

## تأثير نوع معامل العبور على أداء الخوارزميات الجينية المستخدمة في التعلم العميق

م. كلودا إبراهيم\*

(تاريخ الإيداع 2/8/2022 . قُبِلَ للنشر في 2022/6/14 )

### □ ملخص □

تم في هذا البحث دراسة تأثير نوع معامل العبور crossover operator على أداء الخوارزمية الجينية المستخدمة في تصميم بنى الشبكات العصبونية التلافيفية العميقة Convolutional Neural Networks (CNN) التي تعتمد في بنيتها على اتصالات التخطي. 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 واعدًا في هذا المجال. حققت هذه الخوارزمية أفضل دقة بين جميع الخوارزميات الحديثة التي تمت المقارنة معها على مجموعة البيانات Cifar10 ومجموعة البيانات Cifar100.

**الكلمات المفتاحية:** الشبكات العصبونية العميقة، CNN، الخوارزميات الجينية، التعلم العميق، التعرف على النماذج، معاملات العبور.

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

# The effect of the type of crossover operator on the performance of genetic algorithms used in deep learning

Eng. Kloda Ibrahim \*

(Received 8/2/ 2022 . Accepted 14/6/ 2022)

## □ ABSTRACT □

In this paper, we have studied the effect of the type of crossover operator on the performance of the genetic algorithm used in the design of deep convolutional neural networks (CNN) architectures that relies on skip connections in its architecture. The performance of the algorithm was evaluated by conducting a series of experiments on the widely used reference datasets for image classification tasks and comparing the results with modern algorithms that showed promising performance in this field. This algorithm achieved the best accuracy among all the modern algorithms that were compared on the Cifar10 dataset and the Cifar100 dataset.

**Keywords:** Deep Neural Networks, CNN, Genetic Algorithms, Deep Learning, Crossover operators.

---

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

## مقدمة

تعد الرؤية الحاسوبية من مجالات علوم الحوسبة الحديثة وشكل من أشكال الذكاء الاصطناعي، حيث تساعد هذه التقنية على رؤية العالم وتحليل البيانات المرئية لاتخاذ القرارات واكتساب فهم حول البيئة والعالم، بالإضافة إلى تحديد ومعالجة الأشياء مثل الصور ومقاطع الفيديو بنفس الطريقة التي يفعلها البشر. حتى وقت قريب كانت الرؤية الحاسوبية تعمل بقدرة محدودة، إلا أن كمية البيانات الرقمية الهائلة التي تُنتجها اليوم مثلت إحدى العوامل الرئيسية الدافعة وراء تطور الرؤية الحاسوبية.

أظهر التعلم العميق، الذي يستخدم الشبكات العصبية العميقة كنموذج، أداءً جيداً في العديد من مهام الذكاء الاصطناعي والتعلم الآلي مثل التعرف على الصور، التعرف على الكلام، ومهام التعلم المعزز. على وجه الخصوص، شهدت الشبكات العصبية التلافيفية في السنوات القليلة الماضية نجاحاً كبيراً في مجال الرؤية الحاسوبية وتحديداً في مهام التعرف على النماذج. [15]

توفر شبكات CNN درجة عالية جداً من الدقة مقارنة بطرق التعلم الآلي الأخرى، حيث تتميز هذه الشبكات في قدرتها على استخراج الميزات تلقائياً دون تدخل بشري وتستخدم معالجة أولية قليلة نسبياً مقارنة بخوارزميات تصنيف الصور الأخرى.

يعتمد أداء شبكات CNN بشكل كبير على بنيتها، حيث أنه عادةً ما ينتج عن بنيتها الأعمق (كلما زاد عدد الطبقات) مستوى أفضل من الدقة، ومع ذلك، هناك بعض الحالات كلما كانت البنية أعمق تنتج مستوى أسوأ من الدقة، لذلك يعد تحديد بنية CNN أمراً بالغ الأهمية. [21]

يتم تحديد بنية هذه الشبكات من خلال عدة بارامترات تدعى hyper-parameters مثل عدد الطبقات ونوعها وعدد وحجم النوى ونوع تابع التفعيل، حيث أن الطرق التقليدية المستخدمة لتحديد قيم هذه البارامترات هي: البحث الشبكي Grid Search [19]، البحث اليدوي [5] Manual Search والبحث العشوائي [1] Random Search، وهذه الطرق تتطلب خبرة واسعة في كل من شبكات CNNs ومجال المشكلة التي تم التحقيق فيها والتي ليست متاحة بالضرورة لكل مستخدم بالإضافة إلى العبء الحسابي الكبير لمساحات البحث الكبيرة.

من ناحية أخرى، في الوقت الحاضر يتم استخدام الحوسبة التطورية (الخوارزميات الجينية، خوارزميات الأسراب وغيرها) بشكل متزايد لحل مشاكل الأمثلة [3] [4] [2] ومنها تحديد البارامترات المثلى للدالة، وبما أن تصميم بنية CNN يرتبط ارتباطاً وثيقاً باختيار العديد من البارامترات لذلك بدأ الاهتمام في تطبيق استراتيجيات الحوسبة التطورية في تحديد البارامترات المثلى في تشكيل بنية CNN [9].

تواجه الشبكات العصبية التلافيفية (CNN) الخاصة بمهام الرؤية الحاسوبية عدة تحديات تعيق نشرها في تطبيقات العالم الحقيقي وأهمها حجم النموذج، ذاكرة وقت التشغيل وعدد العمليات الحسابية [15]. في السنوات الأخيرة، اكتسبت الشبكات العصبية التلافيفية أهمية كبيرة حيث تعد [6] ResNet و [8] DenseNet من أحدث شبكات CNN المصممة يدوياً والتي حققت نجاحاً كبيراً في مجال التعرف على النماذج، ومن ثم تم اقتراح خوارزميات متعددة لتصميم بني CNN أهمها [23] Genetic CNN، [20] Large-Scale Evolution، [22] CGP-CNN.

## أهمية البحث وأهدافه

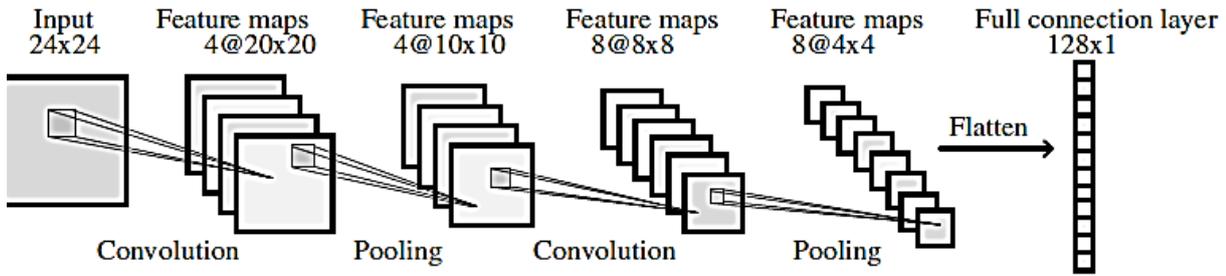
يستمد هذا البحث أهميته من ظهور الشبكات العصبية التلافيفية كنماذج رائدة لمهام التعرف على النماذج بالإضافة إلى فعالية الخوارزميات الجينية في حل المشاكل المعقدة ذات فضاء البحث الكبير. يهدف هذا البحث إلى اختبار تأثير نوع معامل العبور crossover المستخدم على أداء الخوارزمية الجينية المستخدمة لاكتشاف أفضل بنية لشبكات CNN وفقاً للمهمة المعنية بطريقة تلقائية تماماً.

