Mansoura University
Faculty of Computers and Information
Computer Science Department

Scalable and Efficient Object Detection

Pre-Master Project

Department of Computer Science,
Faculty of Computers and Information
Mansoura University

By

Mohamed Abdo Mohamed Abdelaziz Elnashar

Supervised by

Prof. Samir Elmougy

Department of Computer Science

Faculty of Computers and Information

Mansoura University

Dr. Mohamed haggag

Department of Computer Science

Faculty of Computers and Information

Mansoura University

2019/2020

contents

Acknowledgements

First and foremost, we would like to thank the chairman of our faculty.
we would also like to express our gratitude and appreciation to:

Prof. Samir Elmougy

Dr. Mohamed haggag

for all the help and guidance, they provided throughout our education.

This project could not be done without them, who not only served as our supervisor but also encouraged us.
so, we thank them and also, we thank the entire staff of the department

dedication

We proudly dedicate this project and our efforts to our families, friends and our supervisor who made this accomplishment possible.

Abstract

Model efficiency has become increasingly important in computer vision, we systematically study various neural network architecture design choices for object detection and propose several key optimizations to improve efficiency. First, we propose a weighted bi-directional feature pyramid network (BiFPN), which allows easy and fast multi-scale feature fusion; Second, we propose a compound scaling method that uniformly scales the resolution, depth, and width for all backbone, feature network, and box/class prediction networks at the same time. Based on these optimizations, we have developed a new family of object detectors, called Efficient Det, which consistently achieve an order-of-magnitude better efficiency than prior art across a wide spectrum of resource constraints. In particular, without bells and whistles, our EfficientDet-D7 achieves 51.0 mAP in COCO dataset with 52M parameters and 326B FLOPS1 , being 4x smaller and using 9.3x fewer FLOPS yet still more accurate (+0.3% mAP) than the best previous detector.

Abbreviations

COCO	**Common objects in context**
BiFPN	**Bi-directional feature pyramid network**
NASFPN	**Learning Scalable Feature Pyramid Architecture**
FLOPS	**FLoating-point Operations Per Second**
MAP	**Mean Average Precision**

