

A survey of modern deep learning based object detection models

Vision System Lab, Gyumin Park
yywnnaa@gmail.com
Feb 14, 2024



I. INTRODUCTION

Briefly explain object detection

- identifying and localizing all instances of an object (like cars, humans, street signs, etc.) within the field of view

Briefly summarizes the development of object detection

- ensemble of hand-crafted feature extractors - CNNs and deep learning

main contributions of this paper

- 1) provides an in-depth analysis of major object detectors in both categories – single and two stage detectors, take historic look at the evolution of these methods
- 2) present a detailed evaluation of the landmark backbone architectures and lightweight models (could not find any paper which provides a broad overview of both these topics)



Structure of the paper

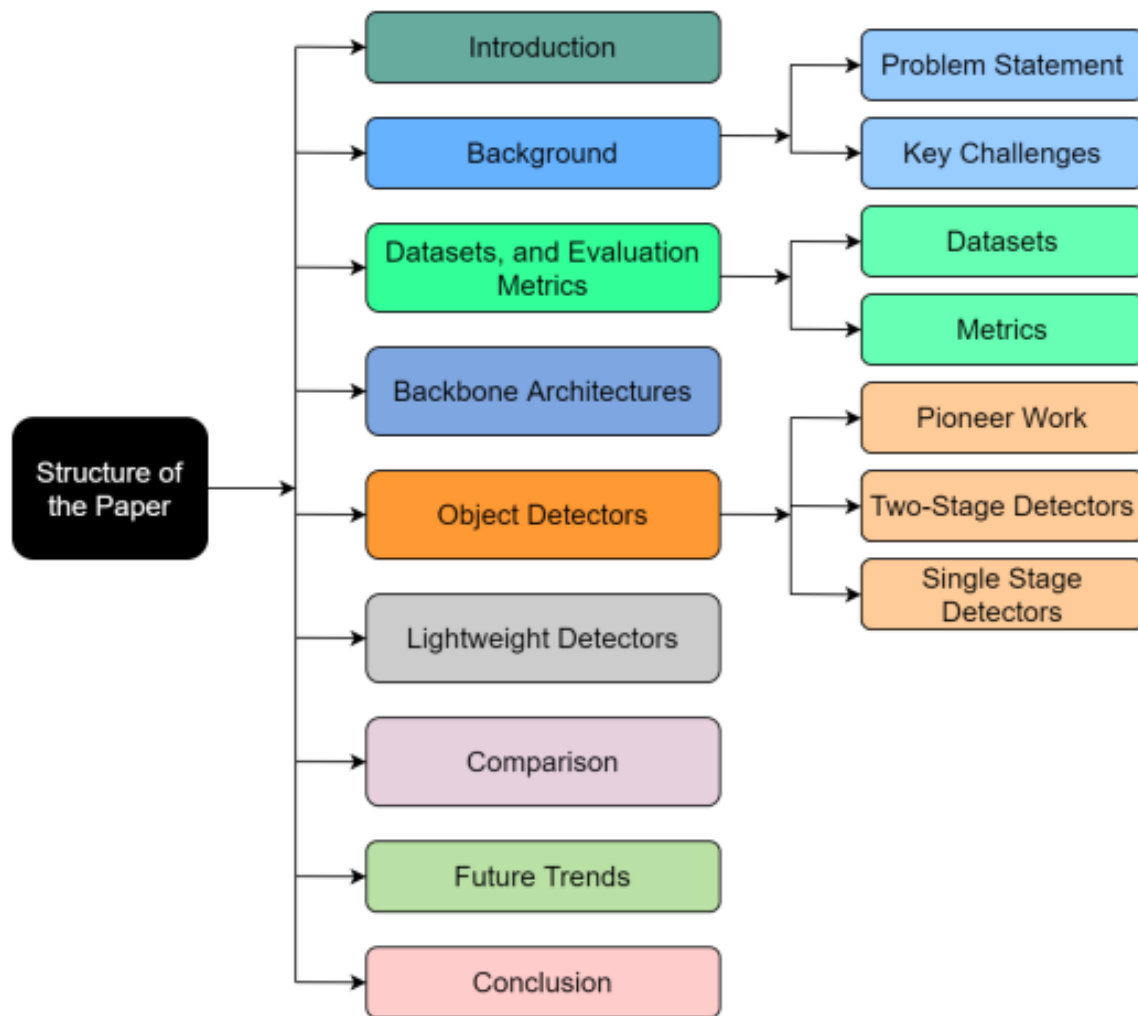


Fig. 1: Structure of the paper.



II. BACKGROUND

A. Problem Statement

- detect all instances of the predefined classes and provide its coarse localization in the image by axis-aligned boxes
- identify all instances of the object classes and draw bounding box around it
- supervised learning problem

B. Key challenges in Object Detection

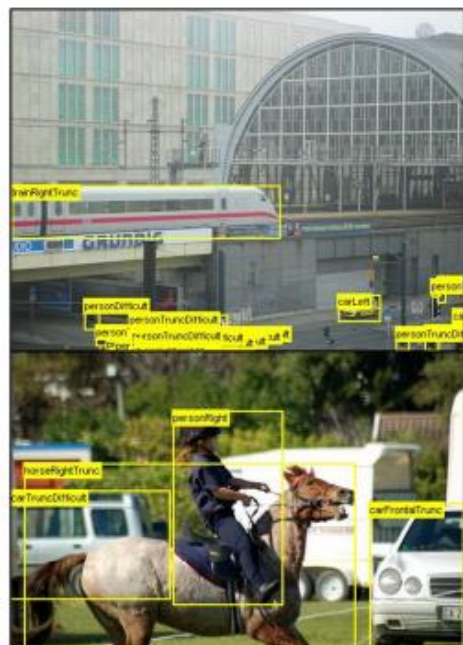
- Intra class variation : occlusion, illumination, pose, viewpoint, etc. -> class prediction fluctuates
- Number of categories :
 - sheer number of object classes available to classify
 - requires more high-quality annotated data -> hard to come by
 - using fewer examples for training a detector is an open research question
- Efficiency : need high computation resources, should also be available on mobile devices



III. DATASETS AND EVALUATION METRICS

A. Datasets

- 1) PASCAL VOC 07/12
- 2) ILSVRC
- 3) MS-COCO
- 4) Open Image



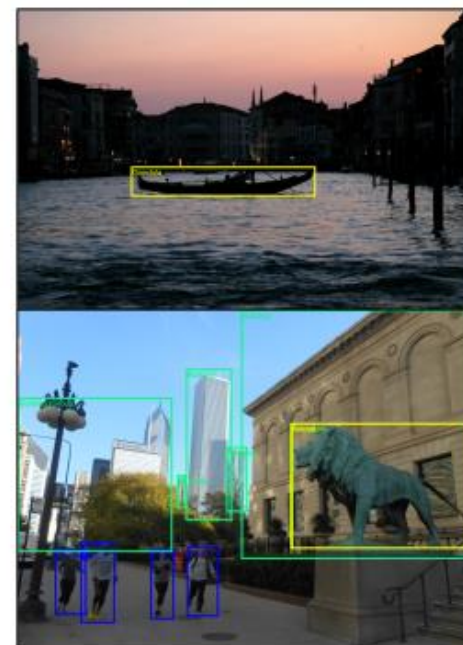
(a) PASCAL VOC 12



(b) MS-COCO



(c) ILSVRC



(d) OpenImage

Fig. 2: Sample images from different datasets.