## طرائق البحث ومواده

### 1. الشبكات العصبية التلافيفية CNN

الشبكات العصبية التلافيفية هي نوع خاص من الشبكات العصبية ذات التغذية الأمامية المشتقة من العمليات البيولوجية في الفص البصري حيث تعتبر حلاً للعديد من مشاكل رؤية الكمبيوتر والذكاء الاصطناعي [11] [15]. تعد LUNET واحدة من أولى الشبكات العصبية التلافيفية التي ساعدت في دفع التعلم العميق، وقد أطلق عليها اسم LUNET-5 وتم استخدامها بشكل أساسي لمهام التعرف على الأحرف مثل قراءة الرموز البريدية والأرقام وما إلى ذلك [14].

في السنوات الأخيرة، اكتسبت الشبكات العصبية التلافيفية أهمية كبيرة حيث تعد ResNet [6] و DenseNet [8] من أحدث شبكات CNN المصممة يدوياً والتي حققت نجاحاً كبيراً في مجال التعرف على النماذج، ومن ثم تم اقتراح خوارزميات متعددة لتصميم بنى CNN استناداً إلى ما إذا كانت المعالجة المسبقة أو المعالجة اللاحقة لبنية CNN مطلوبة عند استخدام هذه الخوارزميات، وفقاً لذلك يمكن تقسيمها إلى فئتين مختلفتين وفقاً للطريقة التي يتم فيها تحديد بارامترات بنية CNN: خوارزميات تصميم بنية CNN شبه التلقائية ( Genetic [23] CNN، [24] Block-QNN-S) وخوارزميات التصميم التلقائية (Large-Scale Evolution [20] CGP-، [22] CNN).



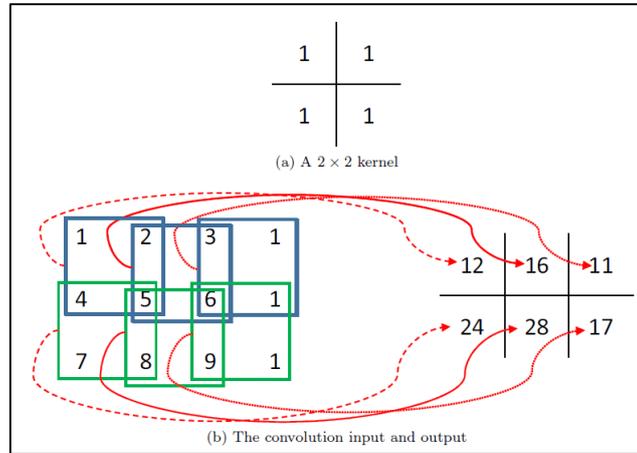
الشكل (1): بنية شبكة CNN

يوضح الشكل (1) مثالاً لبنية شبكة CNN حيث تتكون الشبكة العصبية التلافيفية بشكل عام من عدة طبقات مختلفة لكل منها وظيفتها الخاصة، يتم تصنيف الطبقات الرئيسية لأي شبكة عصبية التلافيفية إلى أربع طبقات وهي [9]:

### 1. الطبقة التلافيفية Convolutional 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 بشكل تلقائي [14]. يوضح الشكل (2) مثالاً لعملية الالتفاف بفرض أن حجم صورة

الإدخال هي  $3 \times 4$  وحجم نواة الالتفاف  $2 \times 2$ ، بحيث يتم البدء بتراكب نواة الالتفاف أعلى صورة الإدخال وحساب الجداء العددي بين الأرقام الموجودة في نفس الموقع في النواة والمدخلات



الشكل (2): عملية الالتفاف convolution operation

تتكون خريطة السمات من عدة قنوات، ترتبط أبعاد هذه القنوات بأبعاد مصفوفة الإدخال (على اعتبار أن دخل هذه الشبكات هو عبارة عن مجموعة من الصور) وأبعاد المرشح (مع اعتبار أنه يجب أن يتم تحديد بعض البارامترات قبل عملية تدريب الشبكة مثل عدد المرشحات، حجم المرشح، البنية الهيكلية للشبكة.. إلخ) بالإضافة إلى العاملين التاليين:

- الخطوة Stride: تمثل عدد العناصر التي يتم إزاحة المرشح بمقدارها بعد كل عملية.
- الحشو Padding: تتمثل عملية الحشو الصفري بإضافة أصفار حول أطراف مصفوفة الصورة المدخلة مما يساعد في تمرير المرشح على أطراف الصورة بشكل أفضل. تفيد عملية الحشو الصفري في التحكم بحجم خرائط السمات الناتجة عن عملية الالتفاف حيث يستخدم عادةً في جعل حجم خرج الطبقة التلافيفية (حجم خرائط السمات) مساوياً لحجم الدخل.

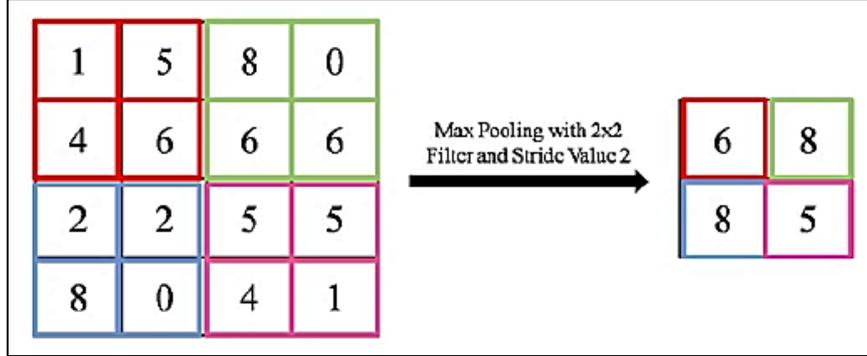
2. **طبقة التنفيل Activation Layer:** بعد انتهاء عملية الالتفاف يتم إدخال خريطة السمات إلى طبقة التنفيل حيث يُطبق تابع التنفيل على كل عصبون أي ما يكافئ عنصر من خريطة السمات. أهم توابع التنفيل المستخدمة في هذا النوع من الشبكات هو تابع الوحدة الخطية المصححة ReLU الذي أثبتت فاعليته مقارنة بالتوابع الأخرى [18].

3. **طبقة التجميع Pooling layer:** بعد تطبيق تابع التنفيل على خريطة السمات، يتم العمل على تقليل أبعادها بطريقة تحافظ على المعلومات من خلال عملية التجميع، ويتم ذلك بعدة طرق أهمها التجميع بالحد الأقصى max-pooling حيث يتم مقابلة كل نافذة (مجموعة من العناصر المجاورة) مع عنصر واحد يمثل أعلى قيمة داخل هذه النافذة [12].

