

الخلاصة:

يعد اكتشاف الكائن أحد المهام الأساسية والتي تلبي متطلبات العصر الأساسية في الرؤية الحاسوبية والتي تتكون من تحديد موقع الكائن على الصورة حيث توجد كائنات معينة، بالإضافة لتصنيف تلك الكائنات، وفي عام 2015 وُلدت خوارزمية YOLO (أنت تنظر لمرة واحدة: You Only Look Once) بنهج جديد يعيد صياغة مشكلة اكتشاف الكائن باعتباره مشكلة تدرج، كما أنه يعمل في شبكة عصبية واحدة، وأدى ذلك إلى انفجار مجال اكتشاف الكائنات وحقق إنجازات ملحوظة أكثر بكثير مما كانت عليه قبل عقد من الزمن فقط، وحتى الآن وبالإقتران مع العديد من الأفكار الأكثر ابتكاراً الصادرة عن مجتمع أبحاث الرؤية الحاسوبية، تمت ترقية YOLO إلى سبعة إصدارات وتم تقييمها كواحدة من خوارزميات اكتشاف الكائنات المتميزة، ويعتبر الجيل الثالث من الخوارزمية هو أحدث إصدار تم تطويره بواسطة المؤلف الأصلي لـ YOLO وتعتبر أسرع وأدق بكثير من كلا الإصدارين السابقين للخوارزمية، وفيما يلي سنقوم باستعراض لمحة مبسطة عن خوارزميتي Faster R-CNN و SSD ليتجلى لنا الاختلاف بينها مع الخوارزمية المستخدمة، كما قمنا بطرح نتائج التنفيذ في الزمن الحقيقي ومقارنته مع الخوارزميتين المذكورتين سابقاً، وقمنا بتشغيل الخوارزمية في بايثون على تسلسل فيديو في الزمن الحقيقي واستعراض النتائج، وعلى الله التوفيق.

Abstract:

Object detection is one of the primary tasks in computer vision which consists of determining the location on the image where certain objects are present, as well as classifying those objects. In 2015, the YOLO (You Only Look Once) algorithm was born with a new approach, reframing object detection as a regression problem and performing in a single neural network. That made the object detection field explode and obtained much more remarkable achievements than just a decade ago. So far, combining with many of the most innovative ideas coming out of the computer vision research community, YOLO has been upgraded to seven versions and assessed as one of the outstanding object detection algorithms. The third generation of YOLO (YOLOv3), is the latest version developed by the original author of YOLO (Joseph Redmon). However, the performance of the YOLOv3 is higher than of the previous versions of both accuracy and speed. Below, we'll discuss the difference between Faster R-CNN, SSD and YOLO v3 algorithms by the results. Also, we implement YOLO v3 algorithm in python in real time video series in YouTube.

التعلم العميق:

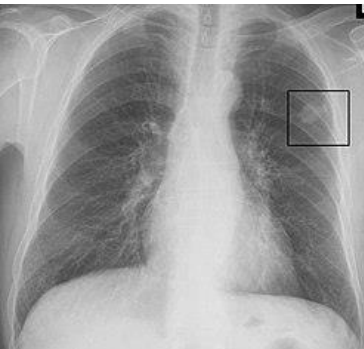
التعلم العميق Deep Learning هي تقنية اخترعها الإنسان من أجل محاولة تقليد الطريقة التي يعمل بها العقل البشري، فالتعلم العميق يحاول أن يحاكي العقل البشري في جميع قدراته، والتي منها: الرؤية، وفهم الحديث، وتكوينه، والسمع، وغيرها من القدرات القوية التي يتمتع بها عقلنا البشري ولا ينافسها فيها أي شيء آخر، ولم يقف الأمر عند هذا الحد فقط بل إن العلماء قد قاموا بدراسة العقل البشري وكيفية عمله من أجل أن يصمموا خوارزميات وبرامج قادرة على محاكاته، ولهذا السبب نجد أن تلك الخوارزميات مستوحاة من الدراسات الطبية والعصبية الخاصة بالإنسان وتحاول قدر الإمكان أن تقلدها، ولكن بطرق حاسوبية لا بيولوجية؛ فالخلايا العصبية Neurons أو الشبكات العصبية Neural Networks قد تم استبدالها بشكل حاسوبي مثل: الشبكات العصبية الالتفافية Convolutional Neural Network، أو الشبكات العصبية التكرارية Recurrent Neural Network، يمكن تطبيق تقنيات وخوارزميات التعلم العميق في العديد من المجالات كالسيارات ذاتية القيادة، وتطبيقات كشف الخداع، ومعالجة اللغات الطبيعية، والطب، وريادة الأعمال، والرؤية الحاسوبية التي هي مجال اهتمامنا في هذا العمل.

تتبع الكائنات:

تعد تقنيات تتبع الكائنات (Object Tracking) من أهم الخوارزميات المستخدمة في مجال الرؤية الحاسوبية والتي تهدف بشكل عام إلى تتبع الكائنات منذ بداية ظهورها في الفيديو أو على عدسة الكاميرا في الوقت الفعلي لحين اختفائها وبالتالي إمكانية تسجيل معلومات حول هذه الكائنات ومعرفة سلوكها، على سبيل المثال في حال المراقبة المرورية تمكننا هذه الخوارزميات من تتبع المركبات ومعرفة إذا كانت قد خالفت قانون السير أم لا (تجاوز السرعة المحددة، تجاوز إشارة المرور)، بالإضافة إلى ذلك يوجد تطبيقات كثيرة ومتنوعة لهذه الخوارزميات منها المراقبة الأمنية (ملاحقة المجرمين) والسيارة ذاتية القيادة والتصوير الطبي وغيرها، وينبغي الإشارة إلى التفريق بين عدة مصطلحات فيما يخص الكائنات والرؤية الحاسوبية لها، وهي:

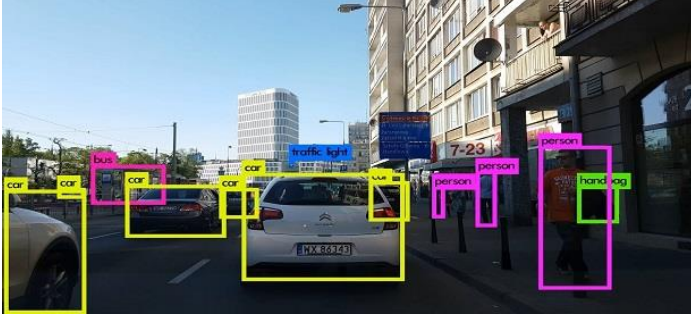
• الكشف عن الكائنات (Object Detection):

هي عملية تحديد الكائنات داخل صندوق إحاطة بناءً على موقعه والتعرف عليها وتسميتها خلال الصور، ومن تطبيقاتها الكشف عن الأورام السرطانية في الصور الشعاعية كما في الشكل المجاور.

