



كلية الهندسة - جامعة بغداد

جمعية كليات الهندسة
اعضاء اتحاد الجامعات العربية

تحسين أداء الشبكات العصبية التلافيفية باستخدام الحوسبة التطورية

كلودا كمال ابراهيم^{1*}، يعرب شحادة ديوب²، لينا رياض خليل مقدسيان³

¹ قسم تكنولوجيا المعلومات، كلية هندسة تكنولوجيا المعلومات والاتصالات، جامعة طرطوس، سوريا، klodaibrahim@gmail.com

² قسم تكنولوجيا المعلومات، كلية هندسة تكنولوجيا المعلومات والاتصالات، جامعة طرطوس، سوريا، yaroubdayoub@gmail.com

³ قسم العلوم الأساسية، كلية هندسة تكنولوجيا المعلومات والاتصالات، جامعة طرطوس، سوريا، lina_makdessian@yahoo.fr

* الباحث الممثل: كلودا ابراهيم، klodaibrahim@gmail.com

نشر في: 30 ايلول 2021

الخلاصة- تم في هذا البحث اقتراح خوارزمية لتصميم بنى الشبكات العصبية التلافيفية (CNN) Convolutional Neural Networks باستخدام الخوارزميات الجينية (GAs) Genetic Algorithms القادرة على تعلّم أفضل بنية لشبكة CNN بطريقة تلقائية تماماً استناداً إلى الموارد الحاسوبية المحدودة، حيث تم اقتراح استراتيجية ترميز تعتمد على كتل شبكات حديثة متطورة مصممة يدوياً، بحيث لا تتطلب الخوارزمية المقترحة وجود مستخدمين لديهم معرفة مسبقة لشبكات CNN أو المشكلة التي تتم معالجتها أو حتى الخوارزميات الجينية. تم تقييم أداء الخوارزمية المقترحة 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. أظهرت النتائج التجريبية أنه يمكن استخدام الخوارزمية المقترحة للعثور تلقائياً على بنية CNN التنافسية مقارنةً بالنماذج الحديثة، حيث حققت هذه الخوارزمية أفضل دقة تصنيف بين CNNs المصممة يدوياً وبشكل تلقائي بالإضافة إلى دقة تصنيف منافسة للخوارزميات شبه التلقائية semi-automatic.

الكلمات الرئيسية - الشبكات العصبونية التلافيفية، CNN، الحوسبة التطورية، التعلم العميق، تصنيف الصور

1. المقدمة

يتم تحديد بنية هذه الشبكات من خلال عدة بارامترات تدعى hyper-parameters مثل عدد الطبقات ونوعها وعدد وحجم النوى ونوع وظيفة التفعيل، حيث أن الطرق التقليدية المستخدمة لتحديد قيم هذه البارامترات هي: البحث الشبكي Grid Search [19]، البحث اليدوي Manual Search [5] والبحث العشوائي Random Search [1]، 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 بشكل متزايد لحل مشاكل الأمثلة [4] [3] [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].

1.1 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 البريديّة والأرقام وما إلى ذلك [14].

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 أنّ 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 CNN 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 ستنتج مستوى أسوأ من الدقة، لذلك يعد تحديد بنية CNN أمراً بالغ الأهمية [21].

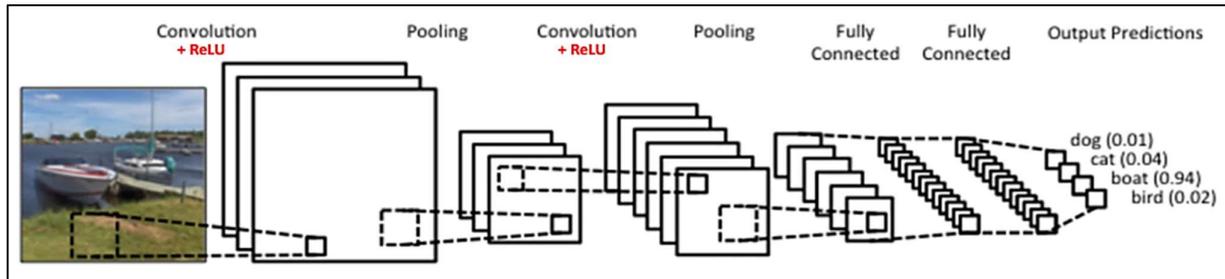
1- الطبقة التلافيفية Convolutional layer: هي العمود الفقري لـ CNN وتأتي تسميتها من عملية الطي أو الالتفاف الرياضية، حيث يكون ناتج هذه العملية هو خريطة السمات feature map التي تعكس استجابة المرشحات لنمط معين في الصورة من خلال أوزان كل مرشح ويتم تحديد أوزان المرشح أثناء عملية تدريب الشبكة [14]. تتكون خريطة السمات من عدة قنوات، ترتبط أبعاد هذه القنوات بأبعاد مصفوفة الإدخال وأبعاد المرشح بالإضافة إلى العاملين التاليين:

