطة وفقاً لنظرية التعقيد.

ثالثاً: التعقيد الحسابي في Pooling Unit:

تتلقى هذه الوحدة الدخل من الوحدة السابقة لها فقط وبالتالي يكون التعقيد الحسابي لها هو حسب عدد الاتصالات O(1).

وفقاً لما سبق نجد أن التعقيد الكلي لبنية شبكة CNN بسيطة مكونة من وحدة كثيفة الاتصال DBU (تحتوي على N من الكتل كثيفة الاتصال DBs)، وحدة متبقية RBU (تحتوي على N من الكتل المتبقية RBs)، وحدة تجميع PU هو مجموع التعقيد لكل وحدة أي:

$$totalcomplexity = N^2 + N + 1 \qquad (6)$$

وبما أنه يتم اختيار الحالة الأسوأ عند دراسة التعقيد يكون التعقيد الكلي هو $O(N^2)$ وهو قيمة كبيرة جداً وفقاً لنظرية التعقيد.

من أجل تخفيض قيمة تعقيد بنية الشبكة البسيطة التي قمنا بدراستها (المكونة من وحدة كثيفة الاتصال DBU (تحتوي على N من الكتل كثيفة الاتصال DBs)، وحدة متبقية RBU (تحتوي على N من الكتل المتبقية RBs)، وحدة تجميع PU)، قمنا دراسة عنصر التصميم الأساسي لبنية الشبكة الكثيفة الاتصال DenseNet التي تسبب كما رأينا أكبر قيمة تعقيد والتي تتمثلبوصلات التجميع الداخلية التي تُعرف باتصالات التخطي، فعلى الرغم من أن اتصالات التخطي هذه هي السبب في نجاح DenseNet كما تم شرحه سابقاً إلا أن البنية المحددة للتجميع في هذه الشبكات تهدر السعة في الواقع من خلال تخصيص الكثير من البارامترات والكثير من الحسابات على طول روابط التجميع الداخلية.

من أجل ذلك سنقوم باختبار مبدأ تصميم يعتمد على استخدام عدد محدد من اتصالات التخطي، بدلاً من قطع الاتصالات بشكل تعسفي أو ما بعد التصميم سيتم الانتقال من طوبولوجيا التجميع الكامل إلى أخريكون فيها عدد الروابط الداخلة لكل كتلةلوغاريتمي وليس خطيبحيث نحافظ على اتصالات قصيرة بين الكتل ضمن الوحدة بالإضافة إلى تقليل عدد هذه الاتصالات ضمن الوحدة، وبالتالي سيتم تحديد عدد الاتصالات في كل كتلة n إلى $log_2 n$ بدلاً من n اتصال.

يوضح الجدول (3) بنية وحدة شبكة كثيفة الاتصال DU مكونة من 8 كتل DB حيث يمثل كل سطر الدخل الذي تتلقاه الكتلة من خرج الكتل السابقة لها وتمثل القيمة X0 رقم الدخل الأولي للوحدة، وذلك بعد أن قمنا بتقليل عدد الاتصالات فيها حيث أصبحت كل كتلة تتلقى مدخلات من الكتل السابقة التي تحقق المتراجحة التالية (6) بدلاً من المتراجحة السابقة (1):

$$n - 2^k \geq 0 \qquad k = 0,1, \dots, log_2 n \qquad (7)$$

حيث n هو رقم الكتلة ضمن الوحدة.

بتطبيق المتراجحة السابقة (7) نجد أن كل كتلة في الوحدة تتلقى الدخل من الكتل السابقة وفق التالي:

- من أجل الكتلة الأولى DB_1 نقوم بتطبيق المتراجحة عندما n=1:

$$1 - 2^0 = 0 \geq 0$$

أي أن الكتلة DB_1 ستتلقى الدخل من الدخل X_0فقط.