ناتج عملية التجميع عبارة عن خريطة سمات لها نفس العمق ولكنها تختلف في العرض والارتفاع، وبالتالي تتميز عملية التجميع بالعديد من المزايا التي من أبرزها:

- تقليل أبعاد خريطة السمات وعدد المتغيرات والحسابات في الشبكة.
- تجعل الشبكة مقاومة لتغيير طفيف أو تشويه في مصفوفة الإدخال.

يوضح الشكل (3) مثالاً على عملية التجميع بالحد الأقصى Max-pooling حيث تمثل المصفوفة على اليسار خريطة السمات (دخل طبقة التجميع) والمصفوفة على اليمين هي ناتج عملية التجميع عبر نافذة بحجم  $2 \times 2$  وبخطوة 2.



الشكل (3): التجميع بالحد الأقصى Max-pooling

4. الطبقة المتصلة بالكامل **Fully Connected layer**: هذه الطبقة هي الأخيرة في الشبكة التلافيفية وهي من نوع multi-layer perceptron وفيها ترتبط العصبونات بالكامل مع كل عُقد الطبقة السابقة وتتم فيها عملية التصنيف النهائية حيث يكون دخلها شعاع مكوّن من خريطة السمات بعد إجراء عملية التجميع، وخرجها عبارة عن شعاع يعبر عن الصف (الفئة) الذي تنتمي إليه خريطة السمات [16].

## 2. الخوارزميات الجينية Genetic Algorithms

تم استلهام الخوارزمية الجينية من النظرية الداروينية حول "البقاء للأصلح"، من خلال إنتاج كروموسومات جديدة (أفراد) من خلال العبور (crossover) وعمليات الطفرات، أي أنه من المرجح أن يظل الفرد الأصلح ويتزاوج. لذلك سيكون سكان الجيل القادم أقوى لأنهم ينتجون من أفراد أقوى، أي أن الحل يتطور من جيل إلى آخر.

تبدأ GA بعدد من الحلول العشوائية (المجموعة الأولية)، ثم يتم تشفير هذه الحلول (الأفراد) وفقاً للمشكلة الحالية، ويتم تقييم جودة كل فرد من خلال وظيفة اللياقة. تعتمد GA بشكل أساسي على ثلاثة عوامل:

- الاختيار Selection: عملية اختيار الآباء "الأفضل" في المجتمع للتزاوج، ويتم تحديد "الأفضل" بناءً على المشكلة الحالية.

- معامل العبور Crossover operator: يأخذ الوالدين (الكروموسومات)، لتكوين نسل جديد عن طريق تبديل أجزاء من الجينات الأم؛ من الأرجح أن النسل الجديد يحتوي على أجزاء جيدة من والديهم، وبالتالي يؤدي بشكل أفضل مقارنة بأسلافهم.

- معامل الطفرة Mutation operator: يأخذ كروموسوماً واحداً، ويغير بعض جيناته لإنشاء

كروموسوم جديد [7].

### المراحل الأساسية للخوارزمية الجينية:

تبدأ الخوارزمية بتوصيف المشكلة المطروحة من أجل تمثيل الصبغيات Chromosomes الممثلة للحلول بوحدة من طرائق الترميز، بعد ذلك يتم توليد مجموعة عشوائية من الحلول المرشحة candidate solutions، يشار إلى كل من هذه الحلول على أنه كروموسوم chromosome، وإلى المجموعة بأكملها باسم السكان population، مع الإشارة إلى السكان الأوليين بالجيل الأول.

في كل جيل، يتم اختيار أفراد المجتمع ودمجهم في محاولة "لتكاثر" الكروموسومات بدرجة لياقة أعلى؛ تدعى هذه العملية بالعبور. يعتمد اختيارنا لنوع العبور بشكل أساسي على نوع الترميز المستخدم، وقد يكون هذا في بعض الأحيان معقداً للغاية ولكنه عادةً ما يؤدي إلى تحسين أداء الخوارزمية الجينية. ومع ذلك، يبقى السؤال حول ما إذا كان هذا النوع من العبور أفضل من الأنواع الأخرى مفتوحاً.

بعد إنشاء عدد معين من الكروموسومات الجديدة بهذه الطريقة، تحل مجموعة فرعية منها (تلك التي تسجل أعلى القيم لتابع اللياقة) مكان جزء متساوٍ من السكان الحاليين (أولئك الذين يسجلون قيم منخفضة لتابع اللياقة). ثم تتكرر هذه العملية جيلاً بعد جيل، مع بقاء السكان في حجم ثابت، حتى يمر عدد معين من الأجيال، أو يسجل كروموسوم درجات أعلى من القيمة المحددة مسبقاً لتابع اللياقة؛ ثم تقوم الخوارزمية بإرجاع الكروموسوم الأفضل (ذو القيمة الأعلى لتابع اللياقة) باعتباره الحل الأمثل [17].

### 3. مراحل تنفيذ الخوارزمية:

يتمثل الإطار العام للخوارزمية في الخطوات التالية:

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

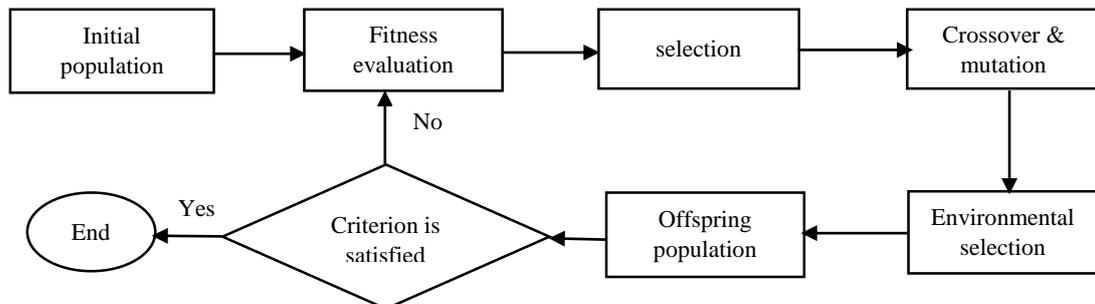
2. تستمر عملية التطور حتى يتم تحقق معيار التوقف المحدد مسبقاً وهو الحد الأقصى لعدد الأجيال في هذا العمل:

- 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 نسل جديد offspring باستخدام العوامل الوراثية التقاطع والطفرة.
- 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.