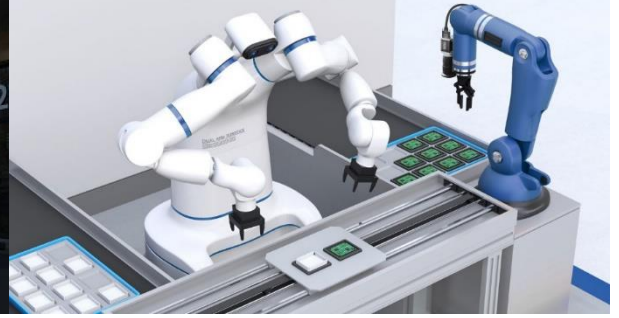


• تتبع الكائنات (Object Tracking):

هي عملية الكشف عن الكائنات وتعقب مواقعها المتغيرة ضمن فيديو محدود أو في الزمن الحقيقي، ومن الأمثلة عليها أنظمة الرقابة المدنية والمرور، والسيارات ذاتية القيادة، والروبوتات بشكل عام، وأنظمة الدفاع والهجوم العسكرية....الخ.



الشكل (2): يوضح تتبع الكائنات لسيارة ذاتية القيادة



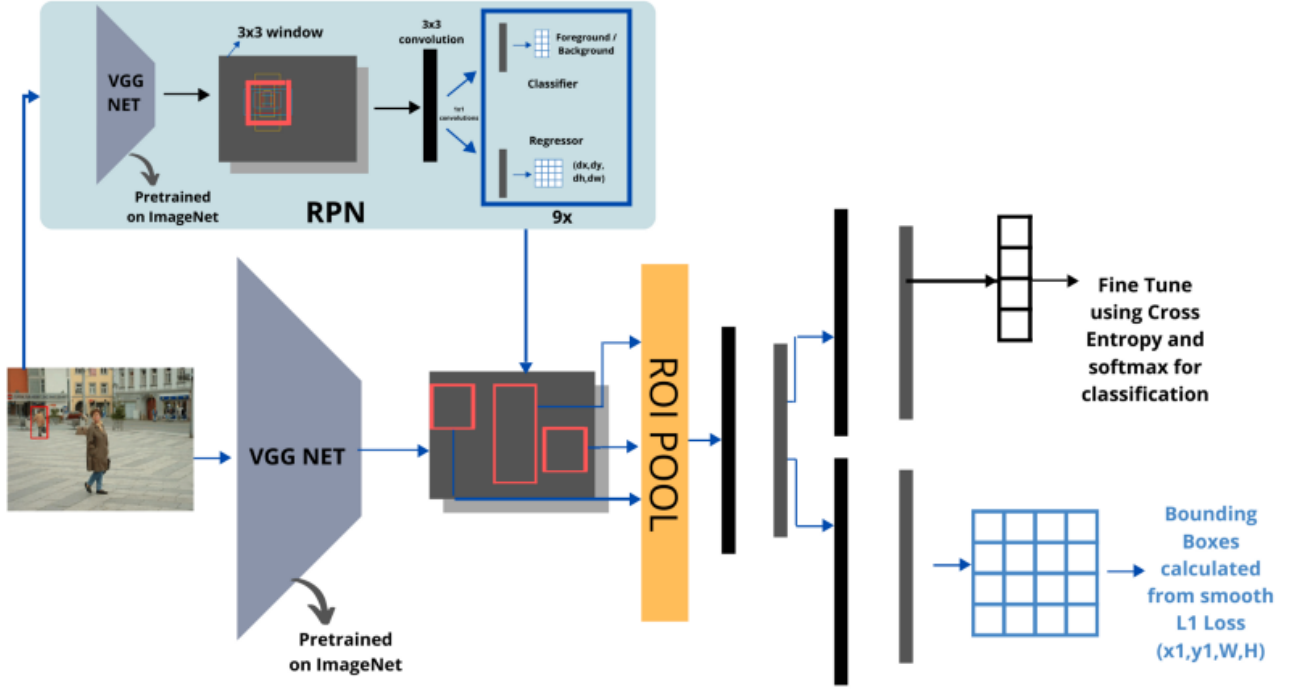
الشكل (1): يوضح تتبع الروبوتات لرقائق الشرائح الإلكترونية

مشكلة البحث:

يعد اكتشاف الكائنات وتتبعها إحدى المشكلات التقليدية في علم الرؤية الحاسوبية، إذ أن الأمر لا يقتصر على تصنيف فئة الكائن ولكن أيضاً تفسير حركته وموقع إحداثياته وأبعاده وتتبعه، وسابقاً كانت الأساليب المستخدمة لحل هذه المشكلات مقتصرة على استخراج مناطق مختلفة في الصورة باستخدام نوافذ منزلة ذات أحجام مختلفة ومن ثم تطبيق تقنيات التصنيف لتحديد فئة الكائن، ولكن تعاني هذه الأساليب من عدة مساوئ كأن تتطلب قدراً كبيراً من الحسابات وتقسيمها إلى عدة مراحل معقدة، وفي عام 2015 قدم الباحث جوزيف ريدمون (Joseph Redmon) وزملاؤه نظاماً جديداً لاكتشاف الكائنات يؤدي جميع المراحل الأساسية لاكتشاف كائن باستخدام شبكة عصبية واحدة ولأول مرة وهي خوارزمية "أنت تنظر لمرة واحدة: You Only Look Once (YOLO)"، حيث قام بإعادة صياغة مشكلة تتبع الكائن على أنه مشكلة انحدار واحدة مباشرة من بكسلات الصورة إلى إحداثيات الصندوق المحيط واحتمالات الفئة؛ إذ يتنبأ هذا النموذج الموحد بمربعات إحاطة متعددة في وقت واحد واحتمالات فئة لكل الكائنات التي تغطيها المربعات، وكانت نتائج الخوارزمية مذهلة وقد تفوقت على الخوارزميات الرئيسية في هذا المجال من حيث السرعة والدقة لاكتشاف إحداثيات الكائن وتحديداتها، وقد تم ترقية الخوارزمية إلى عدة إصدارات حتى وصلت إلى الإصدار السابع (حين كتابة هذا العمل) وترافق مع هذه الترقيات ظهور عدة أفكار ابتكارية في مجتمع الرؤية الحاسوبية، ولكن قدم الباحث الأصلي للخوارزمية ترقيتان فقط واعتزل مجال الرؤية الحاسوبية بالكامل بعد أن وصلت للإصدار الثالث.