- من أجل الكتلة DB_2 نقوم بتطبيق المتراجحة عندما n=2:

$$2 - 2^0 = 1 \geq 0, \qquad 2 - 2^1 = 0 \geq 0$$

أي أن الكتلة DB_2 ستتلقى الدخل من X_0 وخرج الكتلة DB_1.

- من أجل الكتلة DB_3 نقوم بتطبيق المتراجحة عندما n=3:

$$3 - 2^0 = 2 \geq 0, \qquad 3 - 2^1 = 1 \geq 0$$

أي أن الكتلة DB_3 ستتلقى الدخل من خرج الكتلة DB_1 وخرج الكتلة DB_2.

- من أجل الكتلة DB_4 نقوم بتطبيق المتراجحة عندما n=4:

$$4 - 2^0 = 3 \geq 0, \quad 4 - 2^1 = 2 \geq 0, \quad 4 - 2^2 = 0 \geq 0$$

أي أن الكتلة DB_4 ستتلقى الدخل من X_0وخرج الكتلة DB_2 وخرج الكتلة DB_3.

- من أجل الكتلة DB_5 نقوم بتطبيق المتراجحة عندما n=5:

$$5 - 2^0 = 4 \geq 0, \quad 5 - 2^1 = 3 \geq 0, \quad 5 - 2^2 = 1 \geq 0$$

أي أن الكتلة DB_5 ستتلقى الدخل من خرج الكتلة DB_1 وخرج الكتلة DB_3 وخرج الكتلة DB_4.

وبنفس الطريقة نجد أن كل كتلة في الوحدة تتلقى الدخل من الكتل السابقة وفق الجدول التالي:

الجدول (3): دخل كل كتلة في الوحدة الكثيفة الاتصال DU بعد تخفيض عدد الاتصالات

	X_0	DB_1-out	DB_2-out	DB_3-out	DB_4-out	DB_5-out	DB_6-out	DB_7-out
DB_1	×							
DB_2	×	×						
DB_3		×	×					
DB_4	×		×	×				
DB_5		×		×	×			
DB_6			×		×	×		
DB_7				×		×	×	
DB_8	×				×		×	×

نلاحظ من الجدول (3) أن عدد الاتصالاتلكل كتلة DB_n هو $log_2 n + 1$ اتصال وبتطبيق BigO على كل كتلة في الوحدة نجد ما يلي:

–بالنسبة للكتلة الأولى DB_1: دخل هذه الكتلة هو دخل الوحدة الأولي فقط أي X_0 كما هو موضح بالجدول وبالتالي سيكون التعقيد في الكتلة الأولى DB_1وفقاً لما سبق هو $O(1)$.

–الكتلة الثانية DB_2: دخل هذه الكتلة هو X_0 بالإضافة إلى خرج الكتلة الأولى DB_1–out وبالتالي سيكون التعقيد في الكتلة الثانية DB_2وفقاً لما سبق هو $O(2)$.

–الكتلة الثانية DB_3: دخل هذه الكتلة هو خرج الكتلة الأولى DB_1–out وخرج الكتلة الثانية DB_2– out وبالتالي سيكون التعقيد في الكتلة الثالثة DB_3وفقاً لما سبق هو $O(2)$.

–وبنفس الطريقة يتم حساب التعقيد في الكتل التالية ليكون$O(3)$، $O(3)$، ... وهكذا ليصبح التعقيد في الكتلة الأخيرة DB_N هو $O(\log_2(N) + 1)$.