يلخص الشكل (4) مراحل تنفيذ الخوارزمية وفيما يلي شرح مفصل لكل مرحلة من هذه المراحل:



الشكل (4): مراحل تنفيذ الخوارزمية

### أولاً: استراتيجية ترميز الفرد وتشكيل مجتمع السكان الأولي Initial population:

يمثل الفرد في الخوارزمية المقترحة بنية شبكة CNN مقترحة لحل مشكلة التصنيف، تعتمد 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 (DenseNet connected networks (DenseNets) [8]، 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 التلافيفية 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 مساراً بديلاً لمشتقات تابع الخطأ أثناء الانتشار الخلفي للإشارة (مع الانتشار الخلفي). تستخدم اتصالات التخطي بين الطبقات المختلفة غير المتسلسلة بإحدى الطريقتين التاليتين:

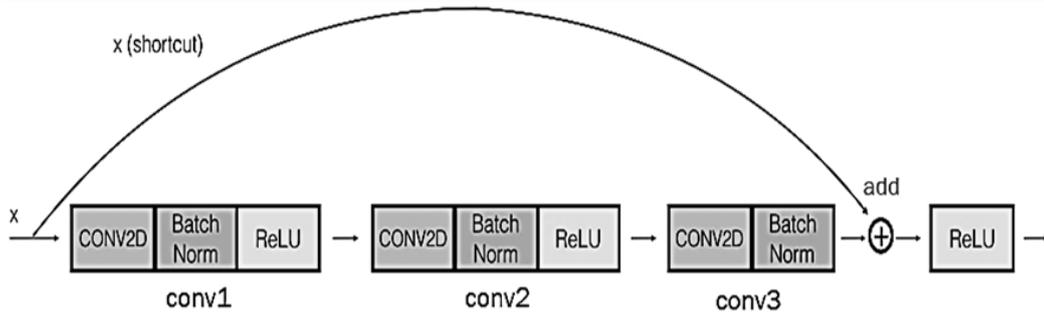
1. الإضافة (Add) كما في الشبكات المتبقية ResNets.
  2. السلسلة (Concatination) كما في الشبكات الكثيفة الاتصال DenseNets.
- يوضح الجدول (1) أربعة أنواع مختلفة من الوحدات ومواقعها في شبكات CNN والمعلومات المشفرة في كل منها، وهي:

الجدول 1: المعلومات المشفرة في بنية الفرد

المعلومات المشفرة في كل منها			نوع الوحدة
عدد قنوات الخرج	عدد قنوات الدخل	عدد الكتل	ResNet Unit (RU)
معدل النمو	عدد قنوات الخرج	عدد الكتل	DenseNet Unit (DU)
		نوع التجميع	Pooling Unit (PU)
		عدد العصبونات	Full-connected Unit (FU)
	نوع الوحدة	عدد الوحدات في كل فرد	على مستوى الفرد

1- وحدة الشبكة المتبقية (ResNet Unit (RU): تتكون كل وحدة من مجموعة من كتل الشبكات المتبقية (ResNet Blocks (RBs بحيث يتم اعتبار عدد هذه الكتل كبارامتر Hyper-parameter يتم تحديد قيمته بشكل عشوائي أثناء تنفيذ الخوارزمية. يُظهر الشكل (5) مثالاً لكتلة شبكة متبقية RB مكونة من تتابع ثلاث طبقات التفاف واتصال تخطي واحد.

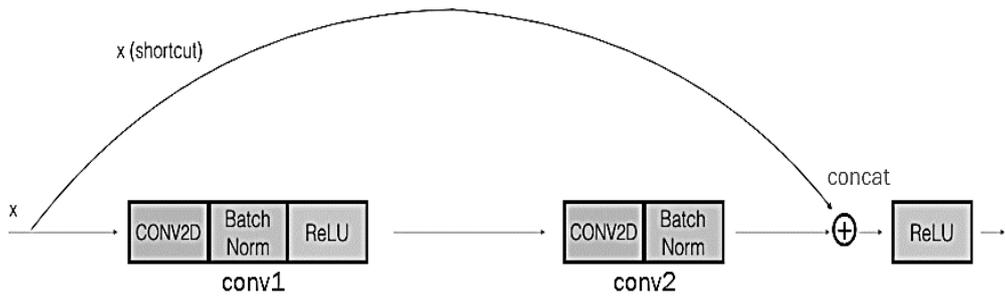
وبالتالي تصبح البارامترات hyper-parameters المشفرة في Unit ResNet هي: عدد الكتل في كل وحدة أي RBs-number، الأحجام المكانية للمدخلات والمخرجات أي عدد قنوات الدخل input\_channels وعدد قنوات الخرج output\_channels.



الشكل (5): بنية ResNet Block

2- وحدة الشبكة الكثيفة الاتصال (DenseNet Unit (DU): تتكون كل وحدة من مجموعة من كتل الشبكات الكثيفة الاتصال (DenseNet Blocks) (DBs) بحيث يتم اعتبار عدد هذه الكتل كبارامتر Hyper-parameter يتم تحديد قيمته بشكل عشوائي أثناء تنفيذ الخوارزمية. يُظهر الشكل (6) مثالاً لكتلة شبكة كثيفة الاتصال DB مكونة من طبقتي التفاف واتصال تخطي واحد.

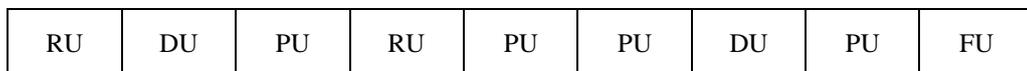
وبالتالي تصبح البارامترات hyper-parameters المشفرة في DenseNet Unit هي: عدد الكتل في كل وحدة DBs-number، الأحجام المكانية للمدخلات والمخرجات أي عدد قنوات الدخل input\_channel وعدد قنوات الخرج output\_channels، بالإضافة إلى معدل النمو growth rate.



الشكل (6): بنية DenseNet Block

3- وحدة التجميع (Pooling Unit (PU): تتكون وحدة التجميع من طبقة تجميع واحدة، يتم تحديد نوع التجميع عن طريق متغير عشوائي max\_or\_avg ضمن المجال [0,1]، فإذا كان  $max\_or\_avg < 0.5$  يكون نوع التجميع هو تجميع المتوسط average-pooling وإلا فهو تجميع الحد الأقصى max-pooling. وبالتالي تصبح البارامترات Hyper-parameters المشفرة في وحدة التجميع هي نوع التجميع فقط Type.

4- وحدة متصلة بالكامل (Full-connected Unit (FU): تتكون هذه الوحدة من طبقة متصلة بالكامل واحدة، وبالتالي تكون البارامترات Hyper-parameters المشفرة في هذه الوحدة هي عدد العصبونات Neurons-number. نظراً لأن أداء شبكات CNN يتأثر إلى حد كبير بعمقها [21]، لذلك يتم استخدام استراتيجية تشفير الفرد متغيرة الطول من أجل الحصول على فرص أفضل للوصول إلى أفضل بنية، وذلك عن طريق اختيار عدد الوحدات ونوعها ضمن كل فرد بشكل عشوائي، وأيضاً يتم تهيئة البارامترات المشفرة لكل وحدة بشكل عشوائي. يوضح الشكل (7) مثالاً لبنية فرد مكون من 9 وحدات.



الشكل (7): بنية فرد مكون من 9 وحدات.

وفق الإطار العام للخوارزمية المقترحة، يتم تهيئة مجتمع السكان الأولي Initial population عشوائياً بحجم N من الأفراد، حيث يتم توليد كل فرد بشكل عشوائي وفق استراتيجية الترميز المقترحة.