- **الخطوة Stride:** 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 الترشح ولا يتم فقد أي معلومات على نهايات المصفوفة.

1.1.1 الهيكل الأساسي لـ CNN

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 [9] كما هو موضح بالشكل (1)، وهي:



الشكل 1: بنية الشبكة العصبونية التلافيفية CNN.

اكتسابه عن طريق الوراثة بشكل كبير، حيث تعتمد على محاكاة عملية التزاوج بين الكائنات الحية من النوع نفسه من أجل الوصول إلى الحل الأنسب للمشكلة المطروحة، وذلك انطلاقاً من مبدأ داروين في الاصطفاء الطبيعي القائم على الاحتفاظ بالميزات والصفات الجيدة الموجودة في جيل الأبناء لنقلها إلى جيل الأبناء بهدف الحصول على ذرية قوية تتمتع بأفضل صفات جيل السلف على أقل تقدير [7].

1.2.1 المراحل الأساسية للخوارزمية الجينية

تبدأ الخوارزمية بتوصيف المشكلة المطروحة من أجل تمثيل الصيغيات 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 crossover، وتختلف طريقة تنفيذها اعتماداً على كيفية ترميز الكروموسوم للحل المرشح وعلى تطبيق الخوارزمية.

بعد إنشاء عدد معين من الكروموسومات الجديدة بهذه الطريقة، تحل مجموعة فرعية منها (عادةً تلك التي تسجل أعلى القيم لتابع اللياقة) مكان جزء متساوٍ من السكان الحاليين (غالباً 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].

2- طبقة التنغيع Activation Layer: بعد انتهاء عملية الالتفاف يتم إدخال خريطة السمات إلى طبقة التنغيع حيث يُطبَّق تابع التنغيع على كل عصبون أي ما يكافئ عنصر من خريطة السمات، ويُؤخذ بعين الاعتبار الحد من خرج العصبون وكذلك اللاخطية في عملية التنغيع حيث أن عمليات الالتفاف التي أجريت قبلها هي عمليات خطية. أهم توابع التنغيع المستخدمة في هذا النوع من الشبكات هو تابع الوحدة الخطية المصححة ReLU الذي أثبت فاعليته مقارنة بالتوابع الأخرى [18].

3- طبقة التجميع Pooling layer: بعد تطبيق تابع التنغيع على خريطة السمات، يتم العمل على تقليل أبعادها بطريقة تحافظ على المعلومات من خلال عملية التجميع، ويتم ذلك بعدة طرق أهمها التجميع بالحد الأقصى حيث يتم مقارنة 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.

4- طبقة الاتصال الكامل Fully Connected layer: ليس بالضرورة أن تكون خريطة السمات مفهومة من قبل الإنسان، ولكن بالنسبة للشبكة تكون عبارة عن شيفرة تخص صف معين (ويجري تحديد هذا الصف ضمن هذه الطبقة). بعد استخراج السمات، يتم استخدام مصنف لتصنيف تلك السمات وذلك باستخدام شبكات عصبونية أمامية التغذية، يكون دخلها شعاع مكون من خريطة السمات بعد إجراء مرحلة التجميع، وخرجها عبارة عن شعاع يعبر عن الصف الذي تنتمي إليه خريطة السمات [16].

