مجلة جامعة طرطوس للبحوث والدراسات العلمية_ سلسلة العلوم الهندسية المجلد (٤) العدد (٣) 2020
Tartous University Journal for Research and Scientific Studies- Engineering Sciences Series Vol. (4) No. (3) 2020

دراسة عن أحدث الخوارزميات المستخدمة للكشف عن الكائنات في صورة وتصنيفها باستخدام التعلم العميق

محمد مازن المحايري*
قصي كنفاني**
رنيم حافظ كيوان***

(تاريخ الإيداع ٣١ / ٣ / ٢٠٢٠ . قُبل للنشر ١٧ / ٦ / ٢٠٢٠)

ملخص

تعتبر مهمة تحديد وتصنيف الكائنات أحد أكثر المجالات إثارة في رؤية الحاسب والذكاء الاصطناعي. تم تطوير خوارزميات وبنى شبكات عصبونية التفافية مدعومة ببيانات تدريب كبيرة وتكنولوجيا حوسبة متقدمة ، سهّلت العمل ومكنت الحاسب في كثير من الأحيان من تجاوز الأداء البشري . احتاجت بعض الخوارزميات تنفيذ الأداء على مرحلتين: مرحلة تحديد المنطقة ومرحلة التصنيف، والبعض الآخر نفذ المهمة بمرحلة واحدة الأمر الذي رفع سرعة التدريب ، بينما آثرت بعض الخوارزميات رفع الدقة على حساب السرعة. تم في هذه المقالة عرض أحدث الخوارزميات المبتكرة في مجال تحديد الكائنات وتصنيفها وهي YOLOv3 و SSD موضحين هيكلة كل خوارزمية والشبكة الأساسية المستخدمة في استخلاص السمات والآلية المتبعة في عملية استخلاص خريطة السمات التي تقود لتحديد الكائنات في الصورة وتصنيفها. أظهرت نتائج التدريب تفوق الخوارزمية SSD من ناحية السرعة [2] (مما زاد من احتمالية التطبيق في الوقت الحقيقي) في حين تفوقت YOLOv3 من ناحية الدقة.

الكلمات المفتاحية: تصنيف الكائنات، تحديد الكائنات، خريطة السمات، الشبكات العصبونية الالتفافية، YOLO، SSD، المربعات المحيطة، قمع التحديدات الزائدة.

*أستاذ – قسم هندسة الحواسيب والأتمتة– كلية الهمندسة الميكانيكية الكهربائية– جامعة دمشق– دمشق– سورية.
**أستاذ مساعد – قسم العلوم الأساسية – كلية الهندسة الميكانيكية والكهربائية– جامعة دمشق– دمشق– سورية.
***طالبة دراسات عليا(دكتوراه) – قسم هندسة الحواسيب والأتمتة– كلية الهندسة الميكانيكية والكهربائية– جامعة دمشق–

مجلة جامعة طرطوس للبحوث والدراسات العلمية_ سلسلة العلوم الهندسية المجلد (٤) العدد (٣) 2020
Tartous University Journal for Research and Scientific Studies- Engineering Sciences Series Vol. (4) No. (3) 2020

A study of the latest algorithms used to detect and classify objects in an image using deep learning

Mohammad Mazen Mahyry*
Qosai Kanafani**
Raneem H Kiwan***

(Received 31 / 3 / 2020 . Accepted 17 / 6 / 2020)

Abstract

The task of identifying and classifying objects is one of the most exciting areas of computer vision and artificial intelligence. By the development of Convolutional Neural Network (CNN) structures supported by large training data and advanced computing technology, which facilitated the work and often enabled the computer to exceed human performance. Some algorithms needed to be implemented performance in two phases: the region determination phase and classification one, while others implemented the task with one stage, which raised the speed of training, while some algorithms chose to increase accuracy at the expense of speed. In this article, the latest innovative algorithms in the field of object detection and classification, YOLOv3 and SSD, were presented showing the structure of each algorithm and the basic network used in extracting features and the mechanism used in the process of extracting the map of features that leads to the detection and classification of objects in the image. The training results demonstrated the superiority of the SSD algorithm in terms of speed [2] (which increased the likelihood of application in real time) while YOLOv3 excelled in terms of accuracy.

Keywords: Object classification, Object detection, Feature map, Convolutional Neural Network (CNN), YOLO (You Only Look Once), SSD (Single shot Detector), SSD (Single Shot Detector), Non- Maximum Suppression (NMS), Bounding box.

*Professor, Department of Computer Engineering and Automation, Faculty of Mechanical and electrical Engineering, Damascus University, Damascus, Syria.
** Associate Professor, Department of Basic Sciences, Faculty of Mechanical and electrical Engineering, Damascus University, Damascus, Syria.
***Postgraduate Student in computer Engineering, Department of Computer Engineering and Automation, Faculty of Mechanical and electrical Engineering, Damascus University, Damascus, Syria.

مقدمة:

مع التطورات الحديثة في نماذج رؤية الحاسب المعتمدة على التعلم العميق، أصبحت تطبيقات تحديد الكائنات detection object وتصنيفها object classification أسهل من أي وقت مضى. إلى جانب التحسينات الكبيرة في الأداء فقد استفادت هذه التقنيات أيضا من مجموعات بيانات الصور الضخمة لتقليل الحاجة إلى مجموعات البيانات الكبيرة.[19]

عمل الباحثون بداية على تطوير خوارزميات تحديد وتصنيف الكائنات بالاعتماد على مفهوم التعلم العميق مكونة من مرحلتين: مرحلة اقتراح المنطقة (RP) Region Proposal متبوعة بمرحلة التحديد والتصنيف. وناقشت الأبحاث مفهومي الدقة والسرعة. إذ ترتبط الدقة بشكل عام بازدياد عمق النموذج لكون الاتصالات بين الطبقات تصبح أقل، إضافة إلى حجم المرشحات filters المستخدمة حيث إن الحجوم الصغيرة للمرشحات تستخلص المعلومات الموزعة محليا مما يزيد من دقة مهمة التحديد والتصنيف. بينما تتناقص السرعة بازدياد عمق الشبكة فيزداد زمن التنفيذ وتصبح مكلفة حسابيا. يمكن قياس تطور الخوارزمية المعتبرة من خلال التحسينات المقدمة لكل من الدقة والسرعة. [1]

اعتمد الباحثون في عملهم على قواعد بيانات صور متوفرة للعموم مثل COCO (Microsoft **C**ommon **O**bject in **Co**ntext)، PASCAL VOC (**PASCAL V**isual **O**bject **C**lasses)، ImageNet وغيرها، والتي تتكون من آلاف أصناف الكائنات الشائعة وملايين الصور، لتدريب النموذج أو لقياس خوارزميات التصنيف.[1]

تضمنت نماذج التحديد والتصنيف شبكات عصبونية التفافية مثل VGG16 و ResNet لاستخلاص السمات المرئية المعقدة عبر التزايد الهرمي للنماذج (إذ تتعلم الطبقات الأولى سمات محلية بسيطة كالحواف، بينما يتم في الطبقات الأعمق الحصول على تمثيلات الكائن الدلالية كالشكل).[1] ساعدت هذه الشبكات في انخفاض معدل خطأ التصنيف في السنوات ٢٠١٥ و ٢٠١٦ و ٢٠١٧ إلى ٣,٦% و ٣,١% و ٢,٣% على التوالي، بينما كان معدل الخطأ البشري ٥%. [1]

قدم David وآخرون في العام ٢٠١٣، إطارًا متكاملًا لاستخدام الشبكات العصبونية الالتفافية في مهام التحديد والتصنيف.حيث استُخدمت الشبكة OverFeat لاستخلاص السمات بالاعتماد على النافذة المنزلقة متعددة القياسات، وقد أثبتت فعاليتها ضمن البنى الالتفافية في مجال زيادة السرعة مقابل انخفاض الدقة إذ وصل معدل الخطأ إلى ٢٩,٩%. أظهرت الدراسة آلية العمل على تحديد موقع الكائنات في الصورة من خلال التنبؤ بحدود الكائن، ليتم بعدها مراكمة المربعات المحيطة الناتجة عن عملية التحديد بدلا عن تثبيتها. بيّن الباحثون إمكانية تعلم مهام مختلفة بنفس الوقت باستخدام شبكة واحدة. حازت الدراسة على المرتبة الأولى في المسابقة (ImageNet Large Scale Visual Recognition Competition: ILSVRC 2013) بنتائج تنافسية للغاية في مجال التحديد والتصنيف. [2]

اقترح Pierre Sermanet وآخرون في العام ٢٠١٤، بنية شبكة عصبونية التفافية أطلقوا عليها اسم Inception، اعتُبرت أحدث خوارزمية لمهام التحديد والتصنيف في (ILSVRC 2014). تمكنت هذه البنية من تحسين استخدام موارد الحسابات ضمن الشبكة، من خلال التصميم المتقن الذي سمح بزيادة العمق والعرض للشبكة بأحجام مختلفة لمرشحاتها على نفس المستوى مع الحفاظ على ثبات الميزانية الحسابية. لتحسين الجودة ، والهندسة المعمارية للبنية تم الاستناد على مبدأ Hebbian والمعالجة متعددة القياسات. تجسدت هذه البنية في (ILSVRC 2014) بالشبكة GoogLeNet المكونة من ٢٢ طبقة ، وأثبتت أنها فعالة حسابيا وعملت على زيادة الدقة بمقدار ٠,٦%. [3]