خوارزمية (Faster R-CNN (Faster Regional Convolutional Neural Network):

هي شبكة عصبية تلافيفية ذات بنية معمارية محددة تقوم بجمع مقترحات تواجد الكائنات ضمن الصورة من خلال تقنيات لاحقة الذكر ومن ثم تخفيض عدد المقترحات للتجهيز لعملية التصنيف للكائنات المكتشفة في الصورة.



الشكل (3): يوضح البنية المعمارية لشبكة Faster R-CNN

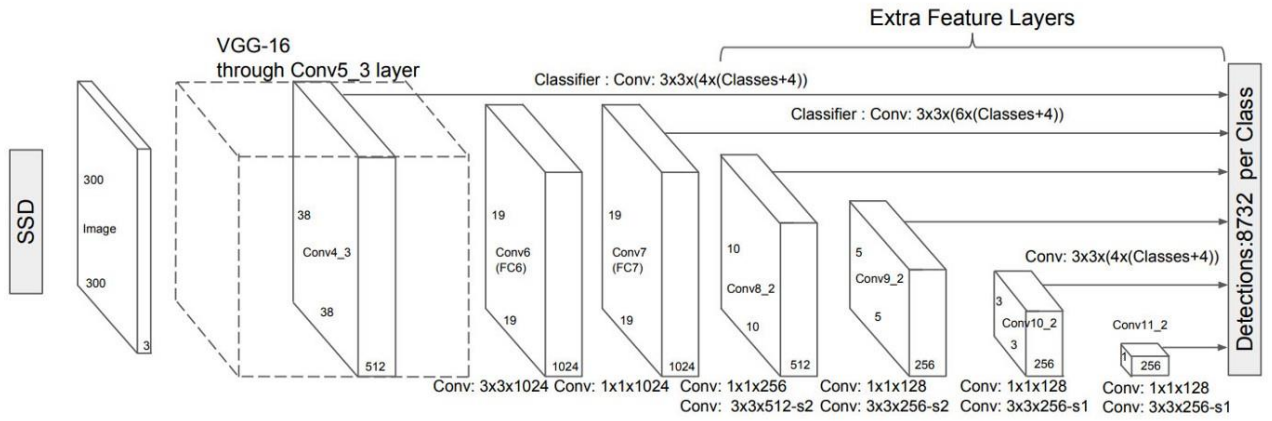
بداية تستخدم الشبكة تقنية الصور الهرمية (Pyramid Images) التي تقوم بتقييس صورة الإدخال والاحتفاظ بها في مكس فيه بالأعلى الصورة بأعلى دقة ممكنة وبالأسفل الصورة بأقل دقة ممكنة؛ وما ذلك إلا لتوفير مقاييس مختلفة لتحليل الصورة، ومن ثم تدخل الصورة على نموذج VGG-16 وهو عبارة عن شبكة عصبية تلافيفية تحتوي على 16 طبقة تلافيفية لها أوزان تسمى الميزات الفائقة (Hybrid Parameters) وهي شبكة كبيرة جداً إذ تحتوي على 138 مليون ميزة معلمة تقريباً بحسب الباحثين وهي قائمة على نموذج ImageNet، ومن ثم تدخل الصورة بعد تحليلها إلى شبكة اقتراح المنطقة (RPN: Region Proposal Network) التي تقوم بجمع مقترحات الكائنات من خلال صناديق الارتكاز (Anchor boxes) وهي عبارة عن صناديق ذات أحجام وأبعاد مختلفة (9 صناديق مختلفة الطول والعرض) من خلال نافذة منزلقة (Sliding Window) تنزلق على المصفوفة لتحمل ميزات الصورة لإدخالها إلى طبقة تلافيفية لتصنيف الكائنات ومن ثم طبقة تجميع قائمة

على عملية الإسقاط في منطقة الاهتمام (Projection on ROI (Projection on Region Of Interest) وأخيراً طبقة SoftMax لتصنيف احتمالات الفئات للكائنات المكتشفة.

خوارزمية الكاشف متعدد الصناديق برمية واحدة (SSD (Single Shot Multi-Boxes Detector):

هي خوارزمية تتبع الكائنات في الزمن الحقيقي تتكون بنيتها المعمارية من نموذجين (كما في شبكة Faster R-CNN السابقة)، وهما:

- **العمود الفقري (Backbone):** وهو عبارة عن شبكة تلافيفية تستخدم لتصنيف الصور ومدرّبة مسبقاً مثل شبكات VGG, ResNet, AlexNet,....etc والتي تم إزالة طبقة الاتصال الكامل منها.
- **الرأس (Head):** ويستخدم لتحديد الكائنات بعد تصنيفها وهي عبارة عن 6 طبقات تلافيفية.



الشكل (4): يوضح المعمارية الدقيقة لخوارزمية SSD

ويمكن تلخيص عملها كالتالي:

- ليس لديها شبكة RPN وتتوقع المربعات المحيطة والفئات بشكل مباشر من خرائط الميزات في مسار نموذجي واحد.
- تعمل على تحسين الدقة من خلال طبقات تلافيفية صغيرة للتنبؤ بفئة الكائن والإزاحات لمربعات الإحاطة الافتراضية (Default Boxes).
- لديها طبقات منفصلة للمربعات الافتراضية للتعامل مع الاختلاف في نسب العرض إلى الارتفاع.
- لديها خرائط متعددة المقاييس (Multi-Scale Features Maps).
- يمكن تدريبها من طرف النموذج الأيسر إلى الطرف الأيمن للحصول على أعلى دقة ممكنة.
- تقدم المزيد من التوقعات ولها تغطية أفضل لمواقع الكائنات ومقاييسها ونسب العرض إلى الارتفاع.