1.2 الخوارزميات الجينية Genetic Algorithms

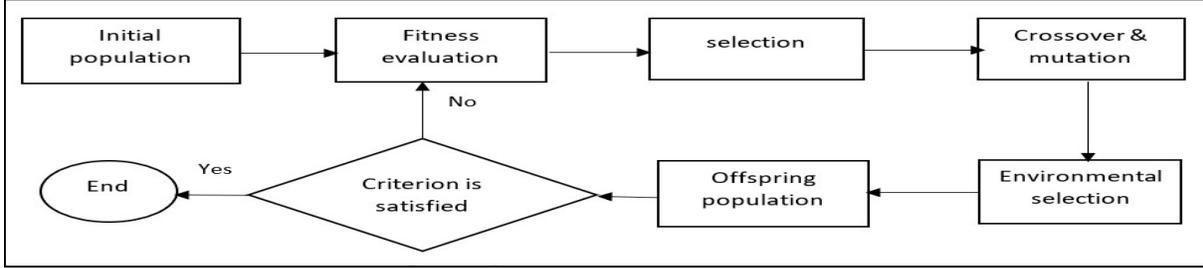
تعد الخوارزميات الجينية من التقنيات المهمة في البحث العشوائي عن الحل الأمثل، وهي تمثيل للاعتقاد السائد بأن الذكاء البشري يخلق مع الإنسان ويتم

2. أهداف البحث

3. منهجية البحث

تم تصميم مراحل الخوارزمية المقترحة 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 باستخدام وحدة المعالجة الرسومية Nvidia GeForce GTX 1060.

يهدف هذا البحث إلى اقتراح خوارزمية لاكتشاف أفضل بنية لشبكات CNN باستخدام الخوارزميات الجينية (GAs) وفقاً للمهمة المعنية بطريقة تلقائية تماماً لا تتطلب أي تدخل يدوي أثناء البحث التطوري استناداً إلى الموارد الحسابية المحدودة، حيث يمكن استخدام CNNs التي تم الحصول عليها بعد انتهاء عملية التطور مباشرة في معالجة البيانات، دون الحاجة إلى مزيد من التنقيح مثل إضافة المزيد من الطبقات التلافيفية أو التجميعية. بالإضافة إلى ذلك، يمكن استخدام الخوارزمية المقترحة مباشرة من قبل باحثين آخرين لا يحتاجون إلى القيام بأي تجهيز مسبق مثل توفير شبكة مضبوطة يدوياً مسبقاً.



الشكل 2: الإطار العام للخوارزمية المقترحة.

3.1 مراحل تنفيذ الخوارزمية المقترحة

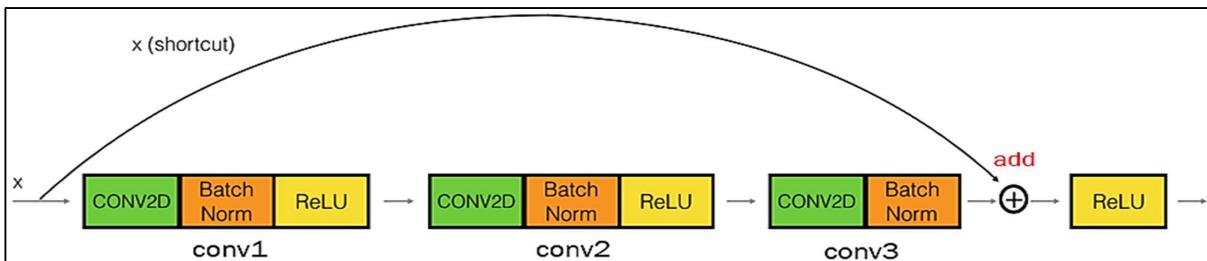
3.1.1 استراتيجية ترميز الفرد

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 تحديده قيمته بشكل عشوائي أثناء تنفيذ الخوارزمية. يُظهر الشكل (3) 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 ، لتقليل التعقيد الحسابي في طبقة الالتفاف التالية conv2. في طبقة الالتفاف الثانية conv2، يتم استخدام المرشحات ذات الحجم الأكبر، مثل 3×3 ، لتعلم الميزات ذات الحجم المكاني نفسه. في conv3، يتم استخدام مرشحات بحجم 1×1 مرة أخرى، ويتم زيادة الحجم المكاني لتوليد المزيد من الميزات. يضاف الدخل مشار إليه بواسطة \oplus إلى ناتج conv3 كخرج نهائي. مع ملاحظة أنه إذا كانت الأحجام المكانية للدخل ومخرجات conv3 غير متساوية، يتم تطبيق مجموعة من العمليات التلافيفية مع مرشحات بحجم 1×1 على الدخل، لتحقيق نفس الحجم المكاني لخرج conv3 من أجل عملية الإضافة. بعد كل عملية التفاف يتم تطبيق تسوية دفعية batch normalization ودالة التفعيل relu. وبالتالي تصبح البارامترات hyper-parameters المشفرة في ResNet Unit هي: عدد الكتل في كل وحدة أي RBs-number، الأحجام المكانية للمدخلات والمخرجات أي عدد قنوات الدخل input_channels وعدد قنوات الخرج output_channels.