منسّق:الخط: (افتراضي) Simplified Arabic، ١٣ نقطة، خط اللغة العربية وغيرها: Simplified Arabic، ١٣ نقطة

منسّق:الخط: (افتراضي) Simplified Arabic، ١٣ نقطة، خط اللغة العربية وغيرها: Simplified Arabic، ١٣ نقطة

منسّق:الخط: (افتراضي) Simplified Arabic، ١٣ نقطة، خط اللغة العربية وغيرها: Simplified Arabic، ١٣ نقطة

منسّق:الخط: (افتراضي) Simplified Arabic، ١٣ نقطة، خط اللغة العربية وغيرها: Simplified Arabic، ١٣ نقطة

منسّق:الخط: (افتراضي) Simplified Arabic، ١٣ نقطة، خط اللغة العربية وغيرها: Simplified Arabic، ١٣ نقطة

منسّق:الخط: (افتراضي) Simplified Arabic، ١٣ نقطة، خط اللغة العربية وغيرها: Simplified Arabic، ١٣ نقطة

منسّق:الخط: (افتراضي) Simplified Arabic، ١٣ نقطة، خط اللغة العربية وغيرها: Simplified Arabic، ١٣ نقطة

منسّق:الخط: (افتراضي) Simplified Arabic، خط اللغة العربية وغيرها: Simplified Arabic

وجد Kaiming He وآخرون في العام ٢٠١٥، أن الشبكات العصبونية الالتفافية العميقة أكثر صعوبة في التدريب من غيرها، فأوجدوا إطارا جديدا للتعلم سهّل من تدريب الشبكات الأكثر عمقا من تلك المستخدمة سابقا. إذ يتم إعادة صياغة الطبقات كتوابع تعلم learning residual متغيراتها هي مدخلات الطبقة ، بدلاً من التعلم كتوابع غير مرجعية. قدمت الدراسة أدلة تجريبية شاملة على أن هذا النوع من البنى أسهل في التحسين، ويساعدنا في الحصول على دقة مناسبة بالرغم من الزيادة الكبيرة في العمق. وصل عمق الشبكة ResNet المستخدمة إلى ١٥٢ طبقة –ثمانية أضعاف عمق الشبكة VGG19 [6] ولكن بدون تعقيد ملحوظ. وقد أحرزت نسبة خطأ ٣,٥٧ ٪ عند الاختبار باستخدام ImageNet مقابل النسبة 7.0% للشبكة VGG19. حصلت هذه البنية على المركز الأول في مهمة التصنيف ضمن (ILSVRC 2015). [4]

انطلق Jie Hu وآخرون في العام ٢٠١٧، من فكرة أن اللبنة الأساسية في الشبكات العصبونية الالتفافية هي المشغل الالتفافي. إذ بحثت الكثير من الدراسات السابقة في العنصر المكاني للعلاقة بين القنوات والحقول المستقبلة المحلية لكل طبقة بغية تعزيز القوة التمثيلية للشبكات العصبونية الالتفافية CNN من خلال تحسين جودة الترميزات المكانية عبر التسلسل الهرمي لسماتها. اقترحت الدراسة وحدة معمارية جديدة أطلقت عليها اسم "الضغط والإثارة" (SE Squeeze and Excitation) تقوم بإعادة ضبط معايرة استجابات السمة لكل قناة من خلال نمذجة الترابط بين خرائط السمات الالتفافية بإثارة السمات ذات الصلة في الوقت الذي يتم قمع السمات التي ليس لها صلة. أوضحت الدراسة إمكانية تكديس تلك الكتل معًا لتشكيل بنى SENet القابلة للتعميم بشكل فعال للغاية عبر قواعد البيانات المختلفة. كذلك فقد جلبت كتل SE تحسينات كبيرة في أداء شبكات CNN الحديثة بتكلفة حسابية طفيفة. حازت الشبكة على المركز الأول للتصنيف في (ILSVRC 2017) متجاوزة المركز الأول للعام ٢٠١٥ بتحسن نسبي يصل ٢٥%، وكانت نسبة الخطأ ٢,٢٥١% إذ وصلت الدقة إلى 91.98%. [6]

في حين أن الأطر المعتمدة مسبقا لتحديد وتصنيف الكائنات كانت عبارة عن طرق بمرحلتين، عمل الباحثون على تطوير بنى تحديد وتصنيف كائنات بمرحلة واحدة، أي إزالة مرحلة اقتراح المنطقة ودمج عمليتي التحديد والتصنيف بمرحلة واحدة،[1] وبالتالي إعادة صياغة مهمة تحديد الكائن باعتبارها مشكلة واحدة ، مباشرةً من بكسلات الصورة الدخل إلى إحداثيات المربع المحيط بالكائن واحتمالات الصنف[20]. مما أدى إلى اختصار الزمن الذي تتطلبة خوارزمية التحديد والتصنيف في مرحلة اقتراح المنطقة.

قدم Jason Brownlee وآخرون، البنية YOLO (You Only Look Once)، وهي سلسلة من نماذج التعلم العميق المصمّمة لتحديد وتصنيف الكائنات بشكل سريع، تتضمن بنيتها شبكة عصبونية التفافية عميقة واحدة (بداية تم استخدام GoogleNet ليتم لاحقا تحديثها إلى الشبكة DarkNet التي تعتمد على VGG16).[8] يتم تقسيم الدّخل إلى شبكة من الخلايا لتتنبأ كل خلية بعدة مربعات محيطة حول الكائن ودرجة الصنف لهذا الكائن. لتكون النتيجة عدد كبير من المربعات المحيطة المرشحة والتي يتم توحيدها لنحصل على التنبؤ النهائي عبر خطوة المعالجة اللاحقة. هناك ثلاثة تغييرات رئيسية تم إحداثها على هذه البنية لغاية كتابة هذه المقالة وهي [9]YOLO v1, 2016 ، YOLO v2 , 2017 [10]، [11] YOLO v3, 2018. الإصدار الأول لهذه البنية احتوى الهيكلية العامة، بينما عمل الإصدار الثاني على صقل التصميم وجعله يستخدم المربعات المحيطة المعرفة مسبقا لتحسين اقتراح المربعات المحيطة، بينما عمل الاصدار الثالث على صقل بنية التصميم وعملية التدريب معا[8]. اشتهرت هذه النماذج بسرعتها لدرجة أنه يمكن استخدامها في تطبيقات الزمن الحقيقي.

صمم [12] Wei Liu وآخرون، بنية لتحديد وتصنيف الكائنات في الصور باستخدام شبكة عصبونية التفافية عميقة واحدة أطلقوا عليها اسم SSD (Single Shot Detector). تعتمد هذه البنية في تشكيل فضاء الخرج من المربعات المحيطية على مجموعة من المربعات المحيطية الافتراضية بقياسات واتجاهات مختلفة لكل موقع من خريطة السمة. في وقت التنبؤ، تقوم الشبكة بحساب درجات لوجود كل صنف من أصناف قاعدة البيانات ضمن كل مربع افتراضي ثم تنتج تعديلات على بيانات المربع (أبعاده) لمطابقته مع شكل الكائن بشكل أفضل. كذلك ، تجمع الشبكة بين التنبؤات الناتجة عن خرائط السمات المتعددة لهذه البنية والتي تملك دقة مختلفة لتحديد وتصنيف الكائنات بأحجام مختلفة. أظهرت النتائج التجريبية على مجموعة بيانات PASCAL و COCO و ImageNet أن SSD تحقق معدل دقة منافس لما تقدمه نظيراتها من البنى المستخدمة في هذا المجال، كما أنه كلما كان حجم الدخل لهذه البنية أكبر– حجم الصورة المدخلة إلى حد معين– قدمت نتائج دقة أفضل.ففي البنية SSD–300 كان متوسط معدل الدقة mAP ٧٤,٣%، في حين وصل في البنية SSD–512 إلى ٧٦,٩%.[12]

أهمية البحث وأهدافه:

أصبحت الأجهزة الرقمية قوية بما يكفي للتعامل مع الحسابات المطلوبة لتنفيذ نماذج CNN بأقرب زمن حقيقي، هناك العديد من الطّرائق المقترحة لعملية تحديد وتصنيف الكائنات في صورة تمّ تنفيذها عمليا ببنى مختلفة خلال العقود الماضية حققت نتائج جيدة من ناحية الدقة تارة وزادت من سرعة المعالجة على حساب الدقة تارة أخرى.