نموذج DarkNet:

هو إطار عمل شبكة عصبية مفتوح المصدر مكتوب بلغة الـ C ويستفيد من CUDA (وهي عبارة عن مكتبة تفرعية مبتكرة من قبل شركة Nvidia لزيادة أداء وقوة معالجة الصور) في عملياته الحسابية ويشبه إطار عمل TensorFlow لمعالجة عمليات الشبكات العصبية ثقيلة الحساب، ويتميز بسرعة ودقة عاليتين وكونه مخصص لاكتشاف الكائنات وله بنية وميزات مختلفة عن إطار التعلم العميق الأخرى؛ بحيث أنه أسرع من شبكة Faster R-CNN ويعتبر البنية المعمارية لخوارزمية YOLO وأحد الأسباب الرئيسية في كون نموذج YOLO عالي السرعة والدقة.

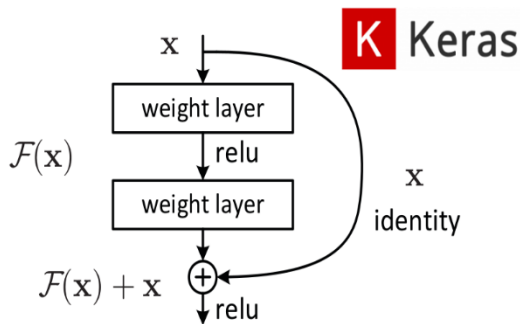
mandelbrot

source	secs	mem	gz	busy	cpu load			
<u>Python 3</u>	263.87	48,268	688	1,054.07	100%	100%	100%	100%
<u>C gcc</u>	1.64	27,024	1135	6.53	99%	99%	100%	100%

الشكل (5): يوضح الفرق بين سرعة النموذج مكتوب بلغة C ولغة Python

خوارزمية YOLO v3 (You Only Look Once version 3):

كانت المشكلة الرئيسية التي تم تلافيتها في الإصدار الثالث من الخوارزمية هو أن الإصدار الثاني يستخدم طبقات تلافيفية عميقة مخصصة لهندسة نموذج DarkNet، بحيث يحتوي على أكثر من 11 طبقة من نموذج الإصدار الأول، وهذا بالنسبة للشبكات العصبية العميقة يعني أن زيادة الطبقات مزيداً من الدقة، ولكن مع ذلك تم تصغير صورة الدخل عند إعادة توجيهه إلى طبقات أعمق مما أدى إلى فقدان الميزات الدقيقة ولهذا السبب لطالما ما عانى الإصدار الثاني مع اكتشاف الكائنات صغيرة الحجم في الصورة، وبالإطلاع على نموذج شبكة ResNet



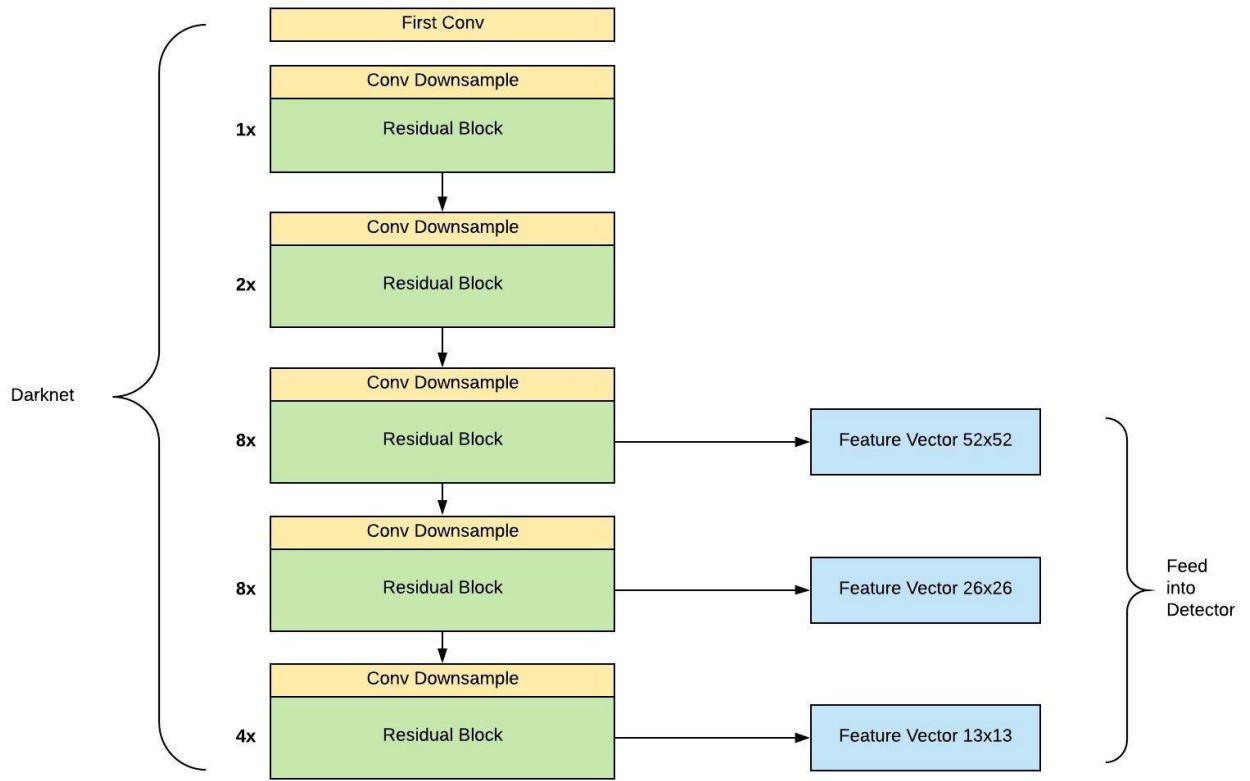
وفكرة البلوكات المتبقية (Residual Blocks) القائمة على عملية تخطي الاتصال (Skip Connection) أوضحت أن هذه العملية تعمل على مساعدة طبقات التنشيط على الانتشار إلى طبقات أعمق دون تلاشي التدرج اللوني الذي يسبب مشكلة تلافي الكائنات الصغيرة، وهذا ما يوضحه الشكل المجاور.

عندئذٍ توصل الباحث جوزيف ريدمون إلى بنية أفضل حيث كان مستخرج الميزات عبارة عن مزيج من نماذج الإصدار الثاني بالإضافة إلى نموذج DarkNet-53 والبلوكات المتبقية في نموذج ResNet وتم بناء هذه الشبكة بهيكل عنق الزجاجة (Bottleneck) (أي نموذج 1×1 متبوعاً بطبقات التقاف 3×3) ويكون داخل كل كتلة متبقية اتصال خطي، وعلاوة على أن الكتل المتبقية من نموذج ResNet سيقوم بتلافي تدهور أداء الشبكة بالطبقات المترابطة فعند اكتشاف الكائنات الصغيرة فإن الطبقات العميقة ستحصل على المزيد من المعلومات مباشرة من الطبقات المسطحة (ذات البعد الواحد) وهي بذلك تتجنب فقدان كل الميزات الدقيقة.