يمثل الفرد في الخوارزمية المقترحة بنية شبكة CNN مقترحة لحل مشكلة التصنيف، تعتمد استراتيجية التشفير المقترحة للفرد على أربعة أنواع مختلفة من الوحدات ومواقعها في شبكات CNN والمعلومات المشفرة في كل منها كما هو موضح في الجدول (1)، وهي:

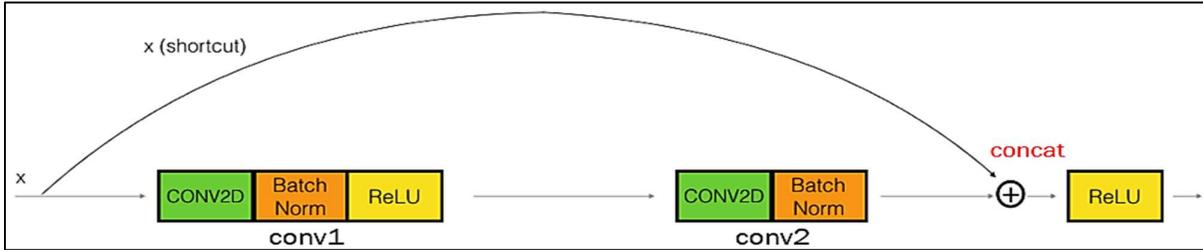
الجدول 1: المعلومات المشفرة في بنية الفرد.

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



الشكل 3: بنية ResNet Block.

جميع الطبقات التلافيفية السابقة. بالإضافة إلى ذلك، هناك بارامتر معدل النمو k للتحكم في الحجم المكاني للمدخلات والمخرجات للطبقة التلافيفية. إذا كان الحجم المكاني للمدخلات a ، فإن الحجم المكاني للمخرج هو $a+k$ ، والذي يتحقق من خلال العملية التلافيفية باستخدام k من المرشحات. 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 المخرج $output_channels$ ، بالإضافة إلى معدل النمو $growth\ rate$.



الشكل 4: بنية DenseNet Block.

الرسميات GPU، ويتم حساب دقة تصنيف الفرد على بيانات التحقق validation-data باستخدام تابع اللياقة fitness function الذي يعطي بالعلاقة (1) كما يلي:

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

يتم اعتماد عدد محدد من الحقب لتدريب كل فرد، عند نهاية كل حقبة 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.

3.1.3 اختيار الآباء

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 أفراد 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 .

3.1.4 العبور والطفرة

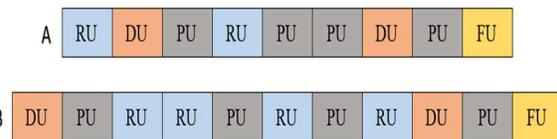
من أجل توليد النسل الجديد، يتم اختيار اثنين من الأفراد الآباء بشكل عشوائي من تجمع التزاوج ثم يتم توليد رقم عشوائي لتحديد ما إذا كان العبور سيتم أم لا. إذا كان الرقم العشوائي المولد أقل من احتمال العبور المحدد مسبقاً، فإنه سيتم إجراء عملية العبور على هذين الفردين وإلا سيتم وضع هذين الفردين الوالدين في مجتمع النسل الجديد offspring population باعتبارهما النسل الجديد offspring.