#### ثانياً: تقييم لياقة الفرد Fitness evaluation:

يتم تقييم لياقة كل فرد في المجتمع الأولي حيث يتم فك تشفير كل فرد إلى شبكة CNN المقابلة ويتم تحديد عدد معين من الفئات classes بواسطة مجموعة بيانات الصور المحددة، بعد ذلك يتم تدريب CNN بواسطة خوارزمية التدرج المنحدر العشوائي Stochastic Gradient Descent (SGD) على بيانات التدريب training-data باستخدام وحدة معالجة الرسومات GPU، ويتم حساب دقة تصنيف الفرد على بيانات التحقق validation-data باستخدام تابع اللياقة fitness function الذي يعطى بالعلاقة (1) كما يلي:

$$Accuracy = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}} \quad (1)$$

يتم اعتماد عدد محدد من الحقب (التكرارات) لتدريب كل فرد، عند نهاية كل حقة سيتم اختبار الفرد على بيانات التحقق أي تقييم لياقته ومن ثم سيتم الاحتفاظ بالبارامترات التي تعطي أفضل دقة للفرد المعني على بيانات التحقق.

عندما يتم الانتهاء من مرحلة التدريب، يتم تعيين أفضل دقة تصنيف على بيانات التحقق على أنها لياقة الفرد، وأخيراً يتم تعريف هوية الفرد ولياقته ووضعها في مكون ذاكرة التخزين المؤقت cache لتسريع عملية تقييم اللياقة حيث لا يحتاج الفرد الذين ينتقل إلى الجيل التالي إلى تقييم اللياقة مرة أخرى إذا لم يتم تغيير بنيته.

#### ثالثاً: اختيار الآباء Selection:

بعد الانتهاء من تقييم لياقة جميع الأفراد، تبدأ مرحلة توليد النسل الجديد، حيث يتم اختيار الأفراد الآباء من الأفراد الحاليين وذلك باستخدام خوارزمية اختيار الدورات الثنائية Binary Tournament Selection حيث تقوم هذه الخوارزمية على اختيار فردين من السكان عشوائياً، ثم يتم اختيار الفرد الذي يتمتع بمستوى أعلى من اللياقة بينهما كأحد الآباء .

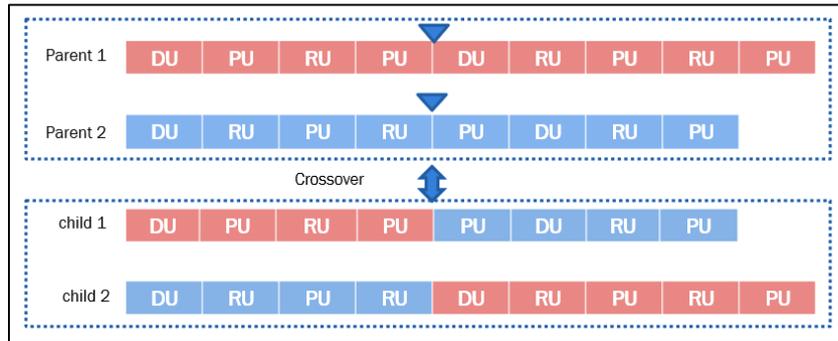
من خلال تكرار هذه العملية يتم اختيار الأفراد الآباء، وهكذا حتى يتم تشكيل تجمع التزاوج mating pool بحيث يتم تعيين حجمه في الخوارزمية المقترحة ليكون مساوياً لحجم مجتمع السكان الأولي N.

#### رابعاً: العبور والطفرة Crossover and Mutation:

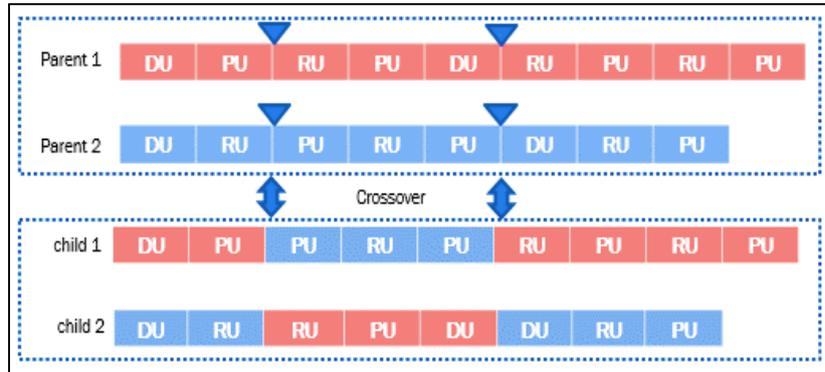
تعد عملية العبور أهم العمليات في الخوارزميات الجينية والتي تحاكي عملية التزاوج البيولوجي بين الأحياء، فالمعتقد السائد هو أن التزاوج بين أفراد يتمتعون بمواصفات جيدة سوف ينتج عنه غالباً أفراد يتمتعون بمواصفات جيدة على أقل تقدير.

في الخوارزمية الجينية بعد اختيار أفضل الأفراد كآباء لتنفيذ العبور، يتم إجراء عملية مزج عشوائي Shuffling للأفراد الآباء الموجودين في تجمع التزاوج للتخلص قدر الإمكان من إجراء العبور بين الفرد ونفسه الناتج عن احتمال تكرار الفرد أكثر من مرة بشكل متتالي بسبب عملية الاختيار التي تسبق العبور، ومن ثم يتم إجراء عملية العبور بين الفرد والذي يليه باستخدام إحدى طرق العبور وبناءً على احتمال حدوث العبور بين الأفراد probability of crossover حيث يتم توليد رقم عشوائي لتحديد ما إذا كان العبور سيتم أم لا. إذا كان الرقم العشوائي المولد أقل من احتمال العبور المحدد مسبقاً، فإنه سيتم إجراء عملية العبور على هذين الفردين وإلا

سيتم وضع هذين الفردين الوالدين في مجتمع النسل الجديد offspring population باعتبارهما النسل الجديد offspring. يوجد عدة أنواع من العبور، في هذا البحث سنقوم باختبار نوعين لمعامل العبور وهما، العبور باستخدام معامل العبور بنقطة واحدة one-point crossover، حيث يتم تحديد نقطة ما عشوائياً على طول كل فرد، والقيام بالقطع عندها ومن ثم إجراء تبديل الجزئين من الأفراد الوالدين اعتباراً من نقطة القطع لإنشاء نسليين، كما هو موضح في الشكل (8). والنوع الآخر هو العبور باستخدام معامل العبور بعدة نقاط Multi-point crossover، حيث يتم تحديد عدة نقاط عشوائياً على طول كل فرد، والقيام بالقطع عندها ومن ثم إجراء تبديل الأجزاء من الأفراد الوالدين اعتباراً من نقطة القطع لإنشاء نسليين، كما هو موضح في الشكل (9).



الشكل (8): العبور بنقطة واحدة one-point crossover



الشكل (9): العبور بعدة نقاط multi-point crossover

يتم بعد ذلك تطبيق طفرة على النسل المؤلّد؛ وهي المرحلة الأخيرة من سلسلة العمليات التطورية التي تسهم في الوصول إلى الحل الأمثل، وهي مستمدة من كون الخوارزميات الجينية فرعاً من تقنيات البحث العشوائي عن الحل الأمثل وهكذا فإن حدوث تغيير مفاجئ وغير متوقع في الجيل (عشوائي) ولو كان باحتمال طفيف يمكن أن يكون له الأثر الإيجابي في الاقتراب من الحل الأمثل بشكل أسرع.