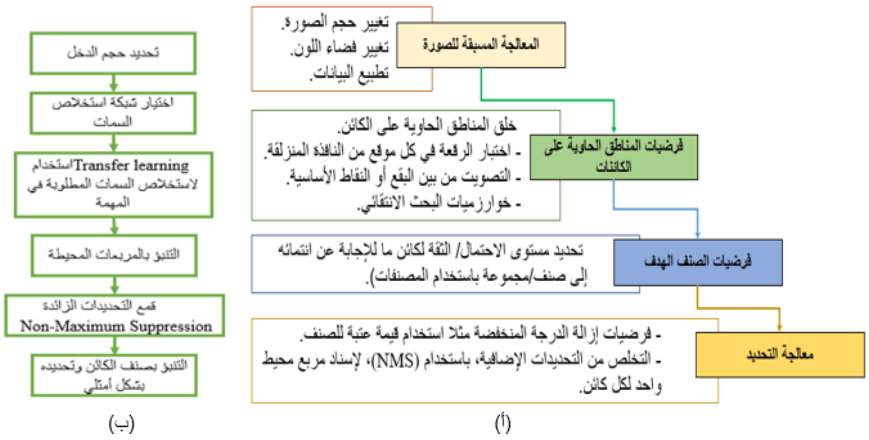
يهدف البحث الحالي إلى دراسة أحدث خوارزميتين متبعتين في مجال التحديد والتصنيف وهما YOLO v3 و SSD حيث تتم مناقشة منصة العمل المستخدمة ، هيكلية كل بنية ، آلية العمل في استخلاص السمات وكيفية تحديد الكائنات في الصورة ودرجة انتماء كل كائن إلى صنفه، عدد المربعات المحيطية المقترحة ومقارنة أدائهما. وذلك لبيان آخر ما توصل له مجال الذكاء الصنعي من آليات في التحديد ليكون منطلقا للبحث وتطويرا لهذا المجال في كل من النقاط التالية:

١– التوصل لنظام يقوم بتحديد الكائنات وتصنيفها آليا باستخدام شبكات التعلم العميق.
٢– تقليل كمية الحسابات الضخمة التي تتطلبها CNN، بما يتقابل مع زيادة الدقة والسرعة.
٣– استخدام الحد الأدنى من الذاكرة للحسابات التي تتطلبها مثل هذه الأبحاث.

طرائق البحث ومواده:

تمكنت البنيتان YOLOv3 و SSD من تحقيق مهمة تحديد وتصنيف الكائنات ببنى عالية الانتاجية ذات مرحلة واحدة باستخدام الشبكات العصبونية الالتفافية كمستخلص سمات معتمدين بذلك على مفهوم نقل التعلم transfer learning، والذي يمكننا من الاستفادة من خرائط السمات التي تم الحصول عليها من تدريب الشبكات العصبونية الالتفافية المعتبرة على قواعد بيانات كبيرة للحصول على خرائط سمات جديدة في العمق تساعدنا في عملية التحديد والتصنيف.

نهجت كلتا الخوارزميتان مفاهيم محددة في عملية بناء البنية ، يوضح الشكل (١–أ) المخطط الصندوقي العام لتسلسل اتباع هذه المفاهيم في بنى التحديد والتصنيف عامة، والشكل (١–ب) يوضح المخطط التدفقي للخطوات العامة المتبعة في بناء البنية SSD و البنية YOLOv3.



الشكل (١) : (أ) المفاهيم العامة لبناء خوارزمية تحديد الكائنات وتصنيفها في صورة،(ب) : المخطط التدفقي لمراحل بناء خوارزمية تحديد الكائنات وتصنيفها في صورة.

1 منصات العمل المستخدمة لتنفيذ نماذج التحديد والتصنيف باستخدام شبكات التعلم العميق:

هناك العديد من المنصات المعتمدة في مهام التحديد والتصنيف باستخدام شبكات التعلم العميق، مثل TensorFlow، Keras، DarkNet. وهي منصات تتنافس فيما بينها لدعم نمذجة هذه المهام. يتم كتابة هذه المنصات بلغات برمجية مختلفة مثل C وغيرها، تحتوي على مكتبات لتدريب الشبكات العصبونية العميقة. تمتلك كل منصة إمكانية تصديرها لتعمل ضمن أي لغة مطورة لنموذج التعلم العميق مثل Python، JavaScript المستخدم لبرمجة تطبيقات الخلوي.

TensorFlow: هي واجهة للتعبير عن خوارزميات التعلم الآلي وتحقيقها وتنفيذها. يتيح هيكلها المرن سهولة نشر العمليات الحسابية عبر الأنظمة الذكية غير المتجانسة دون تغيير يذكر ، بدءًا من الأجهزة المحمولة كالهواتف والأجهزة اللوحية وحتى الأنظمة الموزعة على نطاق واسع لمئات الآلات والآلاف من الأجهزة الحاسوبية. [14] وكذلك النشر عبر مجموعة متنوعة من الأنظمة الأساسية (وحدات المعالجة المركزية ، وحدات معالجة الرسومات) ، ومن أجهزة الكمبيوتر المكتبية إلى مجموعات الخوادم إلى الأجهزة المحمولة والأجهزة المتطورة. [23]

يتميز النظام بالمرونة ويمكن استخدامه للتعبير عن مجموعة متنوعة من الخوارزميات ، بما في ذلك خوارزميات التدريب ونمذجة الشبكات العصبونية العميقة، وتعلم الآلة لأكثر من عشرة مجالات من علوم الحاسوب كالتعرف على الكلام ، رؤية الحاسب ، الروبوتات ، استرجاع المعلومات ، معالجة اللغات الطبيعية ، استخراج المعلومات الجغرافية ، واكتشاف الأدوية الحاسوبية. تم إصدار واجهة برمجة تطبيقات TensorFlow كحزمة مفتوحة المصدر تتعهد بترخيص Apache 2.0 في تشرين الأول ٢٠١٥ ، وهي متاحة على الموقع www..tensorflow.org [14]

Keras: هي مكتبة شبكة عصبونية مفتوحة المصدر مكتوبة بلغة Python. تعمل بتوافق مع المنصة TensorFlow ومجموعة أدوات Microsoft Cognitive وغيرها. تم تصميمها لتمكن مطوري الشبكات العصبونية الالتفافية من التجريب السريع. فامتلكت ميزات عدة كسهولة الاستخدام ، إمكانية البناء على شكل وحدات ، وقابلية

التوسيع.[25] وقدرتها على تقديم تعليقات واضحة وقابلة للتنفيذ عند حدوث خطأ مستخدم. [24] تم فيما بعد –٢٠١٧– دعم Keras في مكتبة TensorFlow الأساسية من قبل فريق Google TensorFlow. لم تكن هذه المكتبة مجرد إطار عمل للتعلم الآلي المستقل وإنما واجهة قدمت مجموعة من التجريدات بالمستوى الأعلى والتي سهلت تطوير نماذج تعليمية عميقة بغض النظر عن الخلفية الحسابية المستخدمة.[25] وبالتالي يُفهم النموذج على أنه تسلسل أو رسم بياني لوحدات قائمة بحد ذاتها قابلة للتكوين بالكامل ويمكن توصيلها بأقل قدر ممكن من القيود. على وجه الخصوص ، فإن الطبقات العصبونية ، توابع التكلفة ، والمحسنات ، وتوابع التفعيل وغيرها ، كلها وحدات مستقلة يمكنك دمجها لإنشاء نماذج جديدة. [24]

DarkNet: هو إطار شبكة عصبونية مفتوحة المصدر مكتوبة بلغة C و CUDA [26] ويعمل كأساس للبنية YOLO وإطار لتدريبها. [27] يتصف بأنه سريع وسهل التثبيت ويدعم حسابات وحدة المعالجة المركزية SPU، ووحدة معالجة الرسوميات. [26]

2 نقل التعلم Transfer Learning:[22]

تعد عملية نقل التعلم transfer learning طريقة شائعة في مجال رؤية الحاسب، يتم التعبير عنها عادة من خلال استخدام نماذج شبكات عصبونية التفافية مدربة مسبقًا على مجموعة بيانات مرجعية كبيرة لحل مشكلة مشابهة لتلك التي نريد حلها، فنتجاوز بذلك التكلفة الحسابية والزمن الطويل لتدريب النماذج الحديثة. من الأمثلة على هذه النماذج المدربة مسبقا VGG و Inception و MobileNet.

تحتوي الشبكة العصبونية الالتفافية النموذجية CNN على جزأين، كما هو موضح في الشكل (٢):

١– القاعدة الالتفافية convolutional base: والتي تتكدس فيها مجموعة من الطبقات الالتفافية تتخللها عدة طبقات تجميع pooling، ويكون عملها الأساسي استخلاص السمات من الصورة الدخل.

٢– المصنف classifier: وهو سلسلة من الطبقات المتصلة بشكل كامل Fully connected Layer. الهدف الرئيسي منه تصنيف الصورة بناءً على السمات المستخلصة.

الشكل(٢): بنية نموذج الشبكة العصبونية الالتفافية النموذجي.[22]

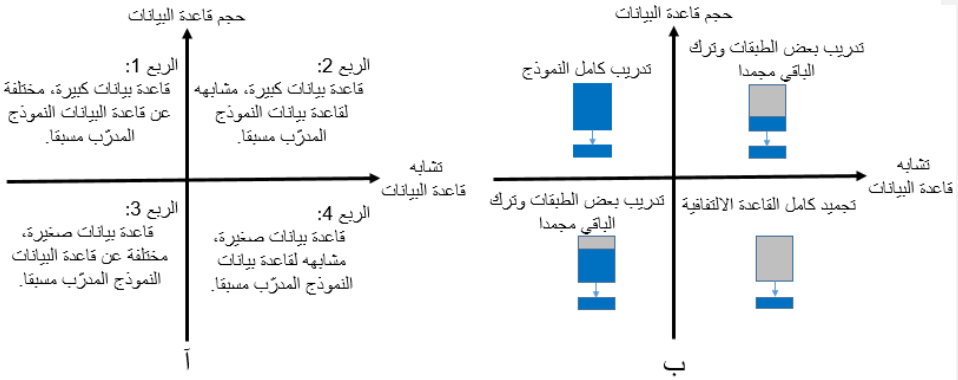
أحد الجوانب المهمة في نماذج التعلم العميق قدرتها على التعلم التلقائي للسمات هرميا. أي أن السمات المحسوبة من قبل الطبقة الأولى عامة (الملامح العامة في الصورة) ويمكن إعادة استخدامها لحل مشكلات مختلفة ، في حين أن السمات المحسوبة من قبل الطبقة الأخيرة محددة (سمات متخصصة) وتعتمد على مجموعة البيانات والمهمة المختارة.