List of Figures

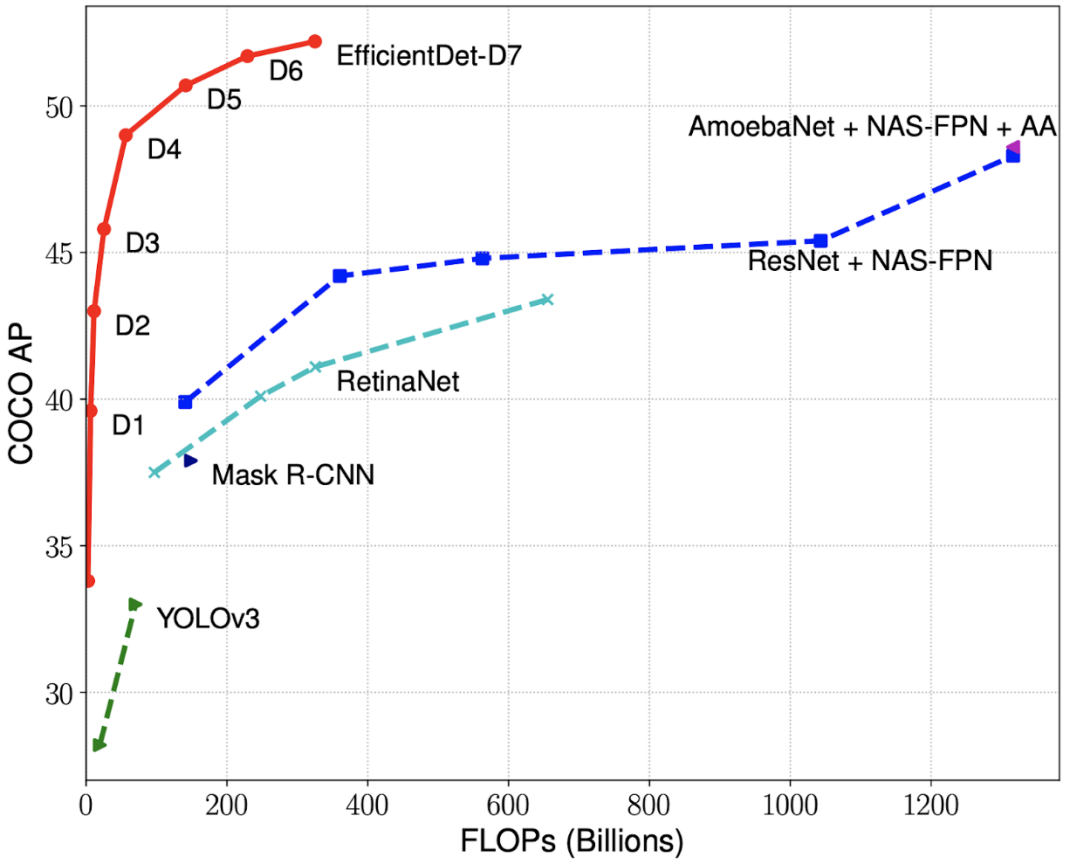
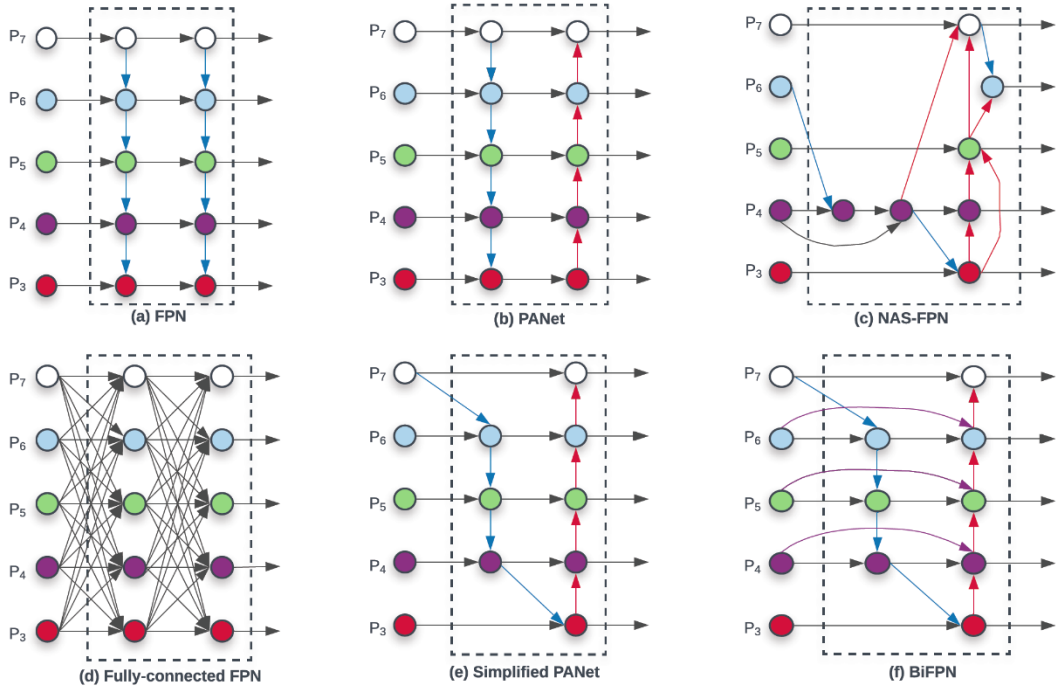


Figure 1: Model FLOPS vs COCO accuracy

All numbers are for single-model single-scale. Our EfficientDet achieves much better accuracy with fewer computations than other detectors. In particular, EfficientDet-D7 achieves new state-of-the-art 51.0% COCO mAP with 4x fewer parameters and 9.3x fewer FLOPS.