يتم توليد رقم عشوائي لتحديد إذا كان سيتم تطبيق طفرة على الفرد أو لا حيث يتم إجراء الطفرة على الفرد الحالي إذا كان الرقم الذي تم إنشاؤه أقل من احتمال الطفرة المحدد مسبقاً. عند تطبيق طفرة على فرد ما، يتم اختيار الموضع بشكل عشوائي من الفرد الحالي وكذلك يتم اختيار عملية طفرة معينة من قائمة الطفرات المقدمة، ثم يتم تنفيذ الطفرة التي تم اختيارها على الموضع المحدد في الفرد، حيث أن عمليات الطفرة المتاحة المحددة في قائمة الطفرات هي:

1. إضافة وحدة Add Unit.

2. حذف وحدة Delete Unit.

## 3. تعديل وحدة Edit Unit.

إذا كانت الطفرة هي تعديل وحدة حالية، فإن التعديل يعتمد على نوع الوحدة وسيتم تغيير جميع المعلومات المشفرة فيها. يتم تخزين النسل الجديد وإزالة الفردين الوالدين لهذا النسل من تجمع التزاوج، ويتم تكرار الخطوات السابقة من اختيار والدين من تجمع التزاوج وتطبيق العبور والطفرة حتى يصبح تجمع التزاوج فارغ. بعد إجراء عملية العبور والطفرة على الأفراد الآباء يمكن أن ينتج حلول غير صالحة للمشكلة المدروسة (أفراد غير صالحين) نتيجة العشوائية المعتمدة في تطبيق معاملات العبور والطفرة على الأفراد الآباء. بناءً على ذلك يتم التحقق من كل فرد جديد إذا كان يمثل حلاً صالحاً أو لا وتسمى هذه العملية بالتصحيح أو الترميم Repairation، وذلك عن طريق التحقق فيما إذا كان الفرد يحقق مجموعة من القيود التي تجعل منه حلاً صالحاً وهي:

1. لا يمكن أن تكون الوحدة الأولى في الفرد وحدة تجمع PU، يمكن أن تكون فقط إما وحدة كثيفة الاتصال DU أو وحدة متبقية RU.
2. يجب أن تكون أبعاد دخل كل وحدة مطابقة لأبعاد خرج الوحدة السابقة لها.
3. عدد كل نوع من الوحدات لا يجب أن يتجاوز الحد الأقصى المحدد له في كل فرد.

في حال حدوث خلل في أحد القيدين (1، 3) يتم إعادة تطبيق معامل العبور والطفرة على الأبوين حتى الحصول على فرد يحقق هذين القيدين. أما بالنسبة للقيد (2) فيتم مباشرة تعديل أبعاد دخل الوحدة لتصبح مطابقة لأبعاد خرج الوحدة السابقة لها.

## خامساً: الاختيار البيئي Environmental selection:

بعد توليد النسل الجديد يتم تقييم لياقة أفراد النسل الجديد ومن ثم تطبيق الاختيار البيئي على الأفراد الحاليين (الآباء والنسل الجديد) من أجل تشكيل الجيل الجديد، حيث يتم أولاً تطبيق الاختيار وفق مبدأ المنافسة الثنائية لاختيار N من الأفراد ذوي اللياقة الأعلى ونقلهم إلى المجموعة التالية من السكان (الجيل التالي). ثانياً، يتم اختيار أفضل فرد من الأفراد الحاليين (الآباء والنسل الجديد) للتحقق مما إذا كان قد تم وضعه في المجموعة التالية من السكان أي الجيل الجديد.

يتم تكرار العمليات السابقة على الجيل الجديد من الأفراد لتوليد الأجيال التالية وتستمر عملية التطور هذه حتى يصل رقم الجيل إلى الحد الأقصى المحدد.

بعد انتهاء عملية التطور يتم اختيار أفضل فرد أي الفرد ذو اللياقة الأعلى، وفك تشفيره إلى بنية شبكة CNN المقابلة استناداً إلى البنية المشفرة ومعلومات تهيئة وزن الاتصال، ومن ثم سيتم اختبار شبكة CNN هذه على مجموعة الاختبار test-dataset.

## النتائج والمناقشة

من أجل تقييم أداء الخوارزمية تم تنفيذ مراحل الخوارزمية الموضحة في الشكل (2) باستخدام لغة python وذلك بالاعتماد على برنامج PyCharm وهو برنامج تطويري يدعم لغة python، ومن ثم تم اختبار الخوارزمية على قواعد البيانات المرجعية Cifar10 و Cifar100 (حيث تتصف كلتا مجموعتي البيانات بالتحدي من حيث أحجام الصور وفئات التصنيف والضوضاء وكذلك التدوير في كل صورة، أيضاً تستخدم هذه المجموعات على نطاق واسع لقياس أداء خوارزميات التعلم العميق فقد أبلغت معظم الخوارزميات المقارنة المختارة عن دقة التصنيف الخاصة بها عليهم) باستخدام وحدة المعالجة الرسومية Nvidia GeForce GTX 1060.

تم التحقق من أداء الخوارزمية المقترحة 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 (2) والجدول (3). نظراً لصعوبة تحليل التعقيد الحسابي لكل خوارزمية مختارة للمقارنة من الناحية النظرية، يتم استخدام عدد الأيام التي استغرقتها عملية التدريب باستخدام GPU كمؤشر على التعقيد الحسابي حيث يتم حساب عدد أيام GPU عن طريق ضرب عدد بطاقات GPU المستخدمة بعدد الأيام التي استغرقتها الخوارزمية للعثور على أفضل بنية، مع الإشارة إلى أن أحدث شبكات CNN المصممة يدوياً لا تحتوي على البيانات المتعلقة بعدد أيام GPU.