عندما يتم استيراد نموذج شبكة عصبونية التفافية تم تدريبه مسبقًا ليُستخدم في حل مشكلة أخرى، يجب ضبط آلية التدريب للنموذج الجديد وفقًا لإحدى الاستراتيجيات الثلاث، كماه موضح في الشكل (٣–ب):

١–تدريب النموذج بأكمله.

٢– تدريب بعض الطبقات الالتفافية (سفلية) إضافة للمصنف ، وترك الطبقات الالتفافية العليا مجمدة.

٣–تجميد القاعدة الالتفافية.



الشكل(٣): مصفوفة حجم التشابه التي تساعد في تحديد استراتيجية بناء نموذج شبكة عصبونية التفافية جديد بالاعتماد على مفهوم نقل التعلم.[22]

بالاعتماد على نقل التعلم يتم تنفيذ النموذج الجديد بتحديد نموذج الشبكة العصبونية الالتفافية المدرب مسبقا على نفس منصة العمل التي سنقوم بتدريب نموذجنا الجديد عليها، فمثلا لو كانت منصة العمل Keras عندئذ يمكن اختيار النماذج التالية : VGG (2014) و InceptionV3 (2015) و ResNet5 (٢٠١٥). ثم وبالاعتماد على مصفوفة حجم التشابهSize–Similarity Matrix الموضحة في الشكل (٣–آ) نقوم بتصنيف مهمة الرؤية الحاسوبية قيد المعالجة آخذين بعين الاعتبار حجم قاعدة البيانات وتشابهها مع قاعدة البيانات التي تم تدريب النموذج المختار عليها. على سبيل المثال ، إذا كانت المهمة المعتبرة هي تحديد القطط والكلاب في الصورة وتصنيفها، فيمكن اعتبار أن ImageNet مجموعة بيانات مماثلة لأنها تحتوي على صور للقطط والكلاب، في حين إذا كانت المهمة هي تحديد الخلايا السرطانية ، فلا يمكن اعتبار ImageNet مجموعة بيانات مماثلة. بعد ذلك، يتم بناء النموذج الجديد والاعتماد على مصفوفة حجم التشابه الموضحة في الشكل (٣–ب) والتي نستخلص منها آلية التدريب للنموذج الجديد وفقًا لإحدى الاستراتيجيات الثلاث المذكورة آنفا.[22] خلال نقل التدريب، يتم نسخ الأوزان من الطبقات الأولى للنموذج المدرب مسبقا إلى النموذج الجديد والتي تتضمن معلومات حول السمات الأساسية الموجودة في الكائنات مثل اللون، الشكل، الحواف، الخطوط.أما طبقة التصنيف الأخيرة، المسؤولة عن التصنيف عالي المستوى للكائن ضمن مجموعات الأصناف فلا يمكن لها أن تُنقل. يتم تدريب النموذج الجديد لاحقا لمهمة تحديد وتصنيف الأصناف الجديدة. [15]

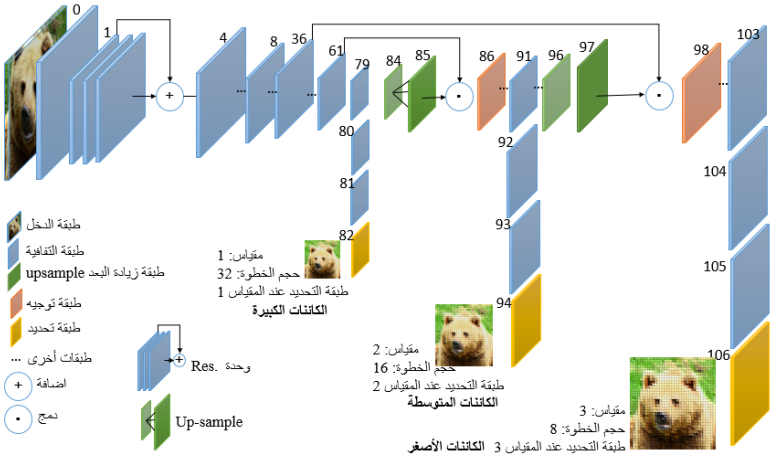
3 التابع Non– Maximum Suppression:[13]

يعتبر هذا التابع بمثابة خطوة أساسية تلي عملية المعالجة في العديد من تطبيقات رؤية الحاسب. يعمل هذا التابع في "مهمة تحديد وتصنيف الكائن" على تحويل خريطة الاستجابة التي تحوي عدة تحديدات للكائن في الصورة بدرجات دقة مختلفة – عدد من المربعات المحيطة– إلى مربع محيط واحد لكل كائن وهي الحالة المثالية وذلك بالاعتماد على قيمة عتبة لدرجة الثقة بوجود كائن في الصندوق المحيط تتجاوز ٠,٥ يتم تحديدها للبنية عند التدريب.[13, 21] يوضح الشكل (10) آلية عمل هذا التابع.

المناقشة:

قدمت البنيتان YOLOv3 المدربة ضمن المنصة DarkNet والبنية SSD المدربة ضمن المنصة Tevsorflow العديد من التحسينات على النماذج المستخدمة في مجال التحديد والتصنيف. [28]

٤-١ هيكلية الشبكة: اعتمدت YOLOv3 على استخدام شبكة أساسية مدربة مسبقا لمهام التحديد والتصنيف تدعى Darknet-53 نسبة لعدد طبقاتها الالتفافية ٥٣ وهي نموذج هجين بين الشبكتين YOLOv2 [10] و Darknet-19 [11]. يوضح الشكل (٤) الهيكلية العامة للبنية YOLOv3،التي استخدمت من البداية للنهاية طبقات التفافية فقط حجم مرشحاتها 3x3 و 1x1 على التوالي، [29] مما جعلها شبكة التفافية بشكل كامل Fully Convolutional Network (FCN)، إضافة إلى اعتمادها على طبقات تخطي الاتصالات skip connections وطبقات تكبير الصورة Up-sampling في مراحل اسخلاص خرائط السمات. تتخلى هذه البنية عن طبقات التجميع Pooling مقابل استخدام حجم الخطوة Stride في الطبقات الالتفافية بالمقدار ٢ ليتم اختزال خرائط السمات، فلو كانت خطوة الشبكة ٣٢، فسيكون خرج صورة مقاسها ٤١٦ × ٤١٦ خريطة سمات حجمها ١٣ × ١٣.[15] يبين الشكل (٤) أن عدد طبقات

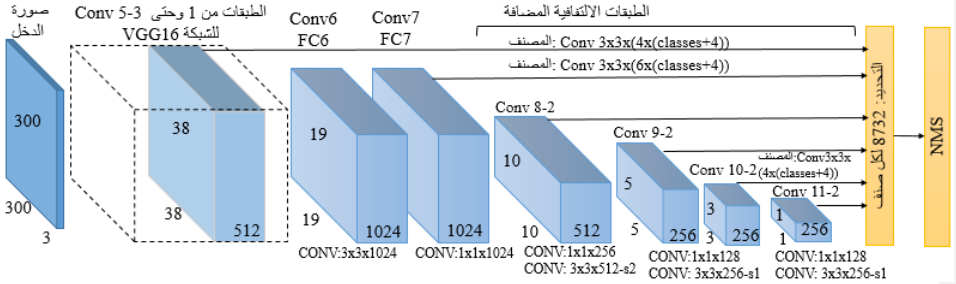


هذه البنية ١٠٦ طبقة وذلك لإضافة ٥٣ طبقة التفافية إلى الشبكة الأساسية لتقوم بمهام التحديد والتصنيف بالتعاون مع طبقات Darknet-53.

الشكل (٤): البنية YOLOv3، تظهر الطبقات والمقاييس وحجم الخطوة.