Figure 2: Feature network design



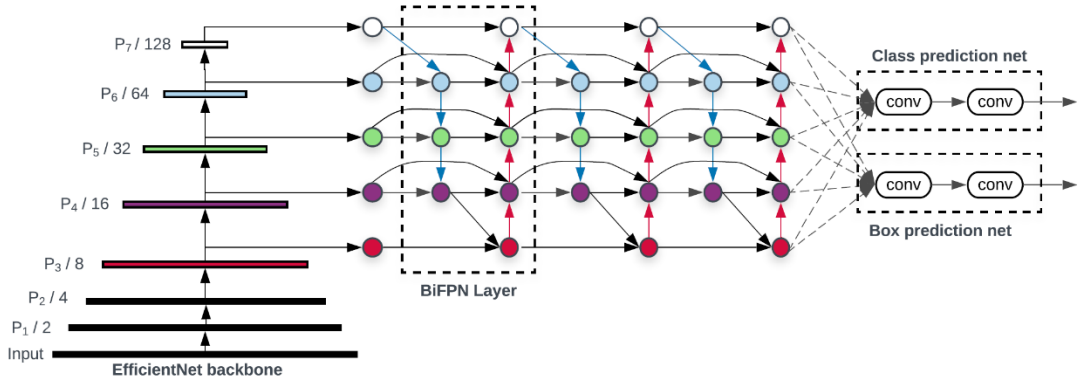
(a) FPN introduces a top-down pathway to fuse multi-scale features from level 3 to 7 (P3 - P7);

(b) PANet adds an additional bottom-up pathway on top of FPN;

(c) NAS-FPN use neural architecture search to find an irregular feature network topology.

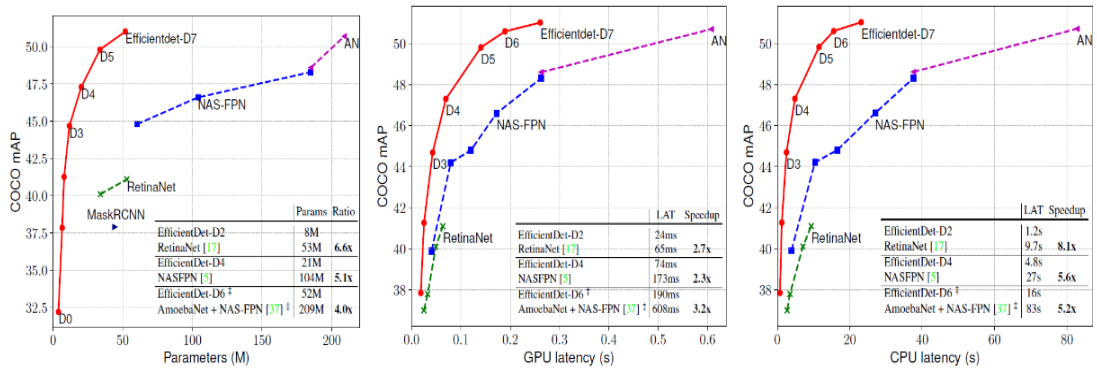
(d) adds expensive connections from all input feature to output features; **(e)** simplifies PANet by removing nodes if they only have one input edge; **(f)** is our BiFPN with better accuracy and efficiency trade-offs.

Figure 3: EfficientDet architecture



It employs EfficientNet as the backbone network, BiFPN as the feature network, and shared class/box prediction network. Both BiFPN layers and class/box net layers are repeated multiple times based on different resource constraints .

Figure 4: Model size and inference latency comparison



(a) Model Size (b) GPU Latency (c) CPU Latency

Latency is measured with batch size 1 on the same machine equipped with a Titan V GPU and Xeon CPU. AN denotes AmoebaNet + NAS-FPN trained with auto-augmentation. Our EfficientDet models are 4x - 6.6x smaller, 2.3x - 3.2x faster on GPU, and 5.2x - 8.1x faster on CPU than other detectors .

List of Tables

Tables of Contents

Table 1: Scaling configs for EfficientDet D0-D

	Input size R_{input}	Backbone Network	BiFPN #channels W_{bifpn}	BiFPN #layers D_{bifpn}	Box/class #layers D_{class}
D0 ($\phi = 0$)	512	B0	64	2	3
D1 ($\phi = 1$)	640	B1	88	3	3
D2 ($\phi = 2$)	768	B2	112	4	3
D3 ($\phi = 3$)	896	B3	160	5	4
D4 ($\phi = 4$)	1024	B4	224	6	4
D5 ($\phi = 5$)	1280	B5	288	7	4
D6 ($\phi = 6$)	1408	B6	384	8	5
D7	1536	B6	384	8	5

φ is the compound coefficient that controls all other scaling dimensions; BiFPN, box/class net, and input size are scaled up using equation 1, 2, 3 respectively. D7 has the same settings as D6 except using larger input size.