2- وحدة الشبكة الكثيفة الاتصال (DenseNet Unit (DU): تتكون كل وحدة من مجموعة من كتل الشبكات الكثيفة الاتصال (DenseNet Blocks) بحيث يتم اعتبار عدد هذه الكتل كبارامتر Hyper-parameter يتم تحديده قيمته بشكل عشوائي أثناء تنفيذ الخوارزمية. يُظهر الشكل (4) مثالاً لكتلة شبكة متبقية DB مكونة من طبقتي التفاضل 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 DB، تتلقى كل طبقة التفاضلية مدخلات ليس فقط من بيانات المدخلات ولكن أيضاً من خرج

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 $max_or_avg < 0.5$ فإن التجميع 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 (A-5) مثالاً لبنية فرد مكون من 9 وحدات، والشكل (B-5) مثالاً لبنية فرد مكون من 11 وحدة.



الشكل (5): A- بنية فرد مكون من 9 وحدة.

B- بنية فرد مكون من 11 وحدة.

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.

3.1.2 تقييم لياقة الفرد

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

3.2.1 مجموعات البيانات المستخدمة لقياس أداء الخوارزمية المقترحة

تم اختيار مجموعات البيانات المرجعية CIFAR10 و CIFAR100 [10] من أجل مهام تصنيف الصور في التجارب، حيث تتصف كلتا مجموعتي البيانات بالتحدي من حيث أحجام الصور وفئات التصنيف والوضاء وكذلك التدوير في كل صورة، أيضاً تستخدم هذه المجموعات على نطاق واسع لقياس أداء خوارزميات التعلم العميق فقد أبلغت معظم الخوارزميات المقارنة المختارة علانية عن دقة التصنيف الخاصة بها عليهم.

تحتوي مجموعة البيانات CIFAR10 على 60000 صورة ملونة RGB، حجم كل صورة $32 \times 32 \times 3$ ، مقسمة إلى 10 فئات لكل منها 6000 صورة. يتم استخدام 50000 صورة منها لتدريب الشبكة training-data، بينما يتم استخدام 10000 صورة أخرى لاختبار الشبكة test-data.

مجموعة بيانات CIFAR100 مماثلة لـ CIFAR10، باستثناء أن لديها 100 فئة. بالإضافة إلى ذلك، تشغل الكائنات التي يتم تصنيفها في مجموعات البيانات المرجعية هذه عادةً مساحات مختلفة من الصورة بأكملها، ومواقعها ليست هي نفسها في الصور المختلفة كما أن اختلافاتها يمثل تحدياً لخوارزميات التصنيف.

3.2.2 إعدادات البارامترات في الخوارزمية المقترحة

جميع البارامترات يتم تعيينها استناداً إلى قيمها الشائعة الاستخدام، لتقليل الصعوبة التي يواجهها الباحثون الذين يرغبون في استخدام الخوارزمية المقترحة في العثور على أفضل بنى CNN لبياناتهم المستقصاة حتى لو لم يكن لديهم خبرة في الخوارزميات الجينية GAS. على وجه الخصوص، تم تعيين حجم السكان والعدد الأجيال ليكون 20، واحتمالات العبور والطفرة على 0.9 و 0.2، على التوالي.

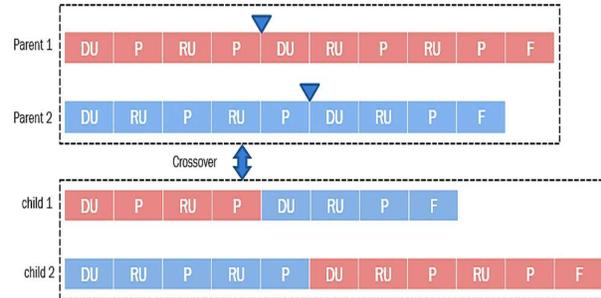
في تقييم لياقة الفرد، يتم تدريب كل فرد بواسطة خوارزمية التدرج المنحدر العشوائي (SGD) مع حجم الدفعة 128 وقيمة زخم 0.9. تتم تهيئة معدل التعلم إلى 0.01 وتم ضبط انحلال الوزن على 5×10^{-4} ، بالإضافة إلى ذلك، يتم تعيين لياقة الفرد إلى الصفر في حال نفاذ الذاكرة أثناء التدريب. عندما تنتهي العملية التطورية، يتم تدريب أفضل فرد على بيانات التدريب الأصلية مع إعدادات SGD نفسها، ويتم الإبلاغ عن معدل الخطأ في بيانات الاختبار للمقارنة.