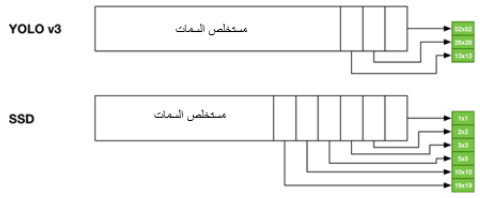
أما البنية SSD الموضحة هيكليتها في الشكل (٥) والمكونة من سلسلة من الطبقات الالتفافية أمامية التغذية feed forward، شبكة استخلاص السمات لهذه البنية (من الكتلة ١ إلى الكتلة ٥) هي جزء مأخوذ من الشبكة VGG16 بأوزانها المدربة مسبقا على قاعدة البيانات COCO، وجودتها العالية في مهام التصنيف. تعمل SSD على تحويل طبقات الاتصال الكامل FC6 و FC7 الموجودة في الشبكة VGG16 إلى طبقات التفافية Conv6 و Conv7 [17] ثم يتم إضافة العديد من الطبقات الالتفافية والتي ستؤدي من خلال خرائط السمات الناتجة عنها إلى تنفيذ مهمة التحديد والتصنيف للكائنات بحجومها المختلفة.[12]

٢ **استخلاص السمات:** تقوم بنى تحديد الكائنات وتصنيفها بشكل عام ، بتمرير السمات التي تعلمتها الطبقات الالتفافية إلى المصنف، الذي يقوم بدوره بالتنبؤ بمعلومات التحديد والتصنيف (إحداثيات المربعات المحيطية ، درجة الثقة للكائن ودرجة الصنف، وغير ذلك...). [15] في بنى التحديد والتصنيف بمرحلة واحدة يكون لدينا كما وجدنا في البنيتين SSD-300 و YOLOv3 شبكة أساسية base network تقوم بمهام استخلاص السمات مدربة على قواعد البيانات القياسية مثل COCO، يليها عدد من الطبقات الالتفافية التي تستخدم سمات الشبكة الأساسية لانجاز مهمة التحديد والتصنيف. يقوم مستخلص السمات في البنية YOLOv3 بادخال صورة حجمها 416x416، في حين يكون حجم الصورة في البينة SSD-300 300x300، حيث أن هذا الحجم في كلتا الحالتين أكبر من الحجم المستخدم في التصنيف باستخدام الشبكة الأساسية (الحجم القياسي لصورة الدخل 224x224) وقد تمت زيادة الحجم للحيلولة دون ضياع التفاصيل الصغيرة في الصورة. يتم إعادة تدريب الشبكة الأساسية على الحجم الجديد، وتضاف طبقات التفافية عديدة مهمتها التنبؤ بالمربعات المحيطة واحتمالات الأصناف للكائنات الموجودة ضمن هذه المربعات، وهذا هو جزء التحديد في مثل هذه البنى.[31]



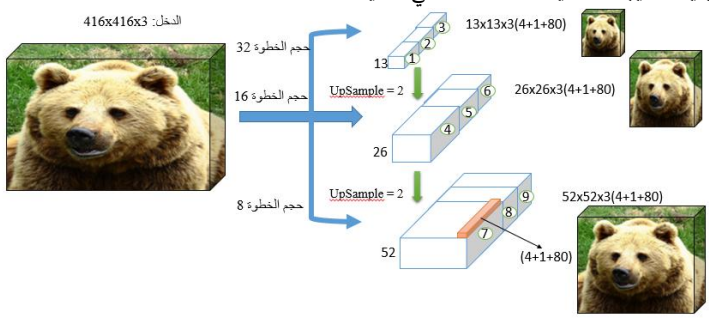
الشكل (٥): البنية SSD-300.

يوضح الشكل (٦) مخطط عام لهاتين البنيتين من وجهة نظر الشبكة الأساسية وخرائط السمات المستخلصة من تدريب هذه البنى، وهي ثلاث خرائط في YOLOv3 وست خرائط في SSD. تقسم الشبكة خريطة السمات الناتجة عن الطبقات الإلتفافية المضافة إلى مجموعة من الخلايا cells، يمكن لكل خلية منها التنبؤ بعدد ثابت من الصناديق المحيطية bounding boxes. [29] [12] يتم في YOLOv3 استخدام تقنية K-means clustering لتحديد ثلاثة نقاط لكل خلية من خلايا خريطة السمات وبالتالي التنبؤ بثلاثة مربعات محيطة فيها لتكون هذه المربعات (أو بعض منها) مسؤولة بدرجة معينة عن الإحاطة بالكائن الذي يكون مركزه هذه الخلية. [15] يستخدم في البنية SSD بعض الصيغ الرياضية لحساب أحجام المربعات المحيطية الافتراضية [30] والتي يختلف عددها بين ٤ أو ٦ مربعات لكل خلية حسب حجم خريطة السمات.



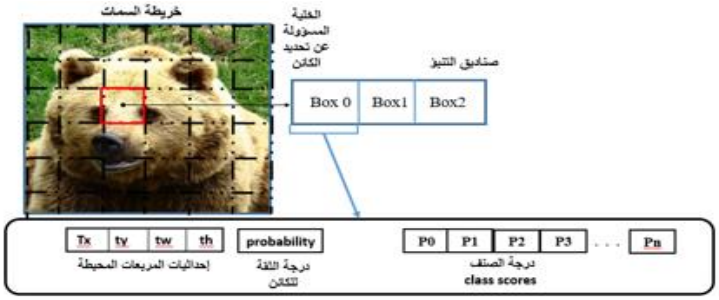
الشكل(٦): مخطط عام للبنيتين YOLOv3 و SSD من وجهة نظر الشبكة الأساسية وخرائط السمات المستخلصة.

٣ خرائط السمات: تمتلك YOLOv3 ثلاثة خرائط سمات بأحجام (قياسات) مختلفة تزداد هذه القياسات كلما اتجهنا للأمام نتيجة طبقات Up–sampling وتستخلص السمات من هذه المقاييس باستخدام منهجية شبكات السمات الهرمية. لاستنتاج خريطة السمات الثانية مثلا، يتم التعامل مع خريطة السمات الموجودة قبل آخر طبقتين (الطبقة ٨٤) ومن ثم إجرء عمليات Up sampling بمقدار 2 و من ثم نأخذ خريطة سمات من الطبقات الأولى (الطبقة ٦١) في الشبكة الأساسية (مستخلص السمات) ونقوم بدمجها مع السمات الناتجة عن up sampling باستخدام الدمج concatenationكما هو موضح في الشكل(٤). تساعد هذه الطريقة في الحصول على معلومات دلالية أكثر فائدة من السمات الناتجة عن Up sampling والمعلومات الحبيبية التي تم الحصول عليها من خريطة السمات في الطبقات الأولى. نحصل على تنبؤ N x N x [3*(4+1+80)] حيث تمثل ٣ عدد المربعات المحيطية لكل خلية، ٤ أبعاد المربع المحيط، ١ يمثل التنبؤ بدرجة الكائن، و ٨٠ لتنبؤات الصنف – قيمة لكل صنف بعدد الأصناف المختبرة. [15] ثم نضيف بضعة طبقات التفافية لمعالجة خريطة السمات المدمجة ونكرر نفس المخطط للحصول على مربعات التنبؤ للمقياس الثالث. وكذا فإن تنبؤاتنا للمقاييس الثلاثة تستفيد من كل الحسابات السابقة بالإضافة للسمات الحبيبية المحسوبة في الطبقات الأولى. يوضح الشكل (٧) المقاييس الثلاثة في البنية YOLOv3. يوضح الشكل (٨) مكونات المربع المحيط الذي يمثل نتيجة التصنيف لكل مصنف في الخلية.



الشكل (٧): خرائط السمات الثلاثة للبنية YOLOv3 بقياساتها المختلفة.

في البنية SSD، يتم تمرير صورة الدخل بالحجم 300x300 عبر الشبكة الأساسية المأخوذة من الشبكة VGG16 والطبقات الالتفافية المضافة فينتج خرائط سمات بأحجام (قياسات) متناقصة كلما تقدمنا في المسار الأمامي نتيجة down sampling وتكون قياسات الخرائط 19x19 ، 10x10 ، 5x5، 3x3، 1x1 بالإضافة إلى خريطة السمات 38x38 الناتجة عن الطبقة Conv4_3 من الشبكة VGG16. تستخدم خرائط السمات هذه لمهمة التنبؤ بالمربعات المحيطية [12].بحيث أن خرائط السمات الناتجة عن الطبقات الالتفافية الأولى تحدد وتصنف الكائنات الصغيرة ، بينما يمكن للطبقات اللاحقة، تحديد الكائنات الأكبر حجمًا بشكل أفضل.[32] وبالتالي تتنبأ هذه الطبقات بكائنات متعددة القياسات. [12]



الشكل (٨): مكونات المربع المحيط للبنية YOLOv3.

٤ المربعات المحيطة المتنبأ بها: تنتج خريطة السمات الأولى في البنية YOLOv3 بحجم خطوة ٣٢ عن حجم صورة الدخل 416x416 ليكون حجم خريطة السمات 13=416/32 أي 13x13،(كل خلية من هذه الخلايا تمثل ٣٢ بكسل من صورة الدخل، تدعى هذه البكسلات بمستقبل الخلية) بينما يكون حجم الخطوة في المقياس الثاني ١٦ وفي المقياس الثالث ٨ ليكون حجم خريطتي السمات الناتجتين على التوالي 26x26 للمقياس الثاني و 52x52 للمقياس الثالث، كما هو موضح في الشكل(٧). تفيد هذه المقاييس الثلاثة في الكشف عن الكائنات في الصورة باختلاف أحجامها، فكلما زاد حجم خريطة السمات تكون المستقبلات لخلاياها صغيرة الحجم لذلك يمكنها الكشف عن الكائنات الأصغر. يمكن أن نستنتج أن عدد المربعات المحيطية التي تتنبأ بها البينة YOLOv3 هي ١٠٦٤٧ مربعا محيطيا، (3x13x13 = 507 ، 3x26x26 = 2028 ، 3x52x52 = 8112). وبالتالي يكون عدد القيم التي تدخل في مهمة التحديد والتصنيف لكل خريطة سمات تساوي [(80+1+4)*3] x N x N.