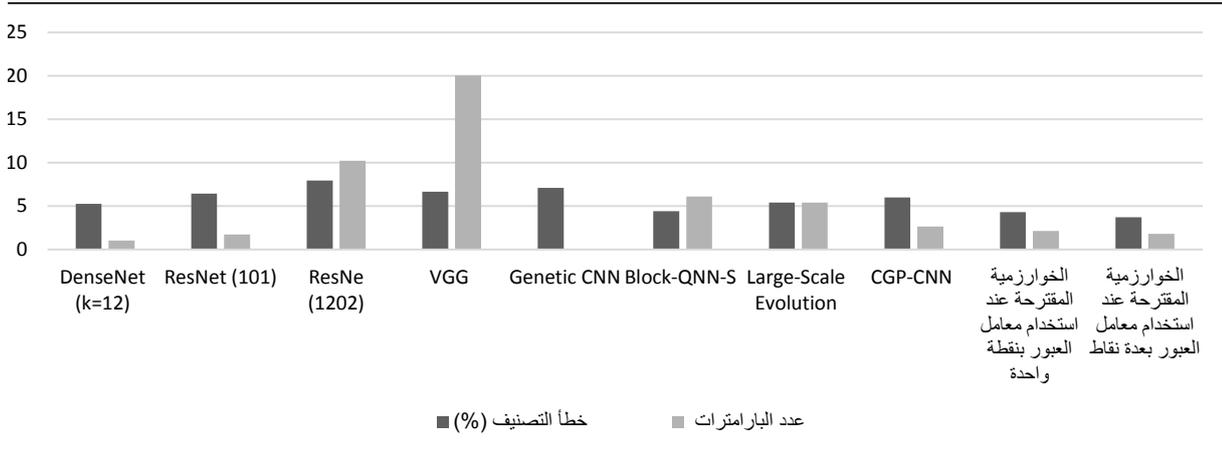
يوضح الجدول (2) النتائج التجريبية للخوارزمية المقترحة والخوارزميات المختارة حيث يشير العمود الأول إلى الخوارزميات المختارة للمقارنة، العمود الثاني يشير إلى خطأ التصنيف على مجموعة البيانات CIFAR10، العمود الثالث يشير إلى عدد البارامترات والعمود الرابع يشير إلى المدة المستغرقة في عملية التدريب للوصول إلى أفضل بنية. بالإضافة إلى ذلك، يشير الرمز "-" في الجدول إلى أنه لم يتم الإبلاغ عن نتيجة علنية من قبل الخوارزمية المقابلة. تُظهر النتائج أن الخوارزمية المقترحة (عند استخدام معامل العبور بنقطة واحدة وكذلك عند استخدام معامل العبور بعدة نقاط) تتفوق على جميع شبكات CNN الحديثة المصنوعة يدوياً (VGG، ResNet، DenseNet) وجميع المنافسين الأقران من الفئة شبه التلقائية (Genetic CNN، Block-QNN-S) والتلقائية (Large-Scale Evolution، CGP-CNN) من حيث معدل خطأ التصنيف على مجموعة البيانات CIFAR10.

كما توضح النتائج أن عدد بارامترات أفضل بنية تم التوصل إليها باستخدام الخوارزمية المقترحة (عند استخدام معامل العبور بنقطة واحدة وكذلك عند استخدام معامل العبور بعدة نقاط) أقل من عدد البارامترات في الخوارزميات التلقائية وشبه التلقائية التي تمت المقارنة معها (Large-Scale Evolution، Block-QNN-S، CGP-CNN) وذلك خلال زمن تنفيذ للخوارزمية المقترحة قدره 40 يوم وفقاً للموارد الحاسوبية المتاحة لدينا.

الجدول (2): مقارنة بين الخوارزمية المقترحة ومجموعة من الخوارزميات على CIFAR10.

	خطأ التصنيف على CIFAR10	عدد البارامترات ( $M = 10^6$ )	زمن تنفيذ الخوارزمية (يوم)
DenseNet (k=12) [8]	5.24	1.0 M	-
ResNet (depth=101) [6]	6.43	1.7 M	-
ResNet (depth=1202) [6]	7.93	10.2 M	-
VGG [21]	6.66	20.04 M	-
Genetic CNN [23]	7.1	-	17
Block-QNN-S [24]	4.38	6.1 M	90
Large-Scale Evolution [20]	5.4	5.4 M	-
CGP-CNN [22]	5.98	2.64 M	27
الخوارزمية المقترحة عند استخدام معامل العبور بنقطة واحدة	4.3	2.1 M	40
الخوارزمية المقترحة عند استخدام معامل العبور بعدة نقاط	3.7	1.8	56

يظهر الشكل (10) المقارنة بين 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.



الشكل (10): مقارنة بين الخوارزمية المقترحة ومجموعة الخوارزميات على Cifar10

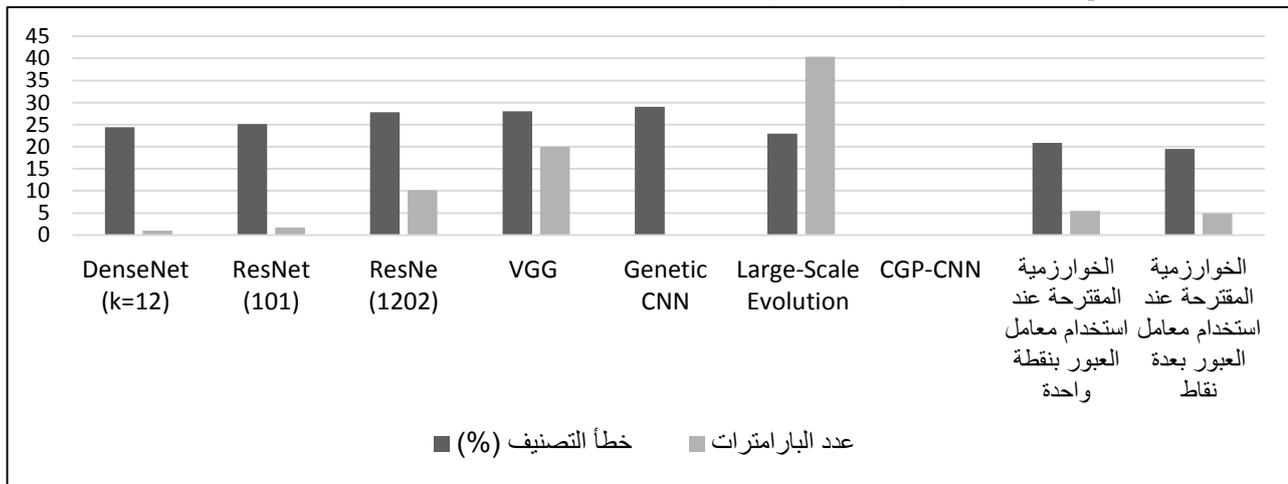
بنفس الطريقة يوضح الجدول (3) النتائج التجريبية للخوارزمية المقترحة والخوارزميات المختارة بالنسبة لقاعدة البيانات CIFAR100 حيث تُظهر النتائج أن الخوارزمية المقترحة (عند استخدام معامل العبور بنقطة واحدة وكذلك عند استخدام معامل العبور بعدة نقاط) تتفوق على جميع شبكات CNN الحديثة المصممة يدوياً (DenseNet، 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 (عند استخدام معامل العبور بنقطة واحدة وكذلك عند استخدام معامل العبور بعدة نقاط) أقل من عدد البارامترات في الخوارزميات التلقائية التي تمت المقارنة معها (Block-QNN-S، Large-Scale Evolution) وذلك خلال زمن تنفيذ للخوارزمية المقترحة قدره 84 يوم وفقاً للموارد الحاسوبية المتاحة لدينا.

الجدول (3): مقارنة بين الخوارزمية المقترحة ومجموعة الخوارزميات على CIFAR100.