بالإضافة إلى ذلك، فإن الخيارات المتاحة لمعدل النمو k في DB هي 12 و 20 و 40 بناءً على تصميم DenseNet، ويتم تحديد الحد الأقصى لعدد الطبقات التلافيفية في DB على أنها 10 (عندما تكون $k = 12$ و $k = 20$) و 6 (عندما $k = 40$). يتم تعيين كل من الحد الأقصى لعدد وحدات RU ووحدات DU في شبكة CNN على 4، ويتم تعيين كل من أعداد DBs و RBs في DU و RU، على التوالي، من 3 إلى 8 مع ملاحظة أن هذه الإعدادات تستند بشكل أساسي إلى مواردنا الحاسوبية المتاحة لأن أي عدد خارج هذه الإعدادات سيخرج بسهولة من الذاكرة (إذا كانت المنصة الحاسوبية للمستخدم مزودة بوحدات معالجة رسومات أكثر قوة، فيمكن تعيين عدد الوحدات ضمن الفرد وعدد الكتل على قيم عشوائية دون قيود). يتم تنفيذ الخوارزمية المقترحة للتجربة على وحدة المعالجة الرسومية GPU ذات الطراز Nvidia GeForce GTX 1060.

3.2.3 النتائج والمناقشة

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

يتم إجراء 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 الجزأين من الأفراد الوالدين اعتباراً من نقطة القطع لإنشاء نسليين، كما هو موضح في الشكل (6).



الشكل (6): العبور بنقطة واحدة one-point crossover.

يتم بعد ذلك تطبيق طفرة على النسل المُولّد، حيث 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 التقاطع والطفرة حتى يصبح تجمع التزاوج فارغ.

3.1.5 الاختيار البيئي

بعد توليد النسل الجديد يتم تقييم لياقة أفراد النسل الجديد، ومن ثم يتم تطبيق الاختيار البيئي على الأفراد الحاليين (الأباء والنسل الجديد) من أجل تشكيل الجيل الجديد، حيث يتم أولاً تطبيق اختيار الدورات الثنائي binary tournament selection عليهم لاختيار N من الأفراد ذوي اللياقة الأعلى ثم يتم وضع هؤلاء الأفراد المختارين في المجموعة التالية من السكان. ثانياً، يتم اختيار أفضل فرد من الأفراد الحاليين (الأباء 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.

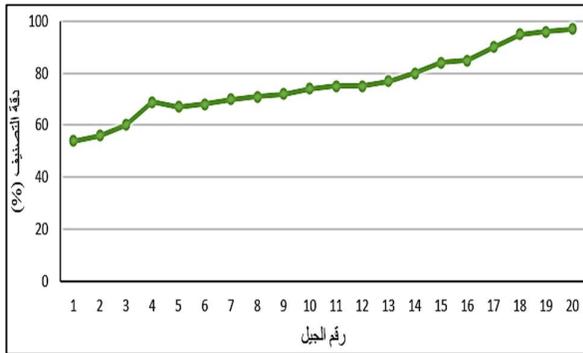
بعد انتهاء عملية التطور يتم اختيار "أفضل فرد" أي الفرد ذو اللياقة الأعلى، ويتم فك تشفيره إلى CNN المقابلة استناداً إلى البنية المشفرة ومعلومات تهيئة وزن الاتصال، ومن ثم سيتم اختبار شبكة CNN هذه على مجموعة الاختبار test-dataset.

3.2 الاختبارات والنتائج

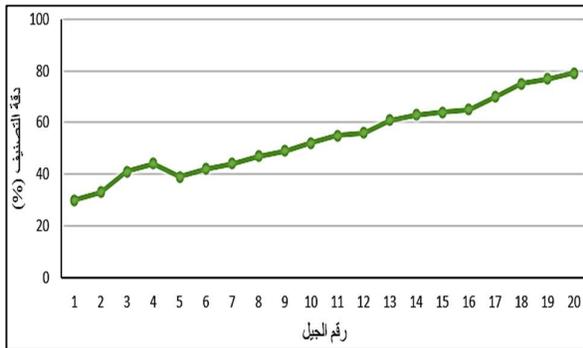
من أجل تقييم أداء الخوارزمية المقترحة، قمنا بإجراء سلسلة من التجارب على مجموعات البيانات المرجعية من أجل مهام تصنيف الصور ومقارنة النتائج مع الخوارزميات الحديثة التي أظهرت أداءً واعداً في هذا المجال.