Table 2: EfficientDet performance on COCO

Model	mAP	#Params	Ratio	#FLOPS	Ratio	GPU LAT(ms)	Speedup	CPU LAT(s)	Speedup
EfficientDet-D0	**32.4**	**3.9M**	**1x**	**2.5B**	**1x**	**16** ±1.6	**1x**	**0.32** ±0.002	**1x**
YOLOv3 [26]	33.0	-	-	71B	28x	51†	-	-	-
EfficientDet-D1	**38.3**	**6.6M**	**1x**	**6B**	**1x**	**20** ±1.1	**1x**	**0.74** ±0.003	**1x**
MaskRCNN [8]	37.9	44.4M	6.7x	149B	25x	92†	-	-	-
RetinaNet-R50 (640) [17]	37.0	34.0M	6.7x	97B	16x	27 ±1.1	1.4x	2.8 ±0.017	3.8x
RetinaNet-R101 (640) [17]	37.9	53.0M	8x	127B	21x	34 ±0.5	1.7x	3.6 ±0.012	4.9x
EfficientDet-D2	**41.1**	**8.1M**	**1x**	**11B**	**1x**	**24** ±0.5	**1x**	**1.2** ±0.003	**1x**
RetinaNet-R50 (1024) [17]	40.1	34.0M	4.3x	248B	23x	51 ±0.9	2.0x	7.5 ±0.006	6.3x
RetinaNet-R101 (1024) [17]	41.1	53.0M	6.6x	326B	30x	65 ±0.4	2.7x	9.7 ±0.038	8.1x
NAS-FPN R-50 (640) [5]	39.9	60.3M	7.5x	141B	13x	41 ±0.6	1.7x	4.1 ±0.027	3.4x
EfficientDet-D3	**44.3**	**12.0M**	**1x**	**25B**	**1x**	**42** ±0.8	**1x**	**2.5** ±0.002	**1x**
NAS-FPN R-50 (1024) [5]	44.2	60.3M	5.1x	360B	15x	79 ±0.3	1.9x	11 ±0.063	4.4x
NAS-FPN R-50 (1280) [5]	44.8	60.3M	5.1x	563B	23x	119 ±0.9	2.8x	17 ±0.150	6.8x
EfficientDet-D4	**46.6**	**20.7M**	**1x**	**55B**	**1x**	**74** ±0.5	**1x**	**4.8** ±0.003	**1x**
NAS-FPN R50 (1280@384)	45.4	104 M	5.1x	1043B	19x	173 ±0.7	2.3x	27 ±0.056	5.6x
EfficientDet-D5 + AA	**49.8**	**33.7M**	**1x**	**136B**	**1x**	**141** ±2.1	**1x**	**11** ±0.002	**1x**
AmoebaNet+ NAS-FPN + AA(1280) [37]	48.6	185M	5.5x	1317B	9.7x	259 ±1.2	1.8x	38 ±0.084	3.5x
EfficientDet-D6 + AA	**50.6**	**51.9M**	**1x**	**227B**	**1x**	**190** ±1.1	**1x**	**16**±0.003	**1x**
AmoebaNet+ NAS-FPN + AA(1536) [37]	50.7	209M	4.0x	3045B	13x	608 ±1.4	3.2x	83 ±0.092	5.2x
EfficientDet-D7 + AA	**51.0**	**51.9M**	**1x**	**326B**	**1x**	**262** ±2.2	**1x**	**24** ±0.003	**1x**

Results are for single-model single-scale. Params and FLOPS denote the number of parameters and multiply-adds. LAT denotes inference latency with batch size 1.