وبالتالي يكون التعقيد الكلي لوحدة شبكة كثيفة الاتصال DenseNet Unit ذات N كتلة هو مجموع التعقيد في كل كتلة، أي:

$$totalcomplexity = 1 + 2 + 2 + 2 + \cdots + (\log_2(N-1) + 1) + (\log_2(N) + 1) \quad (8)$$

نلاحظ أن العلاقة السابقة (8) تمثل أيضاً مجموع حدود متتالية حسابية منتهية، يُعطى الحد العام لها كما في العلاقة (3). وبالتالي يصبح التعقيد الكلي:

$$totalcomplexity = \frac{N}{2}(1 + (\log_2(N) + 1)) = \frac{2N + N\log_2(N)}{2} = N + \frac{N\log_2(N)}{2} \quad (9)$$

وبما أننا في التعقيد نأخذ الحالة الأسوأ، يصبح التعقيد الكلي لوحدة الشبكة الكثيفة الاتصال DU وفق الطريقة المقترحة هو $O(N\log_2(N))$.

بمقارنة التعقيد بين الشبكات الكثيفة DenseNet والتعقيد في هذه الشبكات بعد تطبيق الطريقة المقترحة وفق نظرية التعقيد الحسابيةنلاحظ أنه عندما قمنا بتخفيض عدد الاتصالات بين الكتل في الوحدة الكثيفة الاتصال UnitDenseNet انخفض التعقيد الكلي للوحدة من $O(N^2)$ وأصبح $O(N\log_2(N))$.

وبالتالي يصبح التعقيد الكلي لبنية شبكة CNN بسيطة مكونة من وحدة كثيفة الاتصال DBU (تحتوي على N من الكتل كثيفة الاتصال DBs)، وحدة متبقية RBU (تحتوي على N من الكتل المتبقية RBs)، وحدة تجميع PU بعد تخفيض عدد الاتصالات بين الكتل في الوحدة الكثيفة الاتصال هو مجموع التعقيد لكل وحدة أي:

$$totalcomplexity = N\log_2(N) + N + 1 = N(\log_2(N) + 1) + 1 \quad (10)$$

وبما أنه يتم اختيار الحالة الأسوأ عند دراسة التعقيد يكون التعقيد الكلي هو $O(N(\log_2(N) + 1))$ وهو قيمة متوسطة وفقاً لنظرية التعقيد وأقل من التعقيد الكلي السابق $O(N^2)$.

النتائج والمناقشة

من أجل تقييم أداء الخوارزمية تم تنفيذ مراحلالخوارزميةالموضحة في الشكل (2) بعد إجراء التغييرات على بنية وحدة الشبكة الكثيفة الاتصال DenseNet Unitباستخدام لغة python وذلك بالاعتماد على برنامجPyCharmوهو برنامج تطويري يدعم لغةpython، ومن ثم تم اختبار الخوارزمية على قواعد البيانات المرجعيةCifar10وCifar100باستخدام وحدة المعالجة الرسوميةNvidia GeForce GTX 1060.

تم التحقق من أداء الخوارزمية بعد إجراء التعديلات عليها تجريبياً من حيث خطأ التصنيف وعدد البارامترات، بالإضافة إلى التعقيد الحسابي المتمثل بعدد الأيام التي استغرقتها عملية التدريب باستخدام GPU ومقارنة النتائج مع مجموعة من الخوارزميات المختارة كما هو موضح في الجدول (4) والجدول (5).