الخوارزمية المقترحة عند استخدام معامل العبور بنقطة واحدة	خطأ التصنيف على CIFAR100	عدد البارامترات (M = ) (106)	زمن تنفيذ الخوارزمية (يوم)
DenseNet (k=12) [8]	24.42	1.0 M	-
ResNet (depth=101) [6]	25.16	1.7 M	-
ResNet (depth=1202) [6]	27.82	10.2 M	-
VGG [21]	28.05	20.04 M	-
Genetic CNN [23]	29.05	-	17
Block-QNN-S [24]	20.56	6.1 M	90
Large-Scale Evolution [20]	23	40.4 M	-
CGP-CNN [22]	-	-	-
الخوارزمية المقترحة عند استخدام معامل العبور بنقطة واحدة	20.85	5.5 M	84
الخوارزمية المقترحة عند استخدام معامل العبور بعدة نقاط	19.52	4.9	96

يوضح الشكل (11) 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.

أظهرت النتائج على كلا مجموعتي البيانات Cifar10 و Cifar100 (كما هو موضح في الشكلين 8 و 9)، أن دقة الخوارزمية تزداد بشكل ملحوظ عند استخدام معامل العبور بعدة نقاط بدلاً من معامل العبور بنقطة واحدة (أي خطأ التصنيف ينخفض) ويتم التوصل إلى أفراد بعدد بارامترات أقل، يعود ذلك إلى أن فضاء الحلول يصبح أوسع وبالتالي يتم اختبار عدد أكبر من الحلول الممكنة، ولكن بالمقابل يزداد زمن تنفيذ الخوارزمية بسبب زيادة احتمال الحصول على أفراد غير صالحين وبالتالي يزداد الزمن اللازم لإعادة ترميم هؤلاء الافراد والحصول على أفراد تمثل حلولاً صالحة للخوارزمية.



الشكل (11): مقارنة بين الخوارزمية المقترحة ومجموعة الخوارزميات بعد ضغط بنية الشبكة على Cifar100

## الاستنتاجات والتوصيات

– الهدف من هذه الورقة هو تطوير خوارزمية تصميم بنية CNN باستخدام GAs، القادرة على تصميم أفضل بنية CNN للمهمة المعينة بطريقة تلقائية تمامًا واستنادًا إلى الموارد الحسابية المحدودة واختبار تأثير نوع معامل العبور على أداء الخوارزمية. تم تحقيق هذا الهدف من خلال إستراتيجية التشفير المبنية على كتل شبكات متطورة مع تمثيل متغير الطول للأفراد.

– تم فحص الخوارزمية على مجموعات بيانات تصنيف الصور المعيارية Cifar10 و Cifar100، مقابل ثلاث شبكات CNN حديثة مصممة يدويًا، واثنين من المنافسين قاموا بتصميم بنى CNN بطريقة شبه تلقائية، وآخرين صمموا بنى CNN بطريقة تلقائية تمامًا.

– تم في هذا البحث اختبار تأثير نوعين من معاملات العبور على الخوارزمية المقترحة وهي معامل العبور بنقطة واحدة ومعامل العبور بعدة نقاط.

– تُظهر النتائج أن الخوارزمية المقترحة (عند استخدام معامل العبور بنقطة واحدة أو بعدة نقاط) تتفوق على جميع شبكات CNN الحديثة المصنوعة يدويًا وجميع المنافسين الأقران من الفئة التلقائية من حيث معدل خطأ التصنيف كما تُظهر الخوارزمية المقترحة أداءً تنافسيًا مقارنةً بالمنافسين من الفئة شبه التلقائية.

– بعد دراسة وتطبيق الخوارزمية ومناقشة النتائج التي تم التوصل إليها وجدنا أنه استخدام معامل العبور بعدة نقاط بدلاً من معامل التقاطع بنقطة واحدة أدى إلى:

- انخفاض عدد البارامترات.
  - زيادة زمن تنفيذ الخوارزمية (زمن الوصول إلى أفضل بنية شبكة) لأن معامل العبور بعدة نقاط سيزيد من احتمالية توليد أفراد غير صالحين وبالتالي عملية الترميم المستخدمة (تصحيح الأفراد غير الصالحين) في الخوارزمية سوف تستهلك زمن أكبر حتى الوصول إلى فرد صالح.
  - ولكن بالمقابل أدى إلى زيادة دقة الخوارزمية (انخفاض خطأ التصنيف) بحيث تبقى الدقة التي تم التوصل إليها 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. كما أن عملية الترميم المستخدمة (تصحيح الأفراد غير الصالحين) في الخوارزمية تستهلك بعض الوقت لأنه لا يوجد حد لإيقافها حيث تستمر بإعادة الطفرة والتقاطع حتى الوصول إلى فرد صالح، لذلك سيكون من الجيد لو تم تطوير طريقة أكثر فعالية لترميم الأفراد الغير صالحين دون تكرار عمليات التقاطع والطفرة.

## المراجع

- [1] Bergstra, J., & Bengio, Y. (2012). Random search for hyper-parameter optimization. *The Journal of Machine Learning Research*, 13(1), 281-305.
- [2] 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 on Assembly and Task Planning, 2003*. (pp. 109-114). IEEE.
- [3] 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.
- [4] Dayoub, Y. (2015). Stochastic logicallinguistic approachMulti-level automated object's dialogue control (MADC). *سلسلة العلوم الهندسية* [ISSN: 2079-3081], 35(6).
- [5] Dérnoncourt, F., & Lee, J. Y. (2016, December). Optimizing neural network hyperparameters with gaussian processes for dialog act classification. In *2016 IEEE Spoken Language Technology Workshop (SLT)* (pp. 406-413). IEEE.
- [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] Zhang, C., & Lu, Y. (2021). Study on artificial intelligence: The state of the art and future prospects. *Journal of Industrial Information Integration*, 23, 100224.
- [8] 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).
- [9] Khan, A., Sohail, A., Zahoor, U., & Qureshi, A. S. (2020). A survey of the recent architectures of deep convolutional neural networks. *Artificial Intelligence Review*, 53(8), 5455-5516.
- [10] Krizhevsky, A., & Hinton, G. (2009). Learning multiple layers of features from tiny images.
- [11] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6), 84-90.
- [12] Lee, C. Y., Gallagher, P. W., & Tu, Z. (2016, May). Generalizing pooling functions in convolutional neural networks: Mixed, gated, and tree. In *Artificial intelligence and statistics* (pp. 464-472).
- [13] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *nature*, 521(7553), 436-444.
- [14] 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.
- [15] Lin, M., Chen, Q., & Yan, S. (2013). Network in network. *arXiv preprint arXiv:1312.4400*.
- [16] Mirjalili, S. (2019). Genetic algorithm. In *Evolutionary algorithms and neural networks* (pp. 43-55). Springer, Cham.
- [17] Nwankpa, C., Ijomah, W., Gachagan, A., & Marshall, S. (2018). Activation functions: Comparison of trends in practice and research for deep learning. *arXiv preprint arXiv:1811.03378*.
- [18] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Vanderplas, J. (2011). Scikit-learn: Machine learning in Python. *the Journal of machine Learning research*, 12, 2825-2830.
- [19] 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*.

- [20] Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.
- [21] 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).
- [22] Xie, L., & Yuille, A. (2017). Genetic cnn. In *Proceedings of the IEEE international conference on computer vision* (pp. 1379-1388).
- [23] Zhong, Z., Yan, J., Wu, W., Shao, J., & Liu, C. L. (2018). Practical block-wise neural network architecture generation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2423-2432).