AA denotes autoaugmentation. We group models together if they have similar accuracy, and compare the ratio or speedup between EfficientDet and other detectors in each group.

Table 3: Disentangling backbone and BiFPN

	mAP	Parameters	FLOPS
ResNet50 + FPN	37.0	34M	97B
EfficientNet-B3 + FPN	40.3	21M	75B
EfficientNet-B3 + BiFPN	44.4	12M	24B

Starting from the standard RetinaNet (ResNet50+FPN), we first replace the backbone with EfficientNet-B3, and then replace the baseline FPN with our proposed BiFPN.

Table 4: Comparison of different feature networks

	mAP	#Params ratio	#FLOPS ratio
Top-Down FPN [16]	42.29	1.0x	1.0x
Repeated PANet [19]	44.08	1.0x	1.0x
NAS-FPN [5]	43.16	0.71x	0.72x
Fully-Connected FPN	43.06	1.24x	1.21x
BiFPN (w/o weighted)	**43.94**	**0.88x**	**0.67x**
BiFPN (w/ weighted)	**44.39**	**0.88x**	**0.68x**

Our weighted BiFPN achieves the best accuracy with fewer parameters and FLOPS.

Chapter 1

Chapter -1-

INTRODUCTION

1.1. Introduction

Tremendous progresses have been made in recent years towards more accurate object detection meanwhile, object detectors also become increasingly more expensive. For example, the latest AmoebaNet-based NASFPN detector [37] requires 167M parameters and 3045B FLOPS (30x more than RetinaNet [17]) to achieve state of-the-art accuracy. The large model sizes and expensive computation costs deter their deployment in many real-world applications such as robotics and self-driving cars where model size and latency are highly constrained. Given these real-world resource constraints, model efficiency becomes increasingly important for object detection. There have been many previous works aiming to develop more efficient detector architectures.

1.2. Motivation

build a scalable detection architecture with both higher accuracy and better efficiency across a wide spectrum of resource constraints.

1.3. Objectives

studying various design choices of detector architectures. Based on the one-stage detector paradigm, we examine the design choices for backbone, feature fusion, and class/box network.

1.4. Challenges

Challenge 1: efficient multi-scale feature fusion.

Challenge 2: model scaling

Chapter 2

Chapter -2-

Related Works

2.1 One-Stage Detectors:

Existing object detectors are mostly categorized by whether they have a region-of-interest proposal While two-stage detectors tend to be more flexible and more accurate, one-stage detectors are often considered to be simpler and more efficient by leveraging predefined anchors. Recently, one-stage detectors have attracted substantial attention due to their efficiency and simplicity. we mainly follow the one-stage detector design, and we show it is possible to achieve both better efficiency and higher accuracy with optimized network architectures.

2.2 Multi-Scale Feature Representations:

One of the main difficulties in object detection is to effectively represent and process multi-scale features. Earlier detectors often directly perform predictions based on the pyramidal feature hierarchy extracted from backbone networks. As one of the pioneering works, feature pyramid network (FPN) proposes a top-down pathway to combine multi-scale features. Following this idea, PANet adds an extra bottom-up path aggregation network on top of FPN, STDL proposes a scale-transfer module to exploit cross-scale features; M2det proposes a U-shape module to fuse multi-scale features, and G-FRNet introduces gate units for controlling information flow across features. More recently, NAS-FPN leverages neural architecture search to automatically design feature network topology. Although it achieves better performance, NAS-FPN requires thousands of GPU hours during the search, and the resulting feature network is irregular and thus difficult to interpret. we aim to optimize multi-scale feature fusion with a more intuitive and principled way.

2.3 Model Scaling:

To obtain better accuracy, it is common to scale up a baseline detector by employing bigger backbone networks (e.g., from mobile-size models and ResNet to ResNeXt and AmoebaNet), or increasing input image size (e.g., from 512x512 to 1536x1536). Some recent works show that increasing the channel size and repeating feature networks can also lead to higher accuracy. These scaling methods mostly focus on single or limited scaling dimensions. Recently, demonstrates remarkable model efficiency for image classification by jointly scaling up network width, depth, and resolution. Our proposed compound scaling method for object detection is mostly inspired.