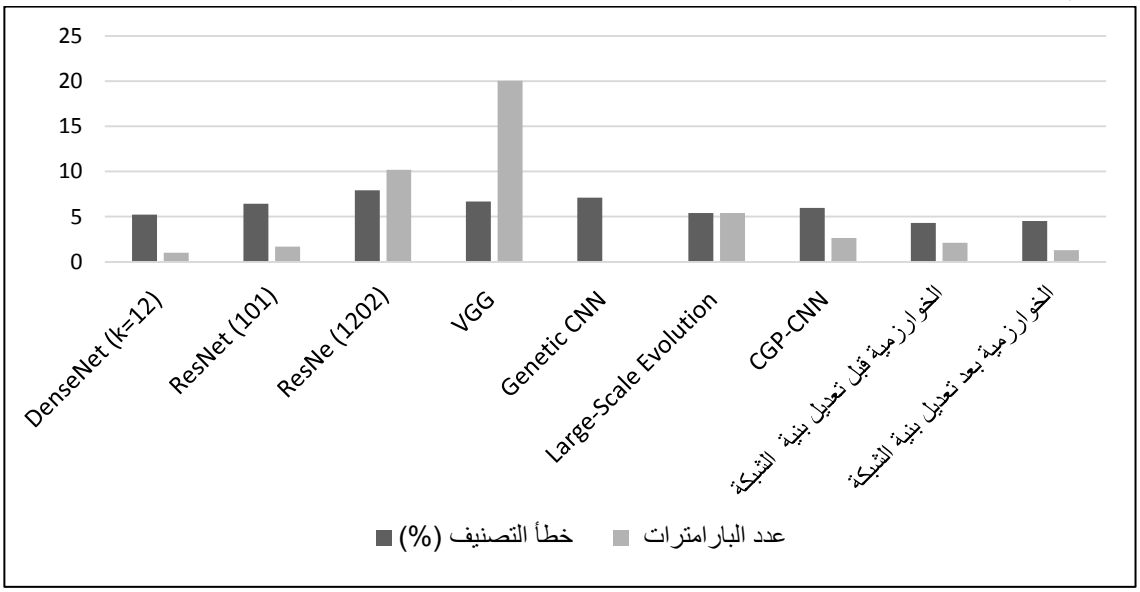
الجدول (4): مقارنة بين الخوارزمية المقترحة ومجموعة من الخوارزميات بعد ضغط البنية على CIFAR10.

	CIFAR10	Parameters ($M = 10^6$)	زمن تنفيذ الخوارزمية (يوم)
DenseNet(k=12) [7]	5.24	1.0 M	-
ResNet(depth=101) [6]	6.43	1.7 M	-
ResNet (depth=1202) [6]	7.93	10.2 M	-
VGG [18]	6.66	20.04 M	-
Genetic CNN [8]	7.1	-	17
Large-Scale Evolution [9]	5.4	5.4 M	-
CGP-CNN [10]	5.98	2.64 M	27
الخوارزمية بنية الشبكة [13]	4.3	2.1 M	40
الخوارزمية بعد تعديل بنية الشبكة	4.5	1.3 M	31

يوضح الجدول (4) النتائج التجريبية للخوارزمية المقترحة والخوارزميات المختارة حيث يشير العمود الأول إلى الخوارزميات المختارة للمقارنة، العمود الثاني يشير إلى خطأ التصنيف على مجموعة البيانات CIFAR10، العمود الثالث يشير إلى عدد البارامترات والعمود الرابع يشير إلى المدة المستغرقة في عملية التدريب للوصول إلى أفضل بنية. بالإضافة إلى ذلك، يشير الرمز "–" في الجدول إلى أنه لم يتم الإبلاغ عن نتيجة علنية من قبل الخوارزمية المقابلة.

تُظهر النتائج أن التعديل الذي قمنا به على بنية الشبكة أدى إلى انخفاض بسيط في الدقة (أي زيادة خطأ التصنيف) مقارنة مع الدقة التي تم التوصل إليها قبل إجراء هذا التعديل وضغط بنية الشبكة، ومع ذلك تبقى الخوارزمية بعد التعديل تتفوق على جميع شبكات CNN الحديثة المصممة يدويًا (DenseNet، ResNet، VGG) والخوارزميات التلقائية التي تتم المقارنة معها (Genetic CNN،Large-Scale Evolution، CGP-CNN) من حيث معدل خطأ التصنيف عل مجموعة البيانات CIFAR10.

كما توضح النتائج أن عدد بارامترات أفضل بنية تم التوصل إليها باستخدام الخوارزمية المقترحة أقل من عدد البارامترات في جميع الخوارزميات التلقائية التي تمت المقارنة معها (Genetic CNN، Large-Scale Evolution، CGP-CNN) وذلك خلال زمن تنفيذ للخوارزمية المقترحة قدره 31 يوم وفقاً للموارد الحاسوبية المتاحة لدينا.