في البنية SSD لدينا C = 80 صنف مع اعتبار أن احتمال الخلفية مساو للصفر (على اعتبار الخلفية صنفا)، فإن كل مربع محيط يتمثل بشعاع له (4+C) بُعد، تمثل ٤ قيم الإزاحة من مركز المربع الافتراضي وأبعاده (Δcx, Δcy, w, h). إضافة إلى C قيمة احتمالية لدرجة الصنف. عدد المربعات المحيطية المتنبأ بها من خرائط السمات في البنية SSD هي ٨٧٣٢، يمكن استنتاجهامن خلال العلاقة: m*n*B حيث أن m*n حجم خريطة السمات و B عدد المربعات المحيطية الافتراضية لكل خلية من خلايا خريطة السمات، :(Conv4_3: 38x38x4=5776, Conv7: 19x19X6=2166, Conv8_2: 10x10x6=600, Conv9_2: 5x5x6=150, Conv10_2: 3x3x4=36, Conv11_2: 4) . وبالتالي يكون عدد القيم التي نحصل عليها في كل خريطة سمات =(4+C)f*B* . m*n*B*(4+C)

٥ درجة وجود الكائن: تتحدد درجة وجود الكائن (درجة الثقة) في كلتا البنيتين من مقدار التداخل بين المربع المحيط المتنبأ به مع المربع المحيط الحقيقي المحدد عند تجهيز قاعدة البيانات ويؤخذ أفضل تداخل من بين أي مربع محيط آخر. فإذا لم يكن المربع المحيط هو الأفضل ولكنه يتداخل مع المربع المحيط الحقيقي بقيمة أعلى من العتبة بقليل، والتي عادة ما تكون ٠,٥ عندها نتجاهل التنبؤ. يتم بالنهاية العمل على إسناد مربع محيط تنبؤي واحد فقط لكل مربع محيط حقيقي. في حال لم يتم إسناد مربع تنبؤي إلى مربع حقيقي فهذا لا يعني أنه يوجد ضياع في الأبعاد ولا في التنبؤات وإنما في درجة وجود الكائن.

٦ درجة الصنف: يتوقع كل مربع درجة الصنف الذي قد يحتوي عليه المربع المحيط باستخدام التصنيف متعدد العناوين أي إعطاء درجة لاحتمال وجود كل صنف من أصناف قاعدة البيانات. يتم استخدام مصنفات منطقية مستقلة

بدلا من طبقة softmax المستخدمة في البنى الالتفافية التقليدية، تساعد هذه الآلية في تحديد درجة الصنف لدى العمل ضمن مجموعة بيانات الصور المفتوحة[6] والتي تحتوي العديد من العناوين المتداخلة (أي المرأة والشخص). غلما أنه باستخدام softmax يتم الفرض أن كل مربع لديه بالضبط صنف واحد وهو ليس كذلك في كثير من الحالات. لذا يعتبر التصنيف المتعدد نهج جيد لمثل هذه النماذج.

التدريب training:

تعتبر الصناديق الافتراضية المتنبأ بها من قبل البنية عند التدريب صناديق محيطية محددة بعناية وممثلة بالاعتماد على أحجامها ونسب أبعادها ومواقعها في الصورة. يهدف نموذج التحديد والتصنيف إلى البحث عن المربعات الافتراضية التي تتوافق مع قيم العتبة المقترحة ومن ثم يتم التنبؤ بإزاحات الأبعاد لتلك المربعات للحصول على التنبؤ النهائي والذي يمتلك معلومات المربع المحيط ودرجة وجود الكائن ودرجة الصنف.[32] يمر ذلك في كلتا البنيتين بعدة خطوات تعتمد على المقارنة بين المربع المحيط الحقيقي المحيط بالكائن والمحدد في قاعدة البيانات مع كل مربع افتراضي متنبأ به من خلال البرامتر Intersection Over Union (IOU) المسمى بمؤشر جاكارد Jacard index، وهو مقياس يقيس مقدار التداخل بين المربعين المحيطين الحقيقي والافتراضي ويقاس بمقدار عتبة تحدد مسبقا كما هو موضح بالشكل (٩)، فإذا كانت قيمة التداخل أكبر من العتبة المحددة ، عندئذ يتم اعتماد المربع المحيط الافتراضي كنتيجة مقبولة، وتكون قيمة عنوان الكائن abjectness label للمربع الافتراضي ١ وتكون درجة وجود الكائن للمربع الافتراضي هي قيمة المقياس جاكارد (مقدار التداخل). تتنبأ كل خلية من خريطة السمات بكائن ما، إذا كان مركز الكائن يقع في حقل الاستلام لتلك الخلية.[29]

أثناء عملية التطابق في SSD يمكن لمربعين محيطين حقيقيين أن يتداخلا مع نفس المربع الافتراضي. بهذه الحالة، سيتم الاحتفاظ بأحدهم وهو الذي تكون عنده قيمة مؤشر جاكارد وبالتالي فإن الشعاع المعبر عن المربع المحيط الافتراضي يمكن أن يحتوي البيانات التالية: إزاحة الموقع g= (cx, cy, w, h) ، درجة وجود الكائن pϵ [0,1]، والعنوان x ϵ {0,1} . يتم حساب إزاحة الموقع وفق العلاقات التالية:

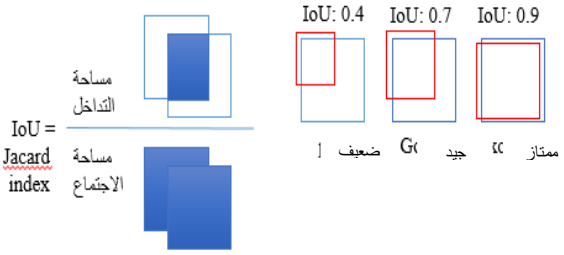
$$cx = (cx_g - cx_d)/w_d \quad (1)$$

$$cy = (cy_g - cy_d)/h_d \quad (2)$$

$$w = \log(\frac{w_g}{w_d}) \quad (3)$$

$$h = \log(\frac{h_g}{h_d}) \quad (4)$$

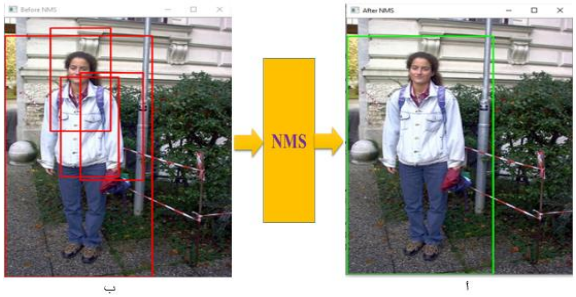
حيث أن (cx ,cy) هي مركز المربع، (w , h) عرض وارتفاع المربع، g تشير للمربع المحيط الحقيقي، d تشير للمربع الافتراضي.[17]



الشكل (٩) مؤشر جاكارد – البرامتر IOU يعبر عن مقدرا التداخل بين مساحة المربع الافتراضي والمربع الحقيقي.

في YOLOv3 يتم التنبؤ بالقيم الأربعة المعبرة عن المربع المحيط، وهي الاحداثيات x, y لكل مربع محيط والتي يتم تعريفها نسبة إلى الزاوية اليسرى العليا لكل خلية شبكة ويتم تطبيعها مع أبعاد الخلية بحيث تكون قيم الإحداثيات محصورة بين ٠ و ١. وكذلك يتم تحديد عرض المربع وارتفاعه كقيمة الجذر التربيعي للعرض والطول. تعتمد هذه البنية على المربعات المحيطة المحددة مسبقا ذات نسب أبعاد مختلفة وهي ثلاثة مربعات في البنية YOLOv3 تتضمن معلومات حول شكل الكائنات التي نتوقع اكتشافها مما يساعد البنية على التنبؤ بشكل أفضل. وقد أوجد الباحث [15] أفضل طريقة لاكتشاف أفضل نسب قياسات من خلال k–means clustering. إذ أن كل مربع محيط يمكن أن يخصص ليقوم بتحديد كائن ما بحجم ونسبة قياس محددة. إذ أن هناك مربع واحد لكل نقطة في الخلية (الحاوية ٣ نقاط) يكون المحدد الأول مسؤول عن تحديد الكائنات التي تتشابه مع حجم المربع المحيط الأول، أما المحدد الثاني مسؤول عن الكائنات المشابهة في الحجم للمربع الثاني.وبالتالي عدد المحددات في الخلية متطابق مع عدد المربعات فيها.وهكذا فالكائنات الصغيرة يتم تحديدها بمحدد مختلف عن المحدد الذي يقوم بتحديد الكائنات الكبيرة.[30]

من المناقشة وجدنا أن البنى المطروحة تتنبأ بعدد كبير من المربعات المحيطية الافتراضية والتي يتم العمل خلال التدريب على اختيار أفضلها وتجاهل عدد كبير منها معتمدين على تابع الضياع loss function والمتمثل هنا بـ IOU (Intersection Over Union) حيث يتم تحديد قيمة العتبة بشكل متوافق مع الأداء المطلوب وبنهاية هذه المرحلة سيكون لدينا عدد أقل من المربعات المحيطية التي يجب أن نختار مربعا واحدا منها لكل كائن بحيث يحقق أفضل إحاطة بالكائن وتحديد موقعه بشكل دقيق بالصورة وهذا ما يقوم به التابع Non– maximum Suppression كما هو موضح في الشكل التالي(١٠).يعتمد هذا التابع على قيمة درجة الصنف حيث أن أعلى درجة صنف هي التي تنتخب المربع الأفضل وبالتالي يكون خرج هذا التابع هو مربع وحيد محيط بالكائن.