	Type	Filters	Size	Output
	Convolutional	32	3×3	256×256
	Convolutional	64	$3 \times 3 / 2$	128×128
1x	Convolutional	32	1×1	
	Convolutional	64	3×3	
	Residual			128×128
	Convolutional	128	$3 \times 3 / 2$	64×64
2x	Convolutional	64	1×1	
	Convolutional	128	3×3	
	Residual			64×64
	Convolutional	256	$3 \times 3 / 2$	32×32
8x	Convolutional	128	1×1	
	Convolutional	256	3×3	
	Residual			32×32
	Convolutional	512	$3 \times 3 / 2$	16×16
8x	Convolutional	256	1×1	
	Convolutional	512	3×3	
	Residual			16×16
	Convolutional	1024	$3 \times 3 / 2$	8×8
4x	Convolutional	512	1×1	
	Convolutional	1024	3×3	
	Residual			8×8
	Avgpool		Global	
	Connected		1000	
	Softmax			

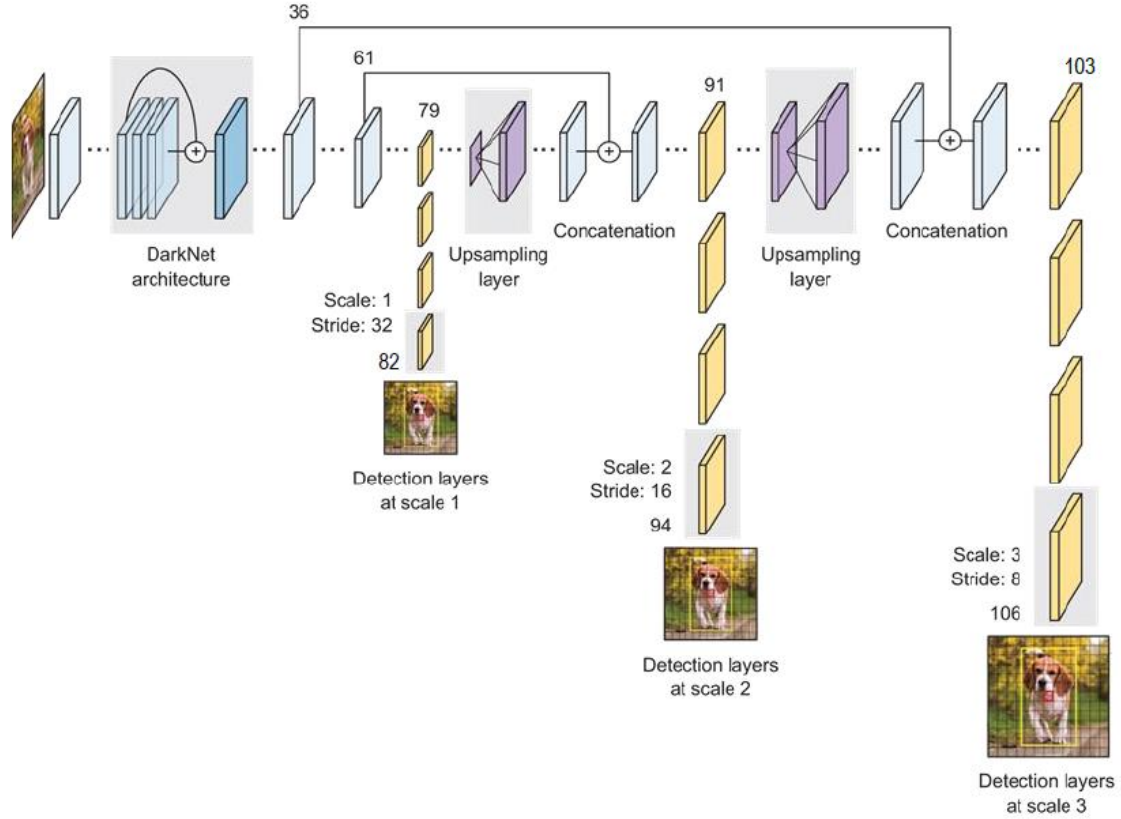
الشكل (6): يوضح بنية DarkNet-53 مع خمس كتل متبقية تحتوي على هيكل عنق الزجاجة

في الإصدارين السابقين للخوارزمية بعد التدريب على مستخرج الميزات بهندسة بنية DarkNet تمت إعادة توجيه الإدخال إلى بعض الطبقات الأخرى وأخيراً يتم عمل التنبؤات في الطبقات الأخيرة من كاشف الكائن، ومع ذلك ألحق نموذج طبقات التنبؤ جانباً بالشبكة بدلاً من تكديسها في الطبقات الأخيرة كما كان من قبل، والميزة الأكثر بروزاً في النموذج الجديد أنها تقوم بالكشف في 3 مستويات مختلفة بحيث تم استخدام الميزات من قبل آخر 3 كتل متبقية لـ 3 أجهزة كشف بمقاييس مختلفة، ويسمى هذا النموذج بشبكة هرمية الميزات (FPN: Feature Pyramid Network) والذي يشبه كثيراً في نهجه نموذج الميزات الهرمية (FPM: Feature Pyramid Model).