الشكل (8): مقارنة بين الخوارزمية المقترحة ومجموعة الخوارزميات قبل ضغط بنية الشبكة على Cifar10

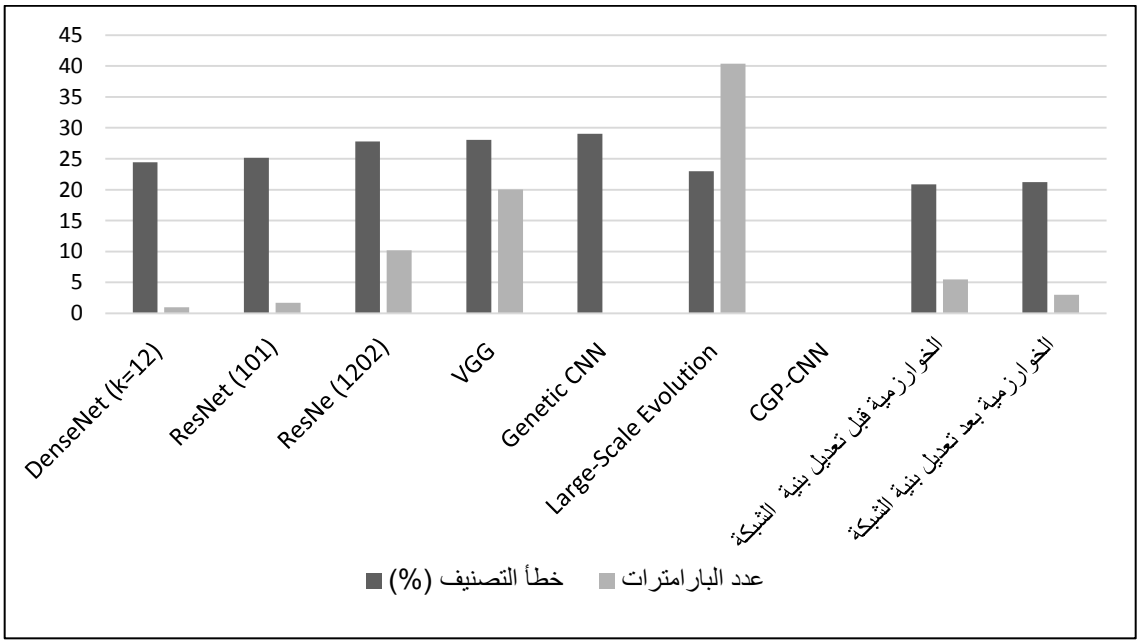
يظهر الشكل (8) المقارنة بين الخوارزمية بعد تعديل بنية الشبكة مع الخوارزميات المختارة من حيث خطأ التصنيف (%) وعدد بارامترات أفضل بنية شبكة تم التوصل إليها (10^6) وذلك على مجموعة البيانات Cifar10. لا يعتبر زمن تنفيذ الخوارزمية كمقياس للمقارنة مع الخوارزميات المختارة بسبب اختلاف الموارد الحاسوبية التي تم استخدامها في تنفيذ كل خوارزمية.

الجدول (5): مقارنة بين الخوارزمية المقترحة ومجموعة الخوارزميات بعد ضغط البنية على CIFAR100.