الشكل (١٠) الصندوق المحيط في (أ) تم اختياره من قبل NMS من بين الصناديق المتنبأ في بها لكل كائن، الصناديق المحيطة في (ب) تعبر عن نتيجة التحديد والتصنيف بدون استخدام التابع NMS.

أداء الخوارزميتان – الاختبار Testing:

لمعرفة أداء نموذج التحديد والتصنيف، يمكن ببساطة حساب عدد التنبؤات الصحيحة بالكائنات في صور مجموعة الاختبار والقسمة على مجموع هذه الصور للحصول على دقة التصنيف. يبين الشكل (١١) خرج اختبار البنيتين (أ) YOLOv3 (ب) SSD على صورة من قاعدة البيانات.

ب أ

الشكل (١١) خرج اختبار البنيتين (أ) YOLOv3 (ب) SSD على صورة من قاعدة البيانات.

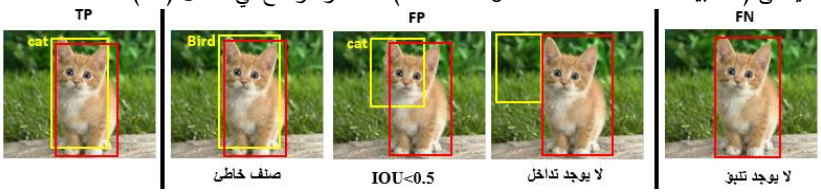
هناك حسابات عديدة أخرى يمكن من خلالها قياس أداء نموذج تحديد الكائنات وتصنيفها وهي:

١- معدل متوسط دقة التصنيف mean Average Precision (mAP): في حالة تعدد الأصناف يتم الاعتماد في ذلك على المقياس mAP، وهو معدل متوسط الدقة لجميع الأصناف الموجودة في قاعدة البيانات والذي يمكن حسابه من خلال حساب دقة التصنيف لكل صنف ومن ثم حساب المعدل بقسمة مجموع قيم الدقة على عدد الأصناف.

٢- قياس مقدار تداخل المربع المحيط المتنبأ به (الافتراضي) مع المربع الحقيقي المحيط بالكائن (IOU).

٣- الاستدعاء Recall (الحساسية أو معدل الإيجاب الصحيح): يحدد فيما إذا كان النموذج سيعثر فعليًا على جميع الكائنات الموجودة في الصورة.

علما أنه لا يمكن اعتماد أحدها بحد ذاته، فمثلا يتيح البرامتر IOU اعتبار أن التنبؤ صحيح (إيجابي حقيقي True Positive TP) إذا كان التداخل بين المربع المحيط الافتراضي والمربع المحيط الحقيقي بنسبة تزيد عن ٥٠٪، وإلا سيتم اعتبار التنبؤ غير صحيح (خطأ إيجابي FP False Positive).ومع ذلك لن يكون ذلك قادرا على معرفة أداء النموذج، إذ لن يتم الإعلام عن حالة الخطأ الناتجة عن عدم التنبؤ بمربعات تتداخل مع المربعات المحيطية الحقيقية وهو ما يدعى (السلبيات الخاطئة False Negatives FN)، كما هو موضح في الشكل (١٢).



الشكال (١٢): تمثيل TP(True Positive) ، FN(False Negative) ، FP (False Positive) بالاعتماد على مقياس التداخل IOU.

الخوارزمية	متوسط الدقة AP عند IOU		معدل متوسط الدقة mAP(%)	عدد المربعات المحيطية المتنبأ بها	حجم صورة الدخل	زمن الاستنتاج (ms)	المرجع
	IOU = 0.5	IOU= 0.75					
YOLOv3	٥٧,٩	٣٤,٤	٥٥,٣	10647	416x416	29	[15]
SSD-300	41.2	23.4	٧٤,٣	8732	300x300	61	[12]

الجدول (١) أداء البنيتين YOLOv3 و SSD.

لدمج كل هذه الجوانب المختلفة في قيمة رقمية واحدة، نستخدم عادةً المقياس mAP. حيث إن العلاقة طردية بين mAP وبين أداء النموذج. من خلال التدريب والاختبار على قاعدة البيانات COCO عند قيمة العتبة IOU = 0.5 نجد أن البنية SSD تحقق mAP = 74.3% [14] بينما تحقق البنية YOLOv3 mAP= 55.3% [32] عند نفس قيمة العتبة، ولكنها أسرع بثلاث مرات من البنية SSD، كما هو موضح في الجدول (1).

Precision: هو عدد TP مقسوما على مجموع التحديدات:

$$Precision = TP / (TP + FP) \qquad (5)$$

حيث أن FP هو تحديد كائن غير موجود فعليا في الصورة، يحدث ذلك عندما يكون المربع المحيط المتنبأ به مختلف تماما عن أي مربع محيط حقيقي في الصورة، أو أن الصنف المتنبأ به ليس هو الصنف الحقيقي.

:Recall

$$Recall = TP / (TP + FN) \qquad (6)$$

الفرق الوحيد بين الصيغتين أن Precision تستخدم عدد FP في المقام بينما Recall تستخدم عدد FN والتي تحدث عندما لا يكون هناك تنبؤ للكائن الموجود فعلا في الصورة (أو أن درجة وجود الكائن منخفضة).

يمكن اعتبار التنبؤ TP إذا كان صنف التنبؤ هو نفس صنف المربع المحيط الحقيقي وحدث التداخل بين المربعات المحيطية الافتراضية والحقيقية بمقدار أكبر من ٥٠% . في حال كان التداخل أقل من ٥٠% ، سيتم اعتبار التنبؤ FP.

في حال كان لدينا تنبؤين أو أكثر قيمة IOU لهما أكبر من ٥٠% لنفس المربع المحيط الحقيقي، فإنه لابد من اختيار واحد فقط من التنبؤات كتنبؤ صحيح، وباقي التنبؤات يمكن اعتبارها FP وذلك لتشجيع النماذج على التنبؤ بمربع واحد فقط لكل كائن. (يتم اختيار التنبؤ الوحيد اعتمادا على أعلى درجة وجود الكائن بغض النظر عن القيم IOU العالية.

لن تشير القيم Precision و Recall إلى أداء النموذج بشكل واضح لذلك تحسب للصنف باعتماد مجال لقيم عتبة IOU كثنائية (recall, precision) تمثل كل منها نقطة في منحني الدقة – الاستدعاء Precision–Recall curve. بحيث يمثل محور السينات الاستدعاء Recall، من ٠ (أي لم يتم إيجاد كائنات حقيقية) إلى ١ (إيجاد كل الكائنات). بينما يمثل محور العينات الدقة Precision.

يمكن الاستفادة من المنحني بمايلي:

١- إيجاد عتبة مناسبة لدرجة وجود الكائن: فاختيار عتبة عالية تعني أننا سنحتفظ بتنبؤات أقل، وبالتالي سيكون لدينا FP أقل أي نرتكب أخطاء أقل. ولكننا بالمقابل سيكون لدينا المزيد من FN أي سنفقد الكثير من الكائنات. أما العتبة الأقل، ستزيد من عدد التنبؤات التي نعتمدها ولكن سيؤدي ذلك في العادة لدقة أقل.

٢- حساب متوسط الدقة Average Precision (AP) من خلال حساب المساحة تحت المنحني باستخدام التكامل integration.

يتم معرفة تحديدات الكائنات النموذجية على أنها صحيحة أو خاطئة اعتمادًا على عتبة IoU. تم اعتماد قيم عشر عتبات مختلفة لقاعدة بيانات COCO للبنية YOLOv3 من ٠,٥ إلى ٠,٩٥ بخطوة ٠,٠٥. يبين الشكل (١٢) منحنيات الدقة –الاستدعاء بالنسبة لكائن محدد من قاعدة البيانات "الشخص".

بطبيعة الحال، فإن النتيجة الأعلى هي الأفضل. لكن لا يعني أن mAP هي كل ما يهمنا فقد حققت YOLOv3 درجات أقل من بعض منافسيها ولكنها حققت سرعة ملحوظة. عندما يكون العمل على الأجهزة المحمولة لا بد من استخدام نماذج تفاضل بين السرعة الدقة. يوضح الجدول (١) أداء البنيتين YOLOv3 و SSD بعد الاختبار على قاعدة البيانات COCO.

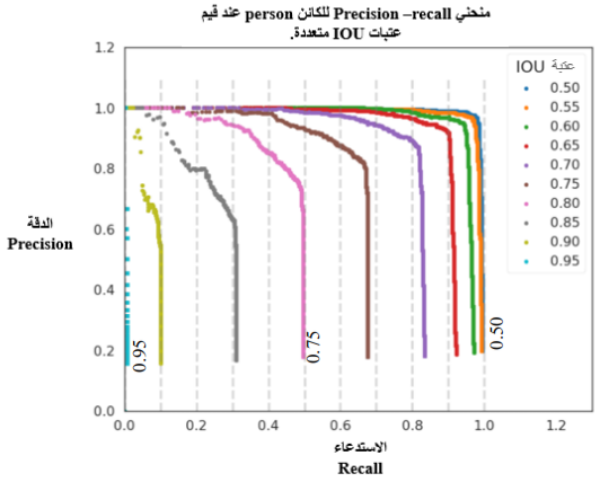
الاستنتاجات والتوصيات:

هناك مفاضلة بين السرعة والدقة. وبما أن المحور الأساسي لنماذج التحديد والتصنيف بمرحلة واحدة هو السرعة. لذلك يجب توخي الحذر عند المفاضلة بين الدقة والسرعة. [33]

من خلال دراسة النتائج وتتبع عمل كلتا البنيتين تبين لنا أنه كلما كانت الشبكة الأساسية أقوى في عملية استخلاص السمات كلما ساعدت الطبقات الالتفافية المضافة في رفع دقة التحديد والتصنيف وهذا ما فعلته البنية SSD إذ أنها استخدمت VGG16 ذات الدقة العالية والقوية في عملية استخلاص السمات.