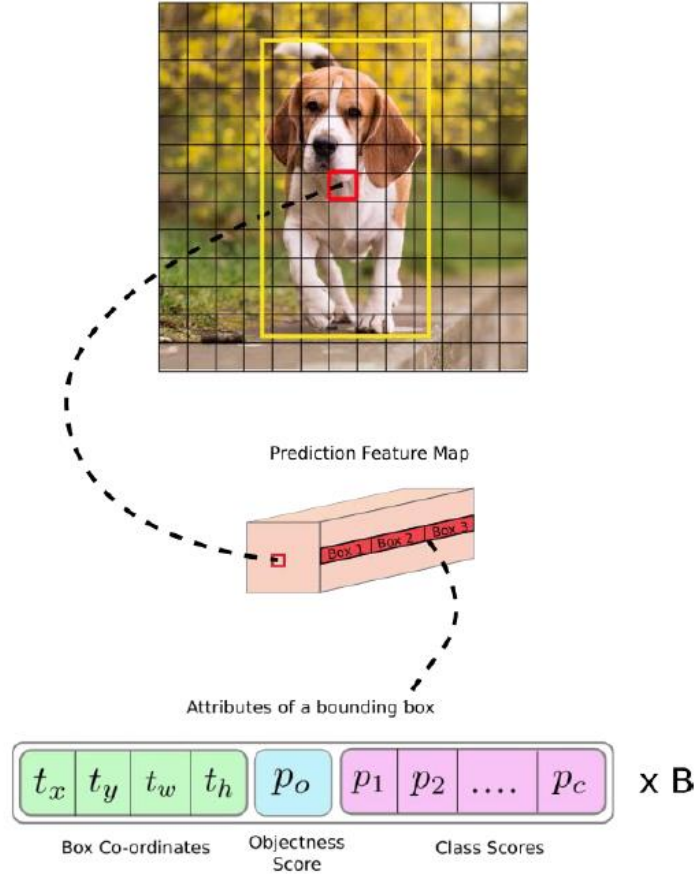
الشكل (7): يوضح إلحاق جهاز الكشف متعدد المقاييس بالشبكة لإجراء الكشف 3 مرات في 3 مستويات مختلفة

وهذه التنبؤات الثلاثة تعمل في الطبقات 82 و 94 و 106 والتي يتم تقديمها بدقة بخطوة (Stride) من الشبكة (أي هي النسبة التي تقلل من خلالها من أبعاد المدخلات) وهي 32 و 16 و 8 على التوالي.



الشكل (8): يوضح بنية شبكة YOLO V3 والتي تجمع كل من مستخرج الميزات وكاشف الكائن

وعلى عكس الإصدار الأول حيث يتم التنبؤ بالمربعات المحيطة (Bounding Boxes) بواسطة نفس الخلية الشبكية التي تشترك بمجموعة من احتمالات التنبؤ من الفئات مما يجعل كل خلية شبكة مسؤولة عن توقع كائن واحد فقط، وفكرة أن خلية الشبكة لديها القدرة على التنبؤ بأشياء متعددة في نفس الوقت ستكتشف المربعات المحيطة كائنات مختلفة حتى لو تم توقعها بواسطة نفس خلية الشبكة وبالتالي فإن مربعات الإحاطة التي تنتبأ بالكائنات لها مجموعة احتمالات التنبؤ الخاصة بها من فئات الكائنات بدلاً من مشاركتها معاً، ولتحسين هذا النهج اقترحت الخوارزمية في إصدارها الثالث معلمات الخرج الإجمالية لكل كاشف مختلف بعدد $S * S * (B * (5 + C))$ ، بحيث يتم الكشف الأول بواسطة الطبقة 82، ولسهولة التفسير استخدم المؤلف صورة دخل $416 * 416$ كأعداد افتراضي، وبعد إعادة التوجيه خلال أول 81 طبقة يتم تصغير صورة الإدخال بخطوة 32 وستكون خريطة الميزات الناتجة بحجم $13 * 13$ والمقابل لخلايا الشبكة $13 * 13$ كما هو موضح في الشكل السابق، وفي كل طبقة كشف يتم الكشف عن طريق تطبيق نواة الكشف $1 * 1$ على خريطة الميزات، وتكون هذه النواة مسؤولة عن التنبؤ بالمربع المحيط لكل خلية شبكة في خريطة الميزات، تم تدريب الإصدار الثالث على مجموعة بيانات COCO مع $B = 3$ و $C = 80$ ، لذا فإن حجم النواة هو: $1 * 1 * (3 * (5 + 80)) = 1 * 1 * 255$ وهي في طبقة الكشف الأولى، وتكون خريطة الميزات النهائية بأبعاد: $13 * 13 * 255$.