	CIFAR100	Parameters ($M = 10^6$)	زمن تنفيذ الخوارزمية (يوم)
DenseNet(k=12) [7]	24.42	1.0 M	-
ResNet(depth=101) [6]	25.16	1.7 M	-
ResNet (depth=1202) [6]	27.82	10.2 M	-
VGG [18]	28.05	20.04 M	-
Genetic CNN [8]	29.05	-	17
Large-Scale Evolution [9]	23	40.4 M	-
CGP-CNN [10]	-	-	-
الخوارزمية قبل تعديل بنية الشبكة[13]	20.85	5.5 M	84
الخوارزمية بعد تعديل بنية الشبكة	21.2	3 M	75

بنفس الطريقة يوضح الجدول (5) النتائج التجريبية للخوارزمية المقترحة والخوارزميات المختارة بالنسبة لقاعدة البيانات CIFAR100 حيث تُظهر النتائج أن التعديل الذي قمنا به على بنية الشبكة أدى أيضاً إلى انخفاض بسيط في الدقة (أي زيادة خطأ التصنيف) مقارنة مع الدقة التي تم التوصل إليها قبل إجراء هذا التعديل وضغط بنية الشبكة، ومع ذلك تبقى الخوارزمية بعد التعديل تتفوق على جميع شبكات CNN الحديثة المصممة يدويًا (DenseNet، ResNet، VGG) كما تتفوق على الخوارزميتين التلقائيتين (Genetic CNN، Large-Scale Evolution).

كما توضح النتائج أن عدد بارامترات أفضل بنية تم التوصل إليها باستخدام الخوارزمية المقترحة أقل من عدد البارامترات في جميع الخوارزميات التلقائية التي تمت المقارنة معها (Genetic CNN، Large-Scale Evolution، CGP-CNN) وذلك خلال زمن تنفيذ للخوارزمية المقترحة قدره 75 يوم وفقاً للموارد الحاسوبية المتاحة لدينا.



الشكل (9): مقارنة بين الخوارزمية المقترحة ومجموعة الخوارزميات بعد ضغط بنية الشبكة على Cifar100

يوضح الشكل (9) المقارنة بين الخوارزمية بعد تعديل بنية الشبكة مع الخوارزميات المختارة من حيث خطأ التصنيف (%) وعدد بارامترات أفضل بنية شبكة تم التوصل إليها (10^6) وذلك على مجموعة البيانات Cifar100. لا يعتبر زمن تنفيذ الخوارزمية كمقياس للمقارنة مع الخوارزميات المختارة بسبب اختلاف الموارد الحاسوبية التي تم استخدامها في تنفيذ كل خوارزمية.

الاستنتاجات والتوصيات

- تم في هذا البحث دراسة التعقيد الحسابي لبنى الشبكات الالتفافية التي يتم توليدها في هذه الخوارزمية. اعتمدت إستراتيجية الترميز المقترحة في الخوارزمية على بنى الشبكات التي تم تصميمها بشكل يدوي واختبارها حيث أعطت نتائج فعالة في مجال التعرف على النماذج وهي الشبكات المتبقية ResNet والشبكات كثيفة الاتصال DenseNet التي تعد من أحدث شبكات CNN المصممة من قبل الخبراء بشكل يدوي والتي تغلبت على مشكلة تلاشي مشتقات تابع الخطأ عن طريق استخدام اتصالات التخطي.
- قمنا باختبار مبدأ تصميم يعتمد على استخدام عدد محدد من اتصالات التخطي بدلاً من قطع الاتصالات بشكل تعسفي أو ما بعد التصميم حيث تم الانتقال من طوبولوجيا التجميع الكامل في وحدة الشبكة كثيفة الاتصال DenseNet Unit التي تسبب أكبر قيمة تعقيد إلى أخرى يكون فيها عدد الروابط الداخلة لكل كتلة لوغاريتمي وليس خطي بحيث نحافظ على اتصالات قصيرة بين الكتل ضمن الوحدة بالإضافة إلى تقليل عدد هذه الاتصالات ضمن الوحدة.
- بعد دراسة وتطبيق الخوارزمية التي تم اقتراحها في هذا البحث ومناقشة النتائج التي تم التوصل إليها وجدنا أن هذا التغيير البسيط في التصميم يوفر أداءً فائقاً حيث أن انخفاض عدد الاتصالات في وحدة الشبكة كثيفة الاتصالDenseNet Unitأدى إلى:
  - انخفاض عدد البارامترات.
  - انخفاض حجم الذاكرة المستهلكة.
  - انخفاض زمن تنفيذ الخوارزمية (زمن الوصول إلى أفضل بنية شبكة).
  - ولكن بالمقابل أدى إلى انخفاض بسيط في الدقة (أي زيادة خطأ التصنيف) ومع ذلك تبقى هذه الدقة أعلى من الدقة في جميع الخوارزميات الحديثة التي تمت المقارنة معها.
- في العمل الحالي تم استخدام الخوارزمية الجينية فقط لاستكشاف بنية الشبكة في حين يتم إجراء التدريب على الشبكة بشكل منفصل، لذلك سيكون من المثير للاهتمام للغاية دمج الخوارزمية الجينية لتدريب بنية الشبكة والأوزان في وقت واحد.