لعل أهم الأمور التي تم التوصل إليها من خلال دراسة بنى التحديد والتصنيف والتي تساعد الباحثين في بناء هيكلية جديدة تمتلك القدرة على زيادة السرعة في إنجاز الحسابات هي الأخذ بعين الاعتبار حجم المرشحات التي تؤثر في دقة المهمة وعدد الطبقات الالتفافية والتي تتناسب عكسا مع كمية الحسابات وطردا مع الدقة، والاتصالات بين الطبقات، كما أن تحديد البرامترات التالية: معدل التعلم و تهيئة الأوزان – عند تدريب الشبكة الأساسية– وتابع التفعيل له أثر كبير في تحديد أداء النموذج.

وكما وجدنا فإن البنية YOLOv3 لم تتمكن من تحديد الكائنات الصغيرة والمتوسطة بدقة وهذا الذي تفوقت به البنية SSD من خلال زيادة عدد خرائط السمات بأحجامها الصغيرة والكبيرة والمربعات الافتراضية التي تعتمد على أبعاد الخلية مركز الكائن. يمكن للبنية YOLOv3أن تعتمد على زيادة عدد خرائط السمات التي تستنتج الكائنات وتعمل على تغيير حجوم مرشحاتها لتتمكن من استخلاص السمات المفيدة في تحديد الكائنات الأكبر والأوسط، كذلك الأمر أن تعتمد في تحديد الكائنات بالتعاون بين خرائط السمات مجتمعة وليس كل خريطة على حدى الأمر الذي يمكّن من تلافي ضياع المكونات في الصورة. بالمقابل لا يمكن للبنية SSD أن تتجاوز مشكلة البطء من خلال تخفيض عدد خرائط السمات لأن ذلك سيؤثر في الدقة وينخفض الأداء ولكن يمكن لهذه البنية أن تزيد من سرعتها من خلال استخدام أجهزة حديثة أو التدريب على التفرع من خلال المكتبات الحديثة المتوفرة في لغات البرمجة الداعمة.



الشكل (١٢) منحنيات الدقة –الاستدعاء بالنسبة للكائن " الشخص" المحدد من قاعدة البيانات COCO.[34] نجد من اختيارنا للعتبة العالية IOU= 0.95 والتي سيتم عندها الاحتفاظ بتنبؤات أقل فسيكون لدينا FP أقل أي نرتكب أخطاء أقل. ولكننا بالمقابل سيكون لدينا المزيد من FN أي (عدم وجود تنبؤاتلكائنات موجودةفعلا في الصورة) وبالتالي سنفقد التنبؤ بالكثير من الكائنات. أما العتبة الأقل، ستزيد من عدد التنبؤات التي نعتمدها ولكن سيؤدي ذلك في العادة لدقة أقل.

المراجع:

[١]K. Anand, W. Kerry, W. Zhenglin*," Deep learning – Method overview and review of use for fruit detection and yield estimation*", Computers and Electronics in Agriculture 162 (2019) 219–234.

[٢]Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., LeCun, Y., 2013. Overfeat: Integrated recognition, localization and detection using convolutional networks.

[٣]C. Szegedy; W. Liu; Y. Jia, "*Going deeper with convolutions*", 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 15 October 2015, 10.1109/CVPR.2015.7298594.

[٤]Kaiming He, Xiangyu Zhang, "*Deep Residual Learning for Image Recognition*", arXiv: 1512.03385v1 [cs.CV] 10 Dec 2015.

[٥]Simonyan and A. Zisserman. "*Very deep convolutional networks for large-scale image recognition"*. In ICLR, 2015.

[٦] Hu, J., Shen, L., Sun, G., Squeeze-and-excitation networks. arXiv:1709.01507v4 [cs.CV] 16 May 2019.

[٧] Zeng, X., Ouyang, W., Yan, J., Li, H., Xiao, T., Wang, K., Liu, Y., Zhou, Y., Yang, B., Wang, Z., 2018. Crafting gbd-net for object detection. IEEE Trans. Pattern Anal. Mach. Intell. 40, 2109–2123.

[٨] Brownlee, J., *Deep Learning for Computer Vision: Image Classification, Object Detection* ..., p 369, 2019.

[٩] Redmon, J., Divvala, S., Girshick, R., Farhadi, A., *You Only Look Once:Unified, Real-Time Object Detection*, object detection, 2015.

[١٠] H.Lechgar, H. Bekkar, H. Rhinane, *Detection of cities vehicle fleet using YOLO V2 and aerial images, he International Archives of the Photogrammetry, Remote Sensing*

and Spatial Information Sciences, Volume XLII-4/W12, 2019 5th International Conference on Geoinformation Science – GeoAdvances 2018, 10–11 October 2018, Casablanca, Morocco.

[١١] Jing Li, Jinan Gu, Zedong Huang and Jia Wen, *Application Research of Improved YOLOV3 Algorithm in PCB Electronic Component Detection*, Appl. Sci. 2019, 9, 3750; doi:10.3390/app9183750.

[١٢] Wei Liu, W., Anguelov, D., Erhan, D., Szegedy, C., SSD: *Single Shot MultiBox Detector*, arXiv:1512.02325v5 [cs.CV] 29 Dec 2016.

[١٣] Rothe, R., Guillaumin, M., and Gool, L. V., *Non-Maximum Suppression for Object Detectionby Passing Messages between Windows*, p1.

[١٤] Barham, P., Brevdo, E., Chen, Z., Citro, C., et al., "*TensorFlow: Large-scale machine learning on heterogeneous systems*" (PDF). TensorFlow.org. Google Research. Retrieved November 10, 2015.

[١٥] Redmon, J., Farhadi, A., YOLOv3: *An Incremental Improvement*, *arXiv*:1804.02767v1 [cs.CV] 8 Apr 2018.

[١٦] Zhang, P., Zhong, Y., Li, X., SlimYOLOv3:Narrower, *Faster andBetterfor Real-TimeUAV Applications, School of Life Science Beijing Institute of Technology*.

[١٧] Yi, J., Wu, P., J. Hoeppner, D., Metaxas, D., *Fast Neural Cell Detection Using Light-Weight SSD Neural Network, 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops.*

[١٨] Licheng Jiao,*A Survey of Deep Learning-based Object Detection, rXiv:1907.09408v2 [cs.CV]* 10 Oct 2019.

[١٩]https://towardsdatascience.com/going-deep-into-object-detection-bed442d92b34 , ٢٠١٩/١٠/٦ 9:44PM.

[20]https://towardsdatascience.com/yolo-you-only-look-once-real-time-object-detectionexplained-492dc9230006, 23/1/2020, 9:53AM.

[21] https://towardsdatascience.com/implementation-of-mean-average-precision-map-with-non-maximum-suppression-f9311eb92522, 24/1/2020, 9:44PM.

[22]https://towardsdatascience.com/transfer-learning-from-pre-trained-models-f2393f124751, 25/1/2020, 12:19AM.

[23] https://en.wikipedia.org/wiki/TensorFlow, 26/1/2020, 12:25PM.

[24] https://keras.io/#why-this-name-keras, 26/1/2020, 2:50PM.

[25] https://en.wikipedia.org/wiki/Keras, 26/1/2020, 2:51PM.

[26] https://pjreddie.com/darknet/, 26/1/2020, 8:45PM.

[27] https://martinapugliese.github.io/recognise-objects-yolo/, 26/1/2020, 9:39PM.

[28]https://medium.com/@franky07724_57962/exploring-opencvs-deep-learning-object detection-library-e51fe7c82246 6/10/2019 10:16PM.

[29]https://blog.paperspace.com/how-to-implement-a-yolo-object-detector-in-pytorch/, 8/2/2020, 9:29PM.

[30] https://www.jeremyjordan.me/object-detection-one-stage/, 9/2/2020, 1:50PM.

[31]https://machinethink.net/blog/object-detection/, 10/2/2020, 12:45AM.

[32]https://medium.com/inveterate-learner/real-time-object-detection-part-1-understanding-ssd-65797a5e675b, 11/2/2020, 9:19PM.

[33]https://towardsdatascience.com/yolo-v3-object-detection-53fb7d3bfe6b,20/2/2020, 1:15PM.

[34]https://medium.com/@timothycarlen/understanding-the-map-evaluation-metric-for-object-detection-a07fe6962cf3, 1/3/2020, 10:45PM

منسّق:الخط: (افتراضي) +برنامج نصي معقد لعناوين (semiT Roman weN)، خط اللغة العربية وغيرها: +برنامج نصي معقد لعناوين (namoR weN semiT)

منسّق:الخط: (افتراضي) +برنامج نصي معقد لعناوين (semiT Roman weN)، خط اللغة العربية وغيرها: +برنامج نصي معقد لعناوين (namoR weN semiT)

منسّق:بلا تسطير

منسّق:بلا تسطير

تغيير رمز الحقل

منسّق:بلا تسطير