Chapter 3

Chapter -3-

THE PROPOSED SYSTEM

Our contributions can be summarized as:

• **We proposed BiFPN**, a weighted bidirectional feature network for easy and fast multi-scale feature fusion.

• **We proposed a new compound scaling method**, which jointly scales up backbone, feature network, box/class network, and resolution, in a principled way.

• **Based on BiFPN and compound scaling**, we developed EfficientDet, a new family of detectors with significantly better accuracy and efficiency across a wide spectrum of resource constraints.

We evaluate EfficientDet on COCO 2017 detection datasets. Each model is trained using SGD optimizer with momentum 0.9 and weight decay 4e-5. Learning rate is first linearly increased from 0 to 0.08 in the initial 5% warm-up training steps and then annealed down using cosine decay rule. Batch normalization is added after every convolution with batch norm decay 0.997 and epsilon 1e-4.

We use exponential moving average with decay 0.9998. We also employ commonly-used focal loss with $\alpha = 0.25$ and $\gamma = 1.5$, and aspect ratio $\{1/2, 1, 2\}$.

Our models are trained with batch size 128 on 32 TPUv3 chips. We use RetinaNet preprocessing for EfficientDet-D0/D1/D3/D4, but for fair comparison, we use the same auto-augmentation for EfficientDet-D5/D6/D7 when comparing with the prior art of AmoebaNet-based NAS-FPN detectors.

Table 2 compares EfficientDet with other object detectors, under the single-model single-scale settings with no test-time augmentation. Our EfficientDet achieves better accuracy and efficiency than previous detectors across a wide range of accuracy or resource constraints. On relatively low-accuracy regime, our EfficientDet-D0 achieves similar accuracy as YOLOv3 with 28x fewer FLOPS.

Compared to RetinaNet and Mask-RCNN, our EfficientDet-D1 achieves similar accuracy with up to 8x fewer parameters and 25x fewer FLOPS. On high-accuracy regime, our EfficientDet also consistently outperforms recent NAS-FPN and its enhanced versions in with an order-of-magnitude fewer parameters and FLOPS. In particular, our EfficientDet-D7 achieves a new state-of-the-art 51.0 mAP for single-model single-scale, while still being 4x smaller and using 9.3x fewer FLOPS than previous best results. Notably, unlike the large AmoebaNet-based NAS-FPN models that require special settings (e.g., change anchors from 3x3 to 9x9 and train with model parallelism), all EfficientDet models use the same 3x3 anchors and trained without model parallelism. In addition to parameter size and FLOPS, we have also compared the real-world latency on Titan-V GPU and single-thread Xeon CPU.

We run each model 10 times with batch size 1 and report the mean and standard deviation. Figure 4 illustrates the comparison on model size, GPU latency, and single-thread CPU latency. For fair comparison, these figures only include results that are measured on the same machine. Compared to previous detectors, EfficientDet models are up to 3.2x faster on GPU and 8.1x faster on CPU, suggesting they are efficient on real-world hardware.

Box/class prediction network

we fix their width to be always the same as BiFPN (i.e., Wpred = Wbifpn), but linearly increase the depth (#layers) using equation: $D_{box} = D_{class} = 3 + \lfloor \varphi/3 \rfloor$.

Input image resolution – Since feature level 3-7 are used in BiFPN, the input resolution must be dividable by $2^7 = 128$, so we linearly increase resolutions using equation: $R_{input} = 512 + \varphi \cdot 12$.

Chapter 4

Chapter -4-

CONCOLUSIONS and Future Works

4.1 Conclusions

we systematically study various network architecture design choices for efficient object detection, and propose a weighted bidirectional feature network and a customized compound scaling method, to improve accuracy and efficiency. Based on these optimizations, we have developed a new family of detectors, named EfficientDet, which consistently achieve better accuracy and efficiency than the prior art across a wide spectrum of resource constraints. In particular, our EfficientDet-D7 achieves state-of-the-art accuracy with an order-of-magnitude fewer parameters and FLOPS than the best existing detector. Our EfficientDet is also up to 3.2x faster on GPUs and 8.1x faster on CPUs. Code will be made public.