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	2750
CGP-CNN [22]	-	-	-
Proposed Algorithm	20.85	6.1 M	84

بنفس الطريقة يوضح الجدول (3) النتائج التجريبية للخوارزمية المقترحة والمنافسين المختارين بالنسبة لقاعدة البيانات CIFAR100 حيث تُظهر النتائج أن الخوارزمية المقترحة تتفوق على جميع شبكات CNN الحديثة المصنوعة يدويًا (VGG، ResNet، DenseNet) والمنافسين الأقران من الفئة التلقائية (Large-Scale Evolution) من حيث معدل خطأ التصنيف، كما تُظهر الخوارزمية المقترحة أداءً تنافسيًا مقارنةً بالمنافسين من الفئة شبه التلقائية (Block-QNN-S، Genetic CNN).



الشكل 7: المسار التطوري للخوارزمية المقترحة على CIFAR10.



الشكل 8: المسار التطوري للخوارزمية المقترحة على CIFAR100.

يُظهر الشكلان (7، 8) المسار التطوري للخوارزمية المقترحة على مجموعتي البيانات CIFAR10 و CIFAR100 حيث تعبر كل قيمة على المسار عن القيمة المتوسطة لدقة التصنيف بالنسبة لأفراد الجيل الواحد.

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 ملاحظة أن أحدث شبكات CNN المصممة يدويًا لا تحتوي على البيانات المتعلقة بعدد أيام GPU.

الجدول 2: مقارنة بين الخوارزمية المقترحة ومجموعة من المنافسين من حيث خطأ التصنيف وعدد البارامترات ومدة التدريب على CIFAR10.

	CIFAR10	Parameters	GPU days
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	2750
CGP-CNN [22]	5.98	2.64 M	27
Proposed Algorithm	4.3	2.7 M	40

يوضح الجدول (2) النتائج التجريبية للخوارزمية المقترحة والمنافسين النظراء المختارين حيث يشير العمود الأول إلى المنافسين المختارين 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 المنافس النظير المقابل. تُظهر النتائج أن الخوارزمية المقترحة تتفوق على جميع شبكات CNN الحديثة المصنوعة يدويًا (DenseNet، VGG، ResNet، Large-Scale) وجميع المنافسين الأقران من الفئة شبه التلقائية (Block-QNN-S، Genetic CNN) والتلقائية (CGP-CNN، Evolution) من حيث معدل خطأ التصنيف على مجموعة البيانات CIFAR10.

الجدول 3: مقارنة بين الخوارزمية المقترحة ومجموعة من المنافسين من حيث خطأ التصنيف وعدد البارامترات ومدة التدريب على CIFAR100.

	CIFAR100	Parameters	GPU days
DenseNet (k=12) [8]	24.42	1.0 M	-
ResNet (depth=101) [6]	25.16	1.7 M	-

4. الاستنتاجات والتوصيات

4.1 الاستنتاجات

الهدف من هذه الورقة هو تطوير خوارزمية تصميم بنية CNN باستخدام GAs، القدرة على تصميم أفضل بنية CNN للمهمة المعينة بطريقة تلقائية تمامًا واستنادًا إلى الموارد الحسابية المحدودة. تم تحقيق هذا الهدف من خلال إستراتيجية التشفير المقترحة المبنية على كتل شبكات متطورة مع تمثيل متغير الطول للأفراد.

تم فحص الخوارزمية المقترحة على مجموعات بيانات تصنيف الصور المعيارية CIFAR10 و CIFAR100، مقابل ثلاث شبكات CNN حديثة مصممة يدويًا، واثنين من المنافسين يصممون بنية CNN بطريقة شبه تلقائية، وآخرين يصممون بنية CNN بطريقة تلقائية تمامًا.

تُظهر النتائج أن الخوارزمية المقترحة 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.

4.2 التوصيات

- دراسة التعقيد الحسابي للخوارزمية المقترحة والعمل على تخفيضه دون التأثير على دقة التصنيف.