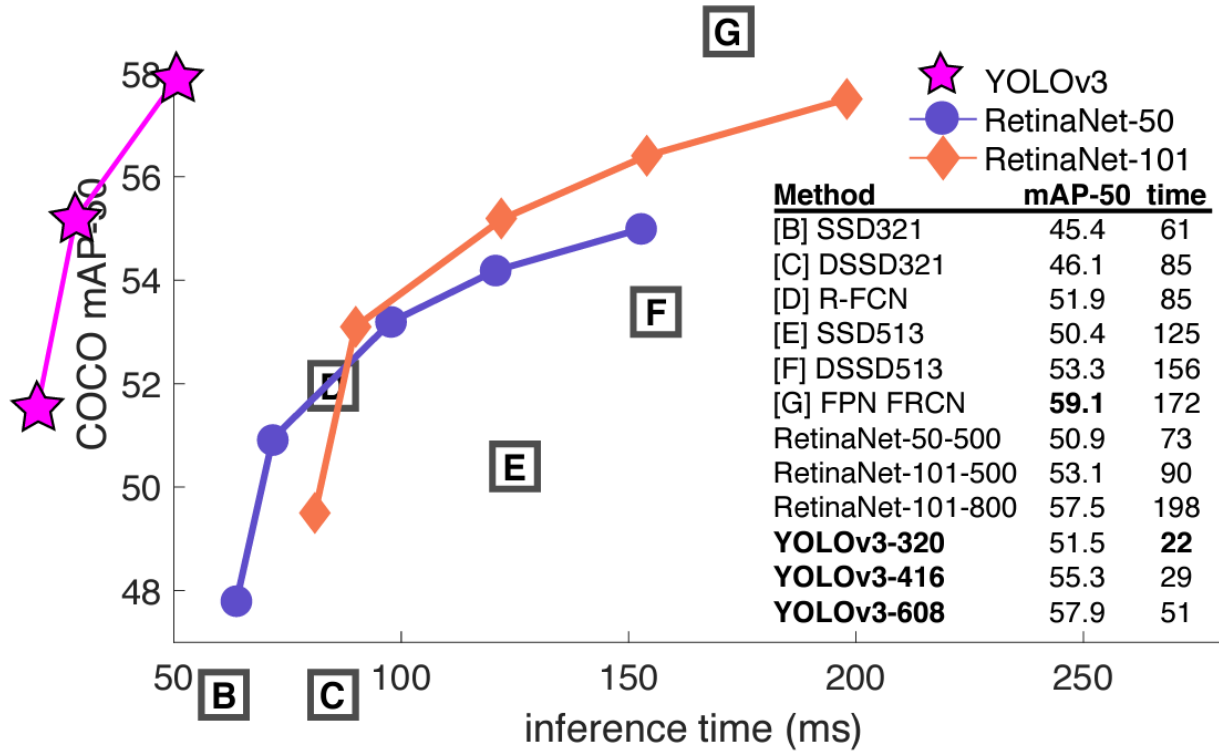


الشكل (9): يوضح تطبيق نواة 1×1 لاكتشاف الكلب في الصورة

يتم تكرار نفس الإجراء ومع ذلك قبل إعادة التوجيه إلى طبقتين من طبقات الكشف الأخرى لعمل تنبؤات يتم تكبير خرائط الميزات في الطبقة 79 و 91، وبعد الاختزال مرة أخرى بخطوة 16 و 8 يكون حجم خريطة الميزات 26×26 و 52×52 المطابق لطبقة الكشف 94 و 106، علاوة على ذلك تساعد عمليات الكشف في طبقات المقياس المختلفة في معالجة مشكلة اكتشاف الأجسام الصغيرة وبهذا نكون قد تلافينا هذه المشكلة التي تعاني من إصدار الخوارزمية الثاني، وتكون خريطة الميزات ذات الحجم الأكبر أكثر تفصيلاً، وبالتالي فإن طبقة الكشف واسعة النطاق (52×52) مسؤولة عن اكتشاف الأجسام الصغيرة بينما تكتشف طبقة الكشف صغيرة الحجم (13×13) الأجسام الأكبر حجماً، ومن خلال التسلسل مع الطبقات السطحية بعد الاختزال في الطبقة الأعمق (التسلسل مع الطبقة 61 قبل الوصول إلى الطبقة 91، والتسلسل مع الطبقة 36 قبل الوصول إلى الطبقة 103) فيمكن الحفاظ على الميزات الدقيقة من الطبقات السابقة التي تساعد بشكل كبير على كشف الأجسام الصغيرة.

ملاحظة: تستخدم الخوارزمية تقنية قمع الحد الأقصى (NMS (Non-Max Suppression لتقليل عدد صناديق الإحاطة المتوقعة بناء على درجة الثقة وتابع IoU لجمع مقترحات الكائنات من خلايا شبكة الصورة.

النتائج:



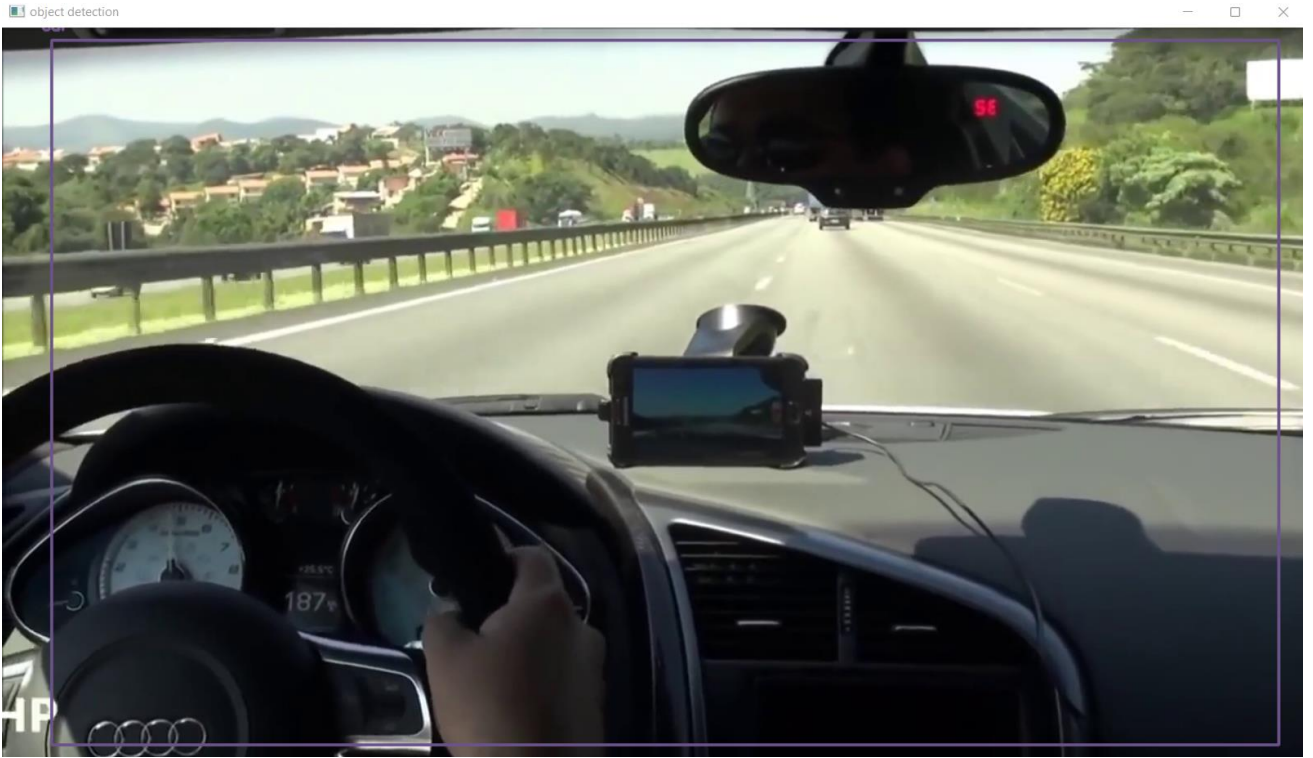
الشكل (10): نتائج المقارنة بين الخوارزمية المستخدمة وأخرى

يوضح الشكل السابق تفوق الخوارزمية المستخدمة على باقي الخوارزميات في المقارنة بناء على زمن التنفيذ ومعدل أداء التوقع (mean Accuracy Precision) mAP لنصف صورة مدروسة بناء على مجموعة بيانات تدعى MSCOCO (MicroSoft Common Object in COntext).

	backbone	AP	AP ₅₀	AP ₇₅	AP _S	AP _M	AP _L
<i>Two-stage methods</i>							
Faster R-CNN+++ [5]	ResNet-101-C4	34.9	55.7	37.4	15.6	38.7	50.9
Faster R-CNN w FPN [8]	ResNet-101-FPN	36.2	59.1	39.0	18.2	39.0	48.2
Faster R-CNN by G-RMI [6]	Inception-ResNet-v2 [21]	34.7	55.5	36.7	13.5	38.1	52.0
Faster R-CNN w TDM [20]	Inception-ResNet-v2-TDM	36.8	57.7	39.2	16.2	39.8	52.1
<i>One-stage methods</i>							
YOLOv2 [15]	DarkNet-19 [15]	21.6	44.0	19.2	5.0	22.4	35.5
SSD513 [11, 3]	ResNet-101-SSD	31.2	50.4	33.3	10.2	34.5	49.8
DSSD513 [3]	ResNet-101-DSSD	33.2	53.3	35.2	13.0	35.4	51.1
RetinaNet [9]	ResNet-101-FPN	39.1	59.1	42.3	21.8	42.7	50.2
RetinaNet [9]	ResNeXt-101-FPN	40.8	61.1	44.1	24.1	44.2	51.2
YOLOv3 608 × 608	Darknet-53	33.0	57.9	34.4	18.3	35.4	41.9