4.2 Future works

Challenge 1: more efficient multi-scale feature fusion.

Challenge 2: model scaling

References

[1] Md Amirul Islam, Mrigank Rochan, Neil DB Bruce, and Yang Wang. Gated feedback refinement network for dense image labeling.

[2] Zhaowei Cai, Quanfu Fan, Rogerio S Feris, and Nuno Vasconcelos. A unified multi-scale deep convolutional neural network for fast object detection.

[3] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection.

[4] Franc¸ois Chollet. Xception: Deep learning with depthwise separable convolutions.

[5] Golnaz Ghiasi, Tsung-Yi Lin, Ruoming Pang, and Quoc V. Le. Nas-fpn: Learning scalable feature pyramid architecture for object detection.

[6] Kaiming He, Ross Girshick, and Piotr Dollar. Rethinking ´ imagenet pre-training.

[7] Kaiming He, Georgia Gkioxari, Piotr Dollar, and Ross Gir- ´ shick. Mask r-cnn.

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition.

[9] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, Quoc V. Le, and Hartwig Adam. Searching for mobilenetv3.

[10] Jonathan Huang, Vivek Rathod, Chen Sun, Menglong Zhu, Anoop Korattikara, Alireza Fathi, Ian Fischer, Zbigniew Wojna, Yang Song, Sergio Guadarrama, et al. Speed/accuracy trade-offs for modern convolutional object detectors.

[11] Seung-Wook Kim, Hyong-Keun Kook, Jee-Young Sun, Mun-Cheon Kang, and Sung-Jea Ko. Parallel feature pyramid network for object detection.

[12] Tao Kong, Fuchun Sun, Chuanqi Tan, Huaping Liu, and Wenbing Huang. Deep feature pyramid reconfiguration for object detection.

[13] Hei Law and Jia Deng. Cornernet: Detecting objects as paired keypoints.

[14] Hanchao Li, Pengfei Xiong, Jie An, and Lingxue Wang. Pyramid attention networks.

[15] Tsung-Yi Lin, Piotr Dollar, Ross Girshick, Kaiming He, ´ Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection.

التحسين في التعرف علي الاشياء

تحسين التعرف علي الاشياء من حيث السرعة والحجم

تقدم نتائج PANet مبنى على ان ال Feature Pyramid Network او ال FPNالفكرة ال اللتي neurons او **ثنائية الأتجاه** و حذف bi directional بسبب ال FPNافضل من ال تستقبل متغير واحد فقط
وهذا ادي neurons الخاص باخر input مع ال neurons الخاص باول outputتم دمج ال بشكل عشوائى network و عدم تجريب أرقام مختلفة يشكل بيها ال nodesالي تقليل عدد ال او **منهج ارشادى للتحجيم** الموضوع heuristic-based scaling approachوتم عمل model واسمه فاى نطلع شكل φعبارة عن 3 معادلات بيسمحولنا عن طريق متغير واحد ولكن اكبر او اصغر Archجديد مبنى على نفس ال

-:المشروع مكونة من اربع فصول محتوياتها موضحة على النحو التالي

الفصل الاول : يشتمل علي مقدمة عن المشروع

الفصل الثانى : يشتمل علي الاعمال السابقة

الفصل الثالث : يشتمل علي العمل المقترح

الفصل الرابع : يشتمل علي الاستنتاجات والاعمال المستقبلي

جامعـــــة المنصـــــورة
كليـــــة الحاسبات والمعلومات
قسم علوم الحاسب

التحسين في التعرف علي الاشياء

مشروع تمهيدى ماجستير علوم الحاسب-كلية الحاسبات والمعلومات- جامعة المنصورة

إسم الطالب

محمد عبده محمد عبدالعزيز النشار

تحـــت إشـــــــراف

أ.د./ سمـــير الدسوقي المـــوجي

أستاذ علوم الحاسب

قسم علوم الحاسب

كلية الحاسبات والمعلومات

جامعة المنصورة

د. / محمد حجاج

قسم علوم الحاسب

كلية الحاسبات والمعلومات

جامعة المنصورة

2020/2019