الرموز

Convolutional Neural Network	CNN
Genetic Algorithms	GAs
ResNet Block	RB
DenseNet Block	DB
ResNet Unit	RU
DenseNet Unit	DU
Pooling Unit	PU
Full Connected Unit	FU
Graphics Processing Unit	GPU
Stochastic Gradient Descent	SGD
Rectified Linear Unit	ReLU

وأخيرًا، يُظهر الجدول (4) أفضل بنية شبكة تم التوصل إليها بالنسبة لقاعدة البيانات المعيارية CIFAR10 وفق الخوارزمية المقترحة، كما يُظهر الجدول (5) أفضل بنية تم التوصل إليها على مجموعة البيانات CIFAR100.

الجدول 4: أفضل بنية تم التوصل إليها على CIFAR10.

نوع الوحدة Type	الإعدادات Configuration
RBU	RBs-number: 6, input-channels: 3, output-channels: 128
PU	type: mean-pooling
DBU	DBs-number: 3, k: 40, input-channels: 128, output-channels: 152
PU	type: max-pooling
DBU	DBs-number: 3, k: 40, input-channels: 152, output-channels: 152
PU	type: max-pooling
RBU	RBs-number: 7, input-channels: 152, output-channels: 64
PU	type: max-pooling
DBU	DBs-number: 7, k: 20, input-channels: 64, output-channels: 164
FU	Neurons-number: 1000

الجدول 5: أفضل بنية تم التوصل إليها على CIFAR100.

نوع الوحدة Type	الإعدادات Configuration
DBU	DBs-number: 6, k: 40, input-channels: 64, output-channels: 152
PU	type: mean-pooling
PU	type: max-pooling
DBU	DBs-number: 3, k: 20, input-channels: 152, output-channels: 152
PU	type: max-pooling
RBU	RBs-number: 8, input-channels: 152, output-channels: 204
RBU	RBs-number: 7, input-channels: 204, output-channels: 128
PU	type: max-pooling
FU	Neurons-number: 1000

- [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., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 2278-2324.
- [15] 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.
- [16] Lin, M., Chen, Q., & Yan, S. (2013). Network in network. *ArXiv preprint arXiv: 1312.4400*.
- [17] Mitchell, M. (1998). *An introduction to genetic algorithms*. MIT press.
- [18] 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*.
- [19] 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.
- [20] 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*.
- [21] Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *ArXiv preprint arXiv: 1409.1556*.
- [22] 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).
- [23] Xie, L., & Yuille, A. (2017). Genetic cnn. In *Proceedings of the IEEE international conference on computer vision* (pp. 1379-1388).
- [24] 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).
- [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 approach Multi-level automated object's dialogue control (MADC). *سلسلة العلوم الهندسية ISSN: 2079-3081*, 35(6).
- [5] Dernoncourt, 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] Holland, J. H. (1992). *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. MIT press.
- [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.

Improving the performance of convolutional neural networks using evolutionary computing

Kloda Ibrahim^{1,}, Yaroub Dayoub², Lina Makdessian³*

¹ Department of Information Technology, Faculty of Information and Communication Technology Engineering, Tartous University, Syria.

² Department of Information Technology, Faculty of Information and Communication Technology Engineering, Tartous University, Syria, yaroubdayoub@gmail.com

³ Department of Basic Sciences, Faculty of Information and Communication Technology Engineering, Tartous University, Syria, lina_makdessian@yahoo.fr

*Corresponding author: Eng. Kloda Ibrahim, klodaibrahim@gmail.com

Published online: 30 September 2021

Abstract- In this paper, we have proposed an algorithm to design Convolutional Neural Networks (CNN) structures using Genetic Algorithms (GAs) that are able to learn the best CNN architecture in a completely automatic manner based on limited computing resources. A coding strategy based on sophisticated, hand-designed, modern network blocks has been proposed. So that, the proposed algorithm does not require users with prior knowledge of CNNs, the problem being addressed or even GAs. The performance of the proposed algorithm was evaluated by conducting 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 field. The experimental results showed that the proposed algorithm can be used to automatically find a competitive CNN structure compared to modern models, as this algorithm achieved the best classification accuracy among manually and automatically designed CNNs as well as competitive classification accuracy for semi-automatic algorithms.

Keywords — "Convolutional Neural Networks, CNN, Evolutionary Computing, Deep Learning, Image Classification "