الشكل (11): نتائج المقارنة مع الخوارزمية المستخدمة وأخرى ونماذج العنق

يوضح الشكل السابق مقارنة بين خوارزميات الخطوة الواحدة في الكشف مثل خوارزمية YOLO vx وخوارزمية SSD وخوارزميات مرحلتي الكشف مثل شبكات CNN, R-CNN, Fast R-CNN, Faster R-CNN بمعايير قياس AP, AP50, AP75, APS, APM, APL وهي معايير قياس دقة النموذج لكامل الصورة ومنتصف الصورة وثلاثة أرباع الصورة وصورة بحجم صغير (أقل من $32 * 32 = 1,024$) وصورة بحجم متوسط (أكثر من 1,024 وأقل من $96 * 96 = 9,216$) وصورة كبيرة الحجم (أكثر من 9,216) على التوالي. حقق التنفيذ نتائج جيدة ولكن غير مرضية بالمقارنة مع النتائج المطروحة في الورقة البحثية للمؤلف ويرجع ذلك إلى ضعف الموارد الحاسوبية التي تعالج الخوارزمية في الزمن الحقيقي على مقطع فيديو على منصة يوتيوب كما يظهر تالياً.



الخاتمة:

تعد مشكلة تتبع الأشياء تقليدية في مجال الرؤية الحاسوبية، وقد عمد الباحثون إلى تطوير خوارزميات سريعة ودقيقة قادرة على التنبؤ بمواقع الكائنات بسرعات عالية، وقد حققت خوارزمية YOLO في هذا المجال تطوراً ملحوظاً بمقارنتها مع عدد من الخوارزميات المستخدمة، وتم ترقية هذه الخوارزمية إلى عدة إصدارات وصولاً إلى الإصدار السابع الذي صدر في أواخر حزيران عام 2022، بحيث قدم كل جيل من الخوارزميات دقة أكبر وسرعة أعلى من سابقه، كما تلافى مشاكل الجيل السابق بتقنيات حديثة لتصبح الخوارزمية رائدة في هذا المجال.

المراجع:

<https://pjreddie.com/media/files/papers/YOLOv3.pdf>

<https://github.com/arunponnusamy/object-detection-opencv>

<https://towardsdatascience.com/understanding-fast-r-cnn-and-faster-r-cnn-for-object-detection-adbb55653d97>

[https://medium.com/@smallfishbigsea/understand-ssd-and-implement-your-own-
caa3232cd6ad#:~:text=SSD%20\(Single%20Shot%20MultiBox%20Detector,generally
%20faster%20than%20Faster%20RCNN](https://medium.com/@smallfishbigsea/understand-ssd-and-implement-your-own-caa3232cd6ad#:~:text=SSD%20(Single%20Shot%20MultiBox%20Detector,generally%20faster%20than%20Faster%20RCNN)

الملحق:

```
import cv2
import numpy as np
import pafy                                # pafy lib for reading videos from the youtube
```

```
#setup running settings and set the dimensions of the video
url = 'https://www.youtube.com/watch?v=cYtg4L8ul7w'
vPafy = pafy.new(url)
play = vPafy.getbest(preftype="mp4")
cap = cv2.VideoCapture(play.url)
cap.set(3, 480)
cap.set(4, 640)
```

```
#import the labels, weights and cofiguration
classesf = "yolov3.txt"
weights = "yolov3.weights"
config = "yolov3.cfg"
scale = 0.00392
classes = None
with open (classesf, 'r') as f:
    classes = [line.strip() for line in f.readlines()]
COLORS = np.random.uniform(0, 255, size=(len(classes), 3))
net = cv2.dnn.readNet(weights, config)
```

```
#setup the layers of output
def get_output_layers(net):
    layer_names = net.getLayerNames()
    output_layers = [layer_names[i - 1] for i in net.getUnconnectedOutLayers()]
    return output_layers
```

```

#draw the prediction bounding boxes and setup it with colors and labels
def draw_prediction(img, class_id, confidence, x, y, x_plus_w, y_plus_h):
    label = str(classes[class_id])
    color = COLORS[class_id]
    cv2.rectangle(img, (x,y), (x_plus_w,y_plus_h), color, 2)
    cv2.putText(img,label,(x-10,y-10),cv2.FONT_HERSHEY_SIMPLEX,0.5, color, 2)

#setup the read video and detection with network
def video_detector():
    while True:
        ret, image = cap.read()
        Width = image.shape[1]
        Height = image.shape[0]
        blob = cv2.dnn.blobFromImage(image, scale, (416,416), (0,0,0), True, crop=False)
        net.setInput(blob)
        outs = net.forward(get_output_layers(net))
        class_ids = []
        confidences = []
        boxes = []
        conf_threshold = 0.5
        nms_threshold = 0.4
        for out in outs:
            for detection in out:
                scores = detection[5:]
                class_id = np.argmax(scores)
                confidence = scores[class_id]
                if confidence > 0.5:
                    center_x = int (detection[0] * Width)
                    center_y = int (detection[1] * Height)
                    w = int(detection[2] * Width)
                    h = int(detection[3] * Height)
                    x = center_x - w/2
                    y = center_y - h/2
                    class_ids.append(class_id)
                    confidences.append(float(confidence))
                    boxes.append([x, y, w, h])

#reduce the boxes by NMS function
indices = cv2.dnn.NMSBoxes(boxes, confidences, conf_threshold, nms_threshold)
for i in indices:
    i = 0
    box = boxes[i]
    x = box[0]

```



```
y = box[1]
w = box[2]
h = box[3]
draw_prediction(image, class_ids[i], confidences[i], round(x), round(y),
round(x+w), round (y+h))

cv2.imshow("object detection", image)
if cv2.waitKey(1) & 0xFF == ord('q'):
    break
if __name__ == "__main__":
    video_detector()
```