III. DATASETS AND EVALUATION METRICS

Issues of Data Skew/Bias

the number of images for different classes vary significantly in all the datasets

a skew in the datasets - bound to create a bias in the training process of any object detection model



III. DATASETS AND EVALUATION METRICS

1) PASCAL VOC 07/12

VOC07 challenge : 5k training images and more than 12k labelled objects

VOC12 challenge : 11k training images and more than 27k labelled objects

20 categories Object classes

introduced the mean Average Precision (mAP)

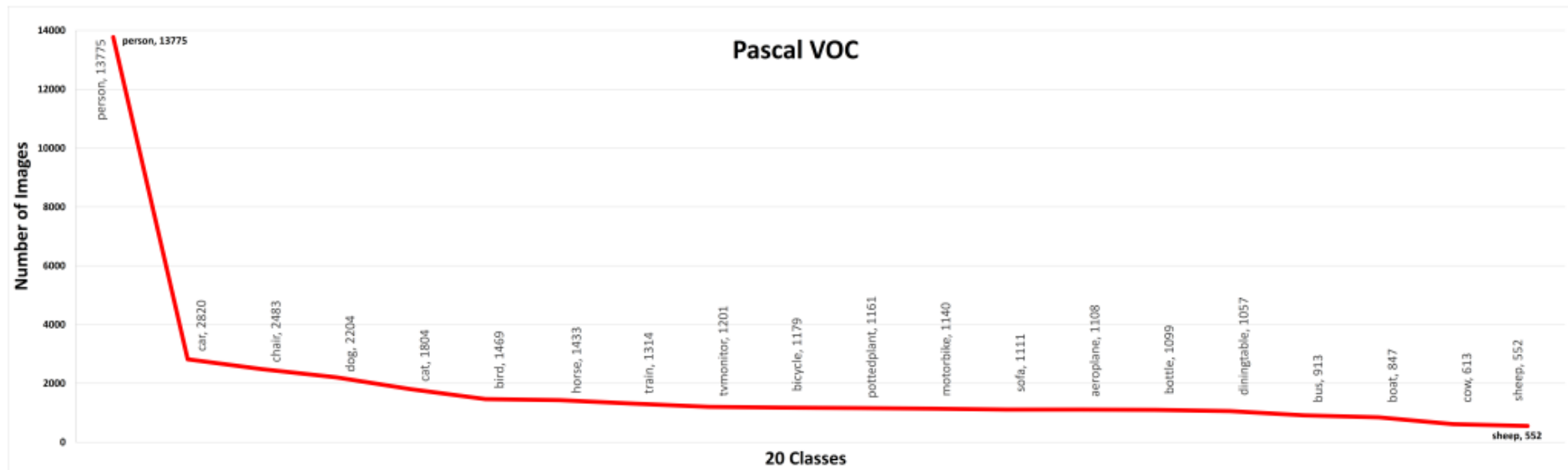


Fig. 3: (This image is best viewed in PDF form with magnification) Number of images for different classes annotated in the PascalVOC dataset [15]



III. DATASETS AND EVALUATION METRICS

2) ILSVRC

more than a million images consisting of 1000 object classification classes

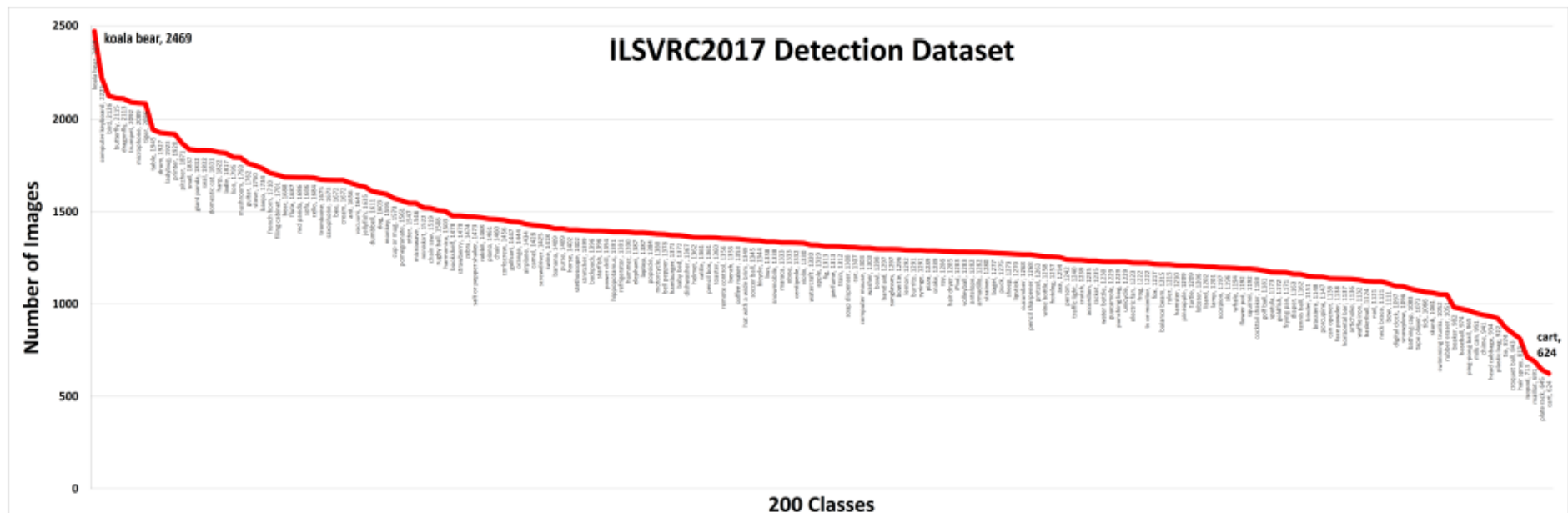


Fig. 4: (This image is best viewed in PDF form with magnification) Number of images for different classes annotated in the ImageNet dataset [15]



III. DATASETS AND EVALUATION METRICS

3) MS-COCO

91 common objects found in their natural context

more than two million instances and an average of 3.5 categories per images

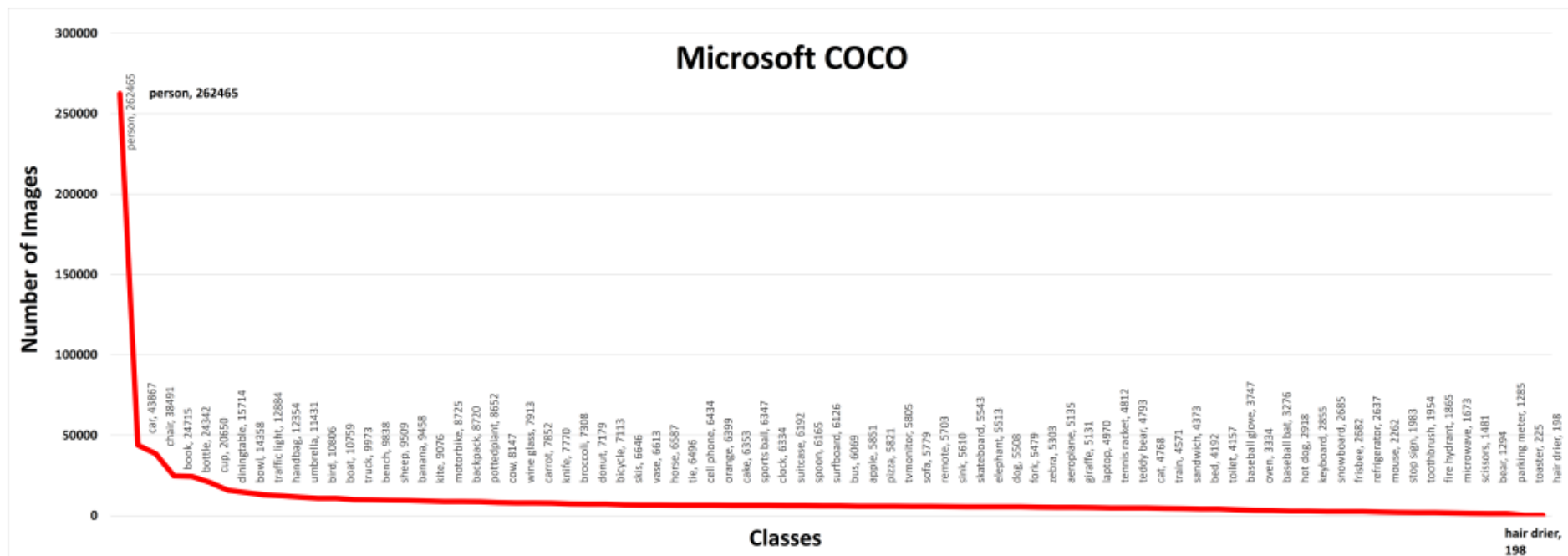


Fig. 5: (This image is best viewed in PDF form with magnification) Number of images for different classes annotated in the MS-COCO dataset [15]



III. DATASETS AND EVALUATION METRICS

4) Open Image

9.2 million images

16 million bounding boxes for 600 categories on 1.9 million images



Fig. 6: (This image is best viewed in PDF form with magnification) Number of images for different classes annotated in the Open Images dataset [15]




III. DATASETS AND EVALUATION METRICS

B. Metrics

Precision is derived from Intersection over Union (IoU),

IoU : ratio of the area of overlap and the area of union between the ground truth and the predicted bounding box

$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$




III. DATASETS AND EVALUATION METRICS

B. Metrics

Confusion Matrix

TN(True Negative, Negative Negative)

FP(False Positive, Negative Positive)

FN(False Negative, Positive Negative)

TP(True Positive, Positive Positive)

	Predicted 0	Predicted 1
Actual 0	TN	FP
Actual 1	FN	TP



III. DATASETS AND EVALUATION METRICS

B. Metrics

$$\begin{aligned} \text{Precision} &= \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}} \\ &= \frac{\text{True Positive}}{\text{All Observations}} \end{aligned} \quad (1)$$

$$= \frac{TP}{FP + TP} = \frac{\text{Pos로 예측해서 클래스까지 맞은 것들}}{\text{Pos로 예측한 것들}}$$

$$\begin{aligned} \text{Recall} &= \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}} \\ &= \frac{\text{True Positive}}{\text{All Ground Truth}} \end{aligned} \quad (2)$$

$$= \frac{TP}{FN + TP} = \frac{\text{Pos로 잘 예측한 것들}}{\text{실제 Object들}}$$



IV. BACKBONE ARCHITECTURES

- extract feature from the input image used by the model
- discussed some milestone backbone architectures used in modern detectors

A. AlexNet :

- a convolutional neural network architecture for image classification, which achieved high accuracy in the ImageNet competition
- consists of convolutional and fully connected layers, utilizing multiple convolutional kernels to extract features from images

B. VGG :

- utilizes small convolutional filters to construct networks of varying depths, achieving superior accuracy in tasks such as classification and localization.
- one of the most widely used network backbones for object classification and detection models.



IV. BACKBONE ARCHITECTURES

C. GoogLeNet/Inception :

- 22-layer deep network that utilizes Inception modules, which consist of multiple-sized filters, to efficiently process input feature maps
- introduced the concept of locally sparse connected architecture to reduce computation waste and overfitting
- achieved high accuracy on the ImageNet dataset without external data
- demonstrated the effectiveness of computation blocks in comparison to parameter-heavy networks

D. ResNets :

- residual learning to address the issue of accuracy degradation in deep neural networks.
- perform element-wise addition between input and output, ResNet mitigates the problem without adding extra parameters or computational complexity.



IV. BACKBONE ARCHITECTURES

E. ResNeXt :

- an advanced architecture derived from ResNet
- offering a simpler and more efficient model
- network is expanded and made easier to generalize
- achieves high accuracy with fewer hyperparameters

F. CSPNet :

- reduces redundant gradient information to decrease computational resources
- achieves this by creating alternate paths for gradient flow within the network
- Implementing CSPNet leads to a reduction in computational overhead by 10% to 20% while maintaining or improving accuracy.

G. EfficientNet :

- an efficient and simple architecture based on network scaling
- adopts a method of uniformly scaling depth, width, and resolution to enhance the model's performance



IV. BACKBONE ARCHITECTURES

- FLOPs : Floating Point Operations per Second

TABLE II: Comparison of Backbone architectures.

Model	Year	Layers	Parameters (Million)	Top-1 acc%	FLOPs (Billion)
AlexNet	2012	7	62.4	63.3	1.5
VGG-16	2014	16	138.4	73	15.5
GoogLeNet	2014	22	6.7	-	1.6
ResNet-50	2015	50	25.6	76	3.8
ResNeXt-50	2016	50	25	77.8	4.2
CSPResNeXt-50	2019	59	20.5	78.2	7.9
EfficientNet-B4	2019	160	19	83	4.2



V. OBJECT DETECTORS

A. Pioneer Work

- Viola-Jones: a precise and fast algorithm specialized in face detection, still widely used in small devices.
- HOG Detector: extracts features by utilizing gradients and orientations of edges for object detection, proposed specifically for pedestrian detection.
- DPM: DPM detects objects using individual object parts and achieves higher accuracy than HOG by segmenting objects into parts and generating detections through composition.



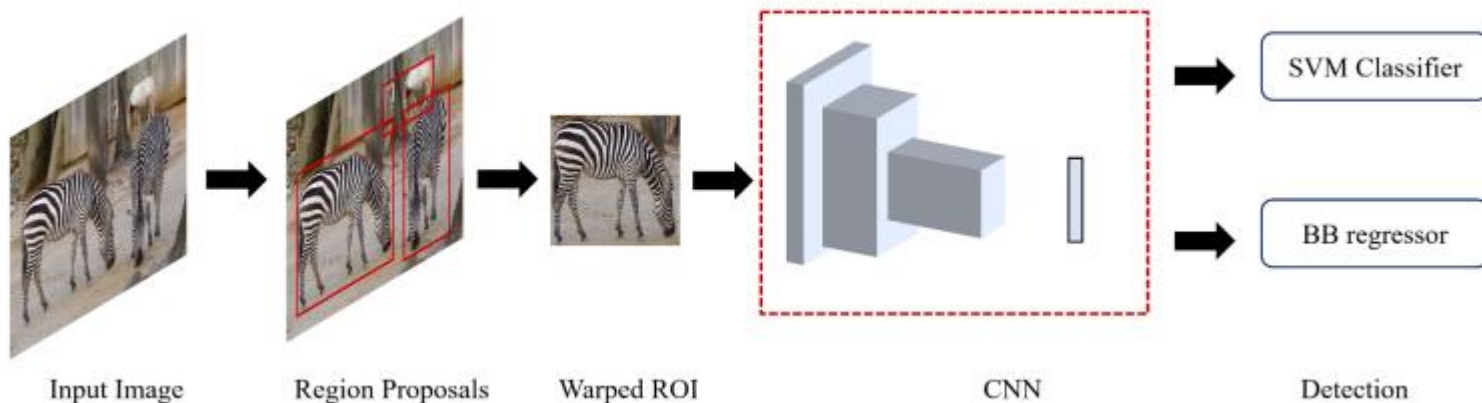
V. OBJECT DETECTORS

B. Two-Stage Detectors

1) R-CNN

- employs a region proposal module with Selective Search to generate 2000 object candidates, followed by a CNN for feature extraction
- fed into class-specific SVMs for confidence scoring, and bounding boxes are predicted using a regressor
- multistage training process was slow, complex, and resource-intensive, limiting its practicality

RCNN



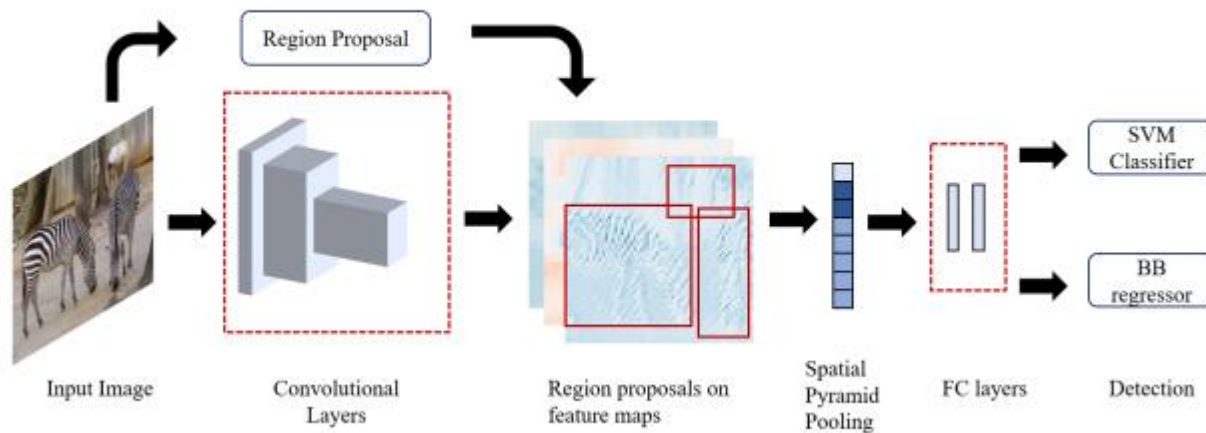


V. OBJECT DETECTORS

2) SPP-Net:

- utilizes Spatial Pyramid Pooling (SPP) layer to process images of arbitrary size or aspect ratio, making the network independent of size/aspect ratio and reducing computations
- Candidate windows are generated using the selective search algorithm, mapped onto feature maps, and converted into fixed length representations before being passed to SVM classifiers for class prediction and scoring

SPP-Net



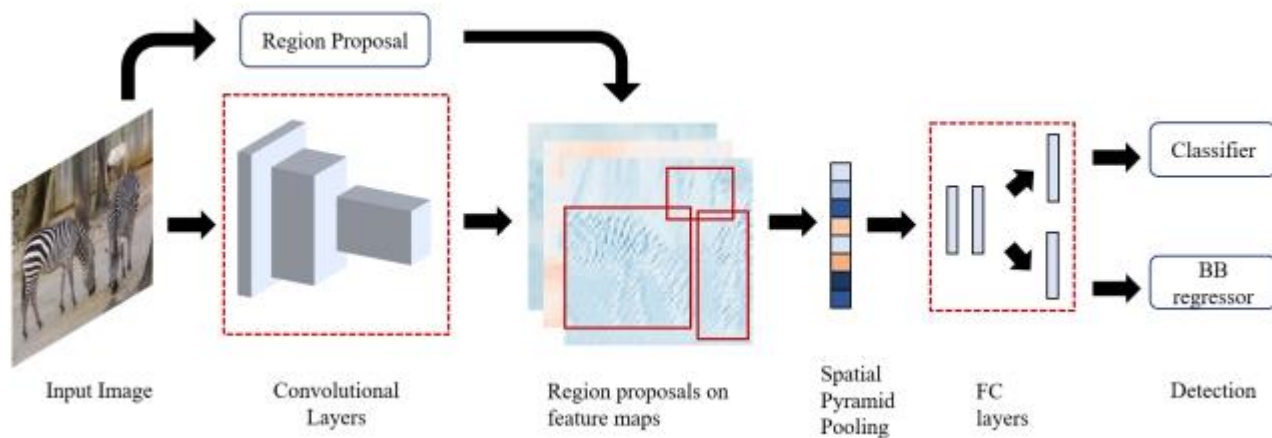


V. OBJECT DETECTORS

3) Fast R-CNN

- single end-to-end trainable system
- utilizes a RoI pooling layer to map object proposals to feature maps, simplifying the architecture and introducing a new loss function for improved performance
- significantly increased speed (146x faster than R-CNN)
- near real-time object detection

Fast RCNN

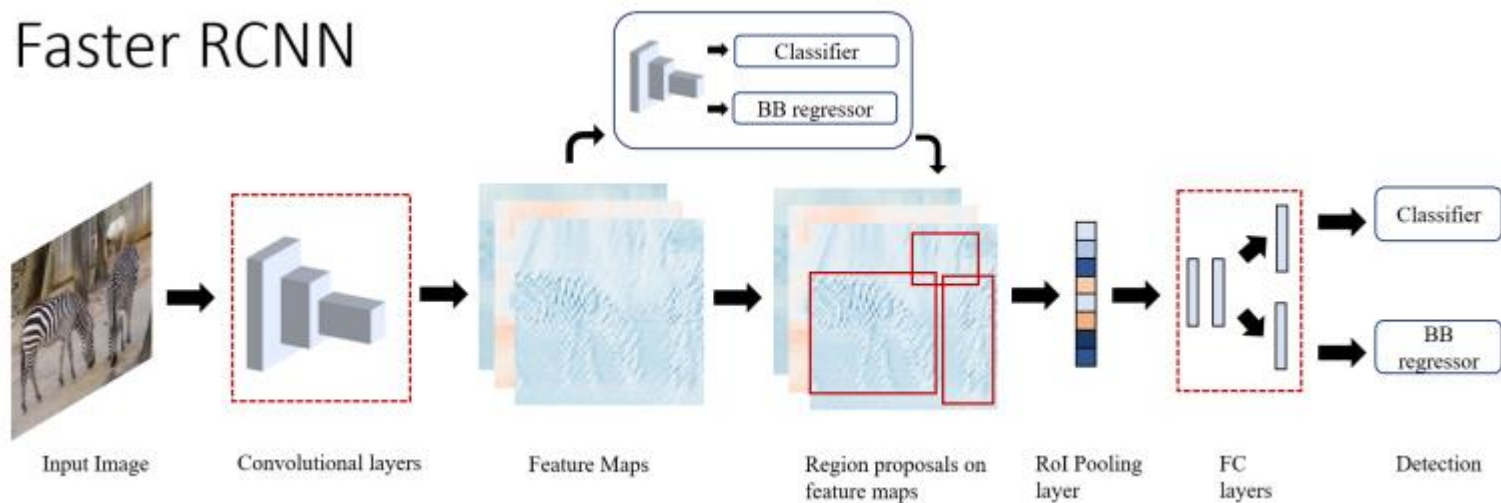




V. OBJECT DETECTORS

4) Faster R-CNN:

- RPN: fully convolutional network that takes an arbitrary input image and outputs candidate windows
- RPN introduces anchor boxes to use multiple bounding boxes with various aspect ratios and localizes objects
- Faster R-CNN improves upon the previous Fast R-CNN, achieving more than a 3% increase in detection accuracy and significantly reducing inference time, allowing it to run almost in real-time

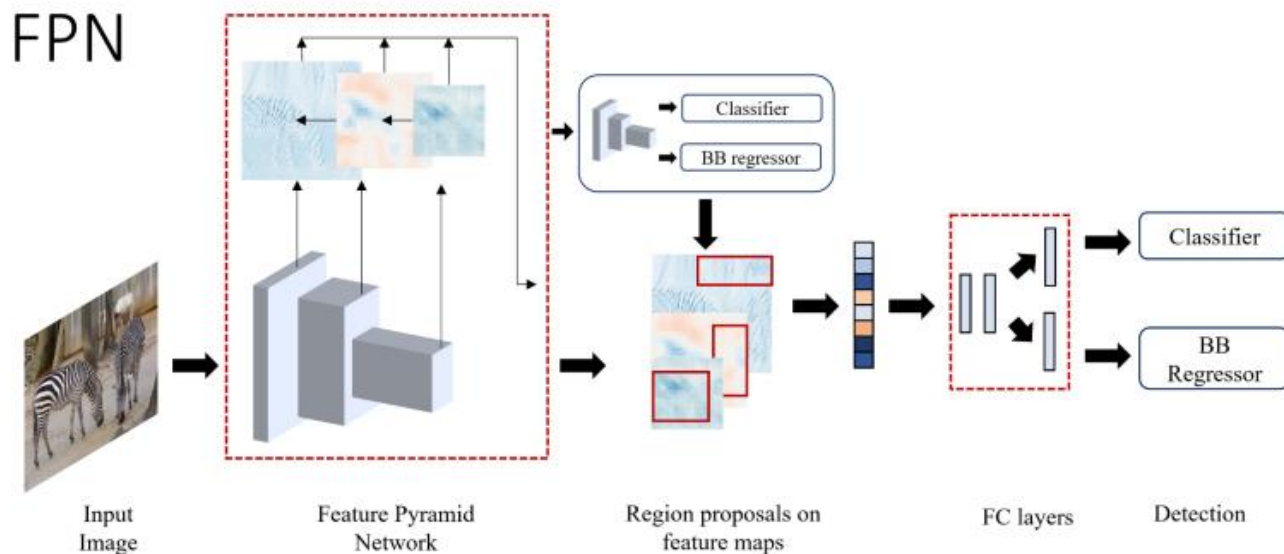




V. OBJECT DETECTORS

5) FPN

- utilizes an image pyramid to generate a feature pyramid at multiple levels
- enhancing the detection of small objects
- substantial increase in inference time
- top-down architecture with lateral connections to construct high-level semantic features across different scales and serves as the RPN for a ResNet-101 based Faster R-CNN
- standard component in future detection models
- improving accuracy across the board and inspiring the development of other enhanced networks like PANet, NAS-FPN, and the state-of-the-art detector, EfficientNet.

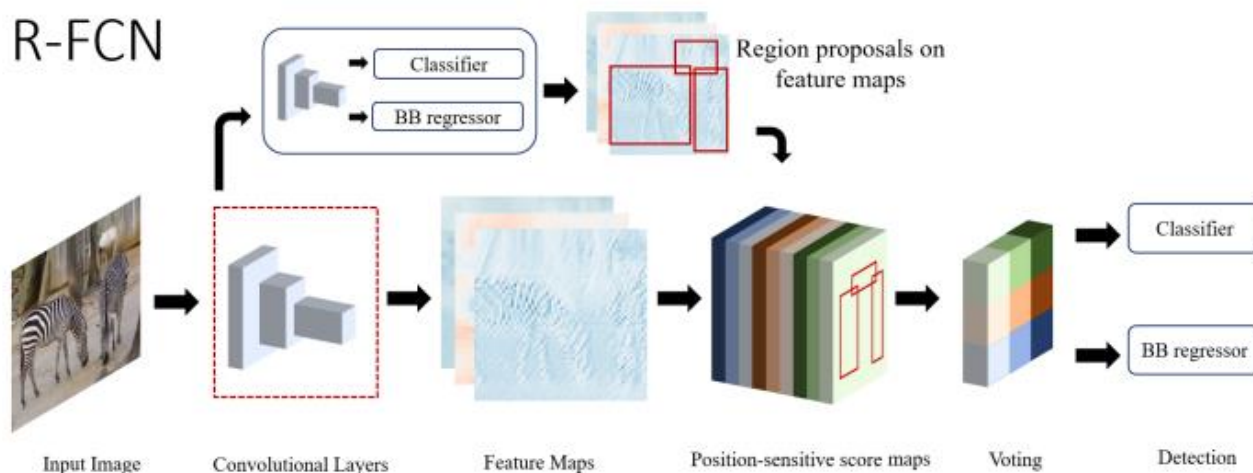




V. OBJECT DETECTORS

6) R-FCN : Region-based Fully Convolutional Network

- efficiently shares computations within the network compared to previous two-stage detectors
- By replacing fully connected layers with convolutional layers and employing position-sensitive score maps, R-FCN addresses translation invariance issues in localization tasks.
- didn't significantly improve accuracy
- outperformed its counterparts in speed by 2.5-20 times, making it a faster and equally capable detector

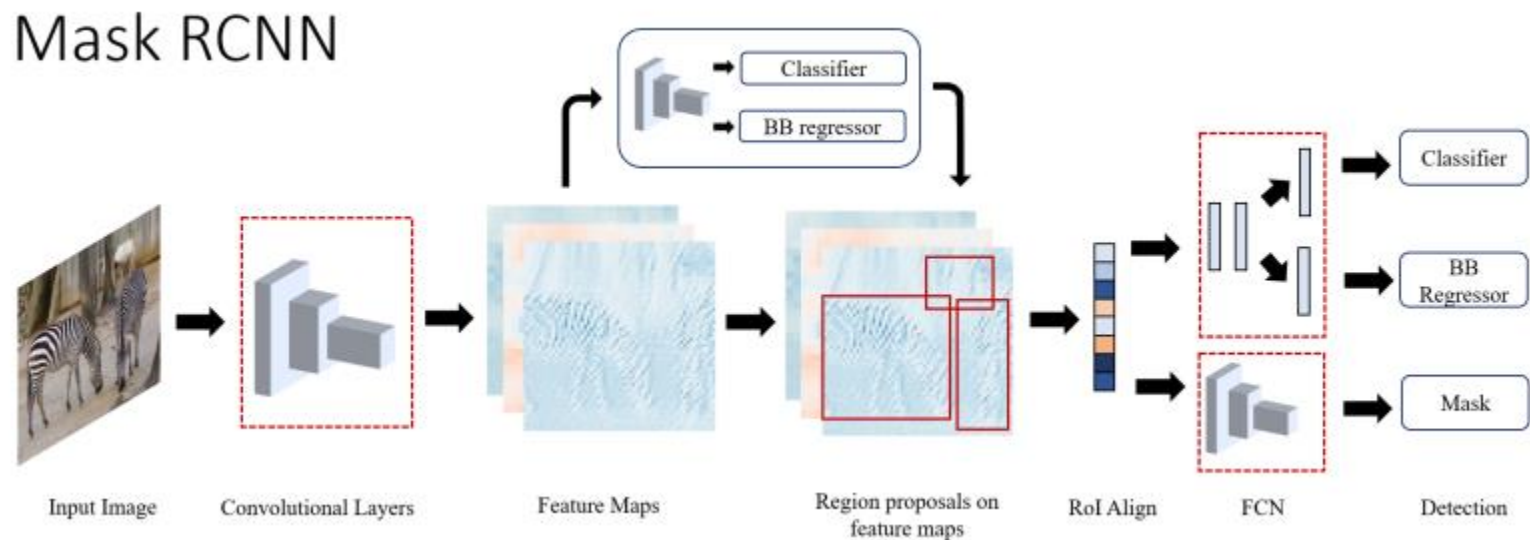




V. OBJECT DETECTORS

7) Mask R-CNN

- extends Faster R-CNN by adding a parallel branch for pixel-level instance segmentation
- reduces overall computational costs by classifying each pixel into segments using a fully connected network applied to Rols
- By employing the RoIAlign layer, it mitigates pixel-level errors due to spatial quantization and enhances accuracy and speed using ResNeXt-101 and FPN.
- offering additional functionalities
- still falls short in real-time performance compared to a single-model architecture
- RoIAlign : precisely aligns Region of Interest (RoI) to improve pixel-level accuracy





V. OBJECT DETECTORS

8) DetectoRS:

- two-stage detection
- proposing Recursive Feature Pyramid (RFP) at the macro level
- Switchable Atrous Convolution (SAC) at the micro level
- RFP enhances feature extraction by stacking multiple Feature Pyramid Networks (FPN) with additional feedback connections
- SAC regulates convolution dilation rates to detect objects across various scales efficiently
- state-of-the-art performance for two-stage detectors
- yet its processing speed limits its suitability for real-time applications
- handling only around 4 frames per second.



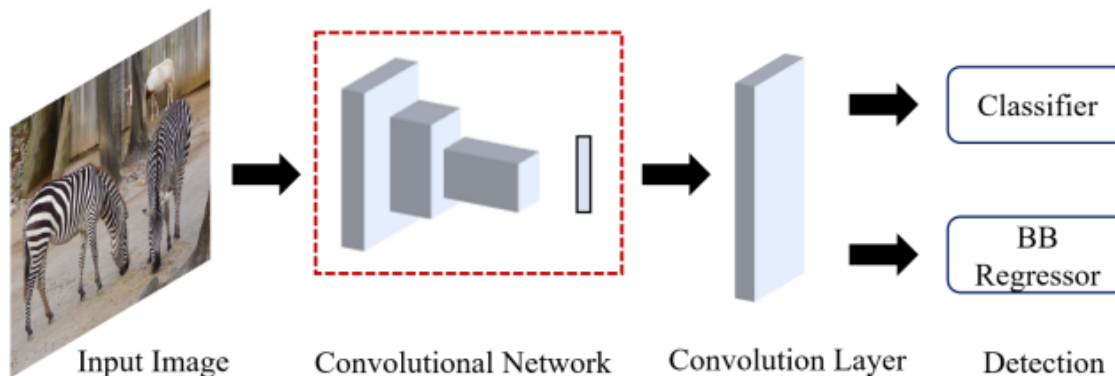
V. OBJECT DETECTORS

C. Single Stage Detectors

1) YOLO (You Only Look Once) :

- reframes two-stage as a regression task
- directly predicting object presence and bounding box attributes
- divides the input image into an $S \times S$ grid, with each grid cell responsible for detecting objects, predicting multiple bounding boxes and confidence scores.
- outperformed contemporaneous single-stage real-time models in both accuracy and speed
- challenges with small or clustered object localization and limitations on the number of objects per cell (addressed in subsequent versions)

YOLO

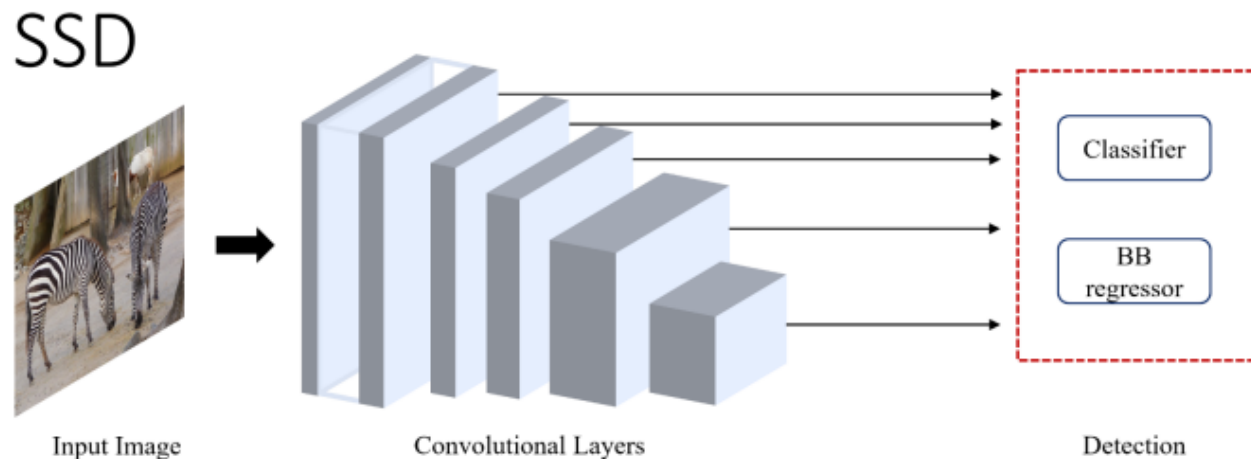




V. OBJECT DETECTORS

2) SSD: Single Shot MultiBox Detector

- comparable accuracy to two-stage detectors like Faster R-CNN in real-time while being significantly faster
- SSD incorporates auxiliary convolution layers of decreasing size at the end of the model to enhance performance
- detects smaller objects early in the network and uses deeper layers for offsetting default boxes and aspect ratios
- its speed and accuracy advantages over YOLO and Faster R-CNN, SSD initially struggled with detecting small objects
- limitation addressed in later versions by employing improved backbone architectures and minor adjustments.





V. OBJECT DETECTORS

3) YOLOv2 and YOLO9000:

- YOLOv2, an enhancement of YOLO, offered a balanced trade-off between speed and accuracy, while the YOLO9000 model achieved real-time prediction for 9000 object classes.
- replacing GoogLeNet with DarkNet-19 as the backbone architecture
- incorporated Batch Normalization for improved convergence, joint training of classification and detection systems for increased detection classes, and other techniques like removing fully connected layers and using learned anchor boxes for better recall
- Combining classification and detection datasets in a hierarchical structure using WordNet improved overall system performance by predicting higher conditional probabilities of hypernyms.

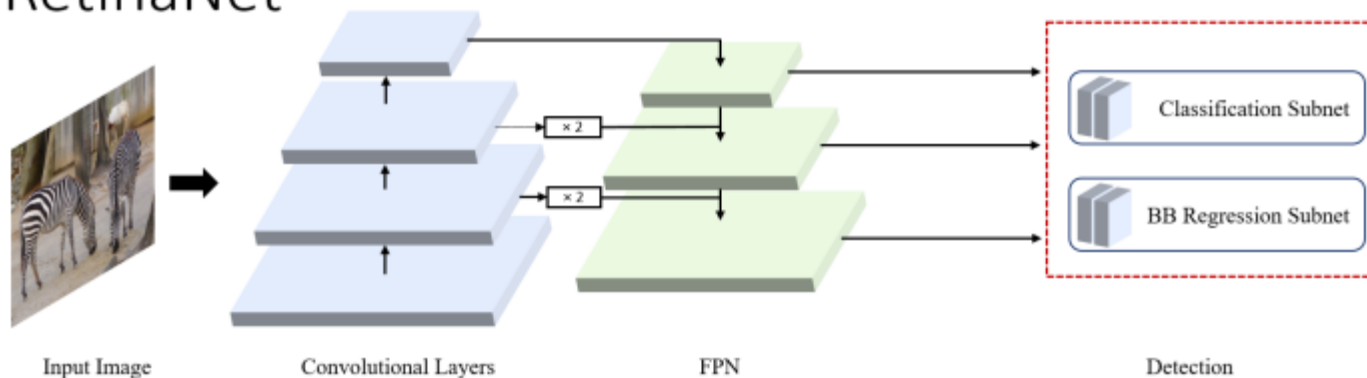


V. OBJECT DETECTORS

4) RetinaNet:

- attributed the lag in single-stage detectors to extreme foreground-background class imbalance
- proposed Focal loss, a reshaped cross-entropy loss, to address this issue by reducing the loss contribution from easy examples
- demonstrated the effectiveness of Focal loss with RetinaNet, a simple single-stage detector that densely samples the input image for object prediction in location, scale, and aspect ratio, utilizing ResNet augmented by Feature Pyramid Network (FPN) as the backbone.
- RetinaNet's use of class-agnostic bounding box regressor, efficient training, and implementation simplicity led to better accuracy and runtime performance compared to two-stage detectors, while also introducing advancements in object detector optimization

RetinaNet

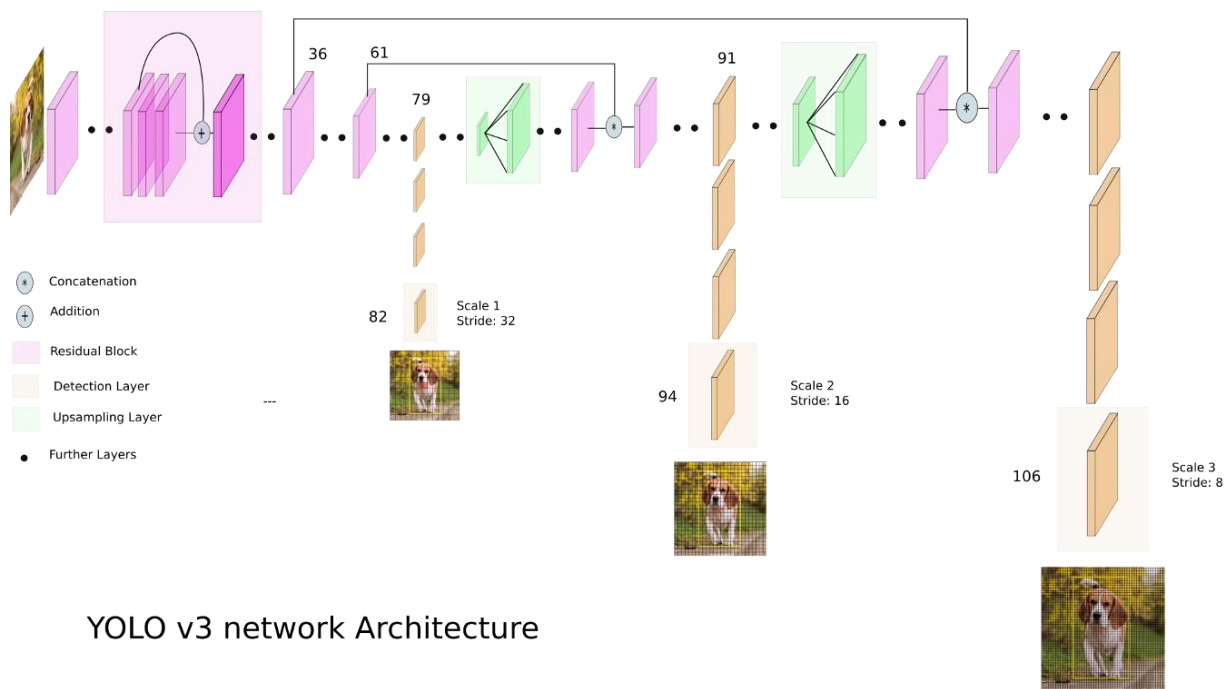




V. OBJECT DETECTORS

5) YOLOv3:

- introduced incremental improvements over previous versions by replacing the feature extractor network with a larger Darknet-53 network and incorporating techniques like data augmentation and batch normalization.
- Despite being faster than YOLOv2, YOLOv3 did not offer any groundbreaking changes and exhibited lower accuracy compared to a one-year-old state-of-the-art detector.

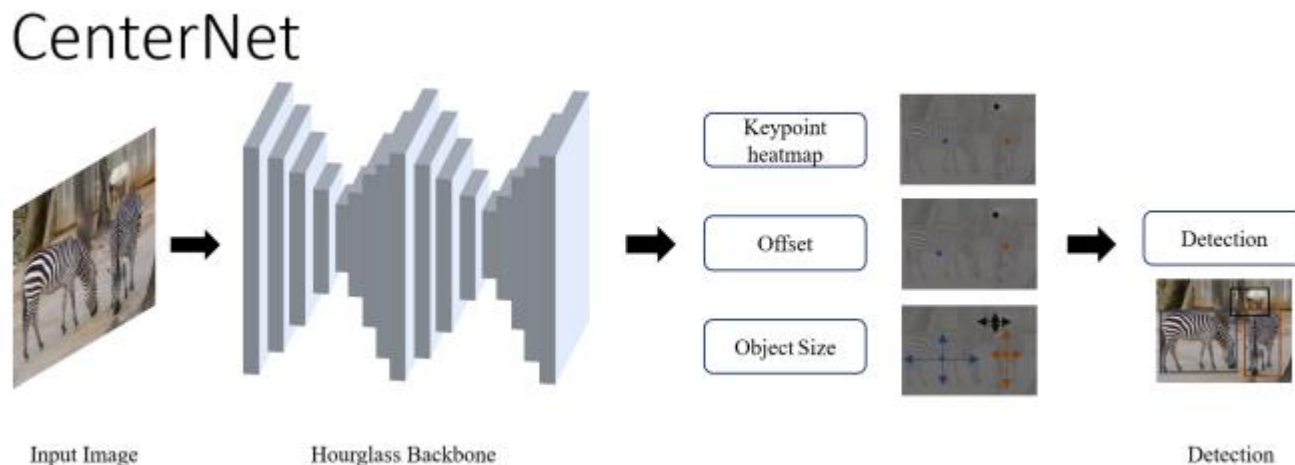




V. OBJECT DETECTORS

6) CenterNet:

- modeling objects as points instead of bounding boxes, with an FCN generating a heatmap for object center prediction.
- By using a pretrained stacked Hourglass-101 as the feature extractor and employing three heads for prediction, CenterNet achieves improved accuracy and faster inference without requiring non-maximum suppression, albeit needing specific backbone architectures for optimal performance.