المراجع

[1] Felsberg, M. (2017). Five years after the Deep Learning revolution of computer vision: State of the art methods for online image and video analysis.

[2] LeCun, Y., Kavukcuoglu, K., & Farabet, C. (2010, May). Convolutional networks and applications in vision. In *Proceedings of 2010 IEEE international symposium on circuits and systems* (pp. 253-256). IEEE.

[3] Bratcu, A. I., Makdessian, L., & Dolgui, A. (2003, July). Minimisation of equipment cost for transfer lines with blocks of parallel tasks. In *Proceedings of the IEEE International Symposium onAssembly and Task Planning, 2003.* (pp. 109-114). IEEE.

[4]Chehade, H., Dolgui, A., Dugardin, F., Makdessian, L., & Yalaoui, F. (2012). Multi-objective approach for production line equipment selection. *Management and Production Engineering Review*, *3*, 4-17.

[5] Dayoub, Y. (2015). Stochastic logicallinguistic approachMulti-level automated object's dialogue control (MADC). *سلسلة العلوم الهندسية ISSN: 2079-3081*, *35*(6).

[6] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770-778).

[7]Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4700-4708).

[8] Xie, L., & Yuille, A. (2017). Genetic cnn. In *Proceedings of the IEEE international conference on computer vision* (pp. 1379-1388).

[9] Real, E., Moore, S., Selle, A., Saxena, S., Suematsu, Y. L., Tan, J., ... & Kurakin, A. (2017). Large-scale evolution of image classifiers. *arXiv preprint arXiv:1703.01041*.

[10] Suganuma, M., Shirakawa, S., & Nagao, T. (2017, July). A genetic programming approach to designing convolutional neural network architectures. In *Proceedings of the genetic and evolutionary computation conference* (pp. 497-504).

[11] Han, S., Pool, J., Tran, J., & Dally, W. J. (2015). Learning both weights and connections for efficient neural networks. *arXiv preprint arXiv:1506.02626*.

[12] Kim, Y. D., Park, E., Yoo, S., Choi, T., Yang, L., & Shin, D. (2015). Compression of deep convolutional neural networks for fast and low power mobile applications. *arXiv preprint arXiv:1511.06530*.

[13]Ibrahim, K., Dayoub, Y., & Makdessian, L. (2021). Improving the performance of convolutional neural networks using evolutionary computing. *Association of Arab Universities Journal of Engineering Sciences*, 28(2).

[14] Khan, A., Sohail, A., Zahoora, U., & Qureshi, A. S. (2020). A survey of the recent architectures of deep convolutional neural networks. *Artificial Intelligence Review*, *53*(8), 5455-5516.

[15] Wu, J. (2019). Convolutional neural networks. *Published online at https://cs. nju. edu. cn/wujx/teaching/15 CNN. pdf*.

[16] HOGAN, S. (2011). A gentle introduction to computational complexity theory, and a little bit more.

[17] Devi, S. G., Selvam, K., & Rajagopalan, S. P. (2011). An abstract to calculate big o factors of time and space complexity of machine code.

[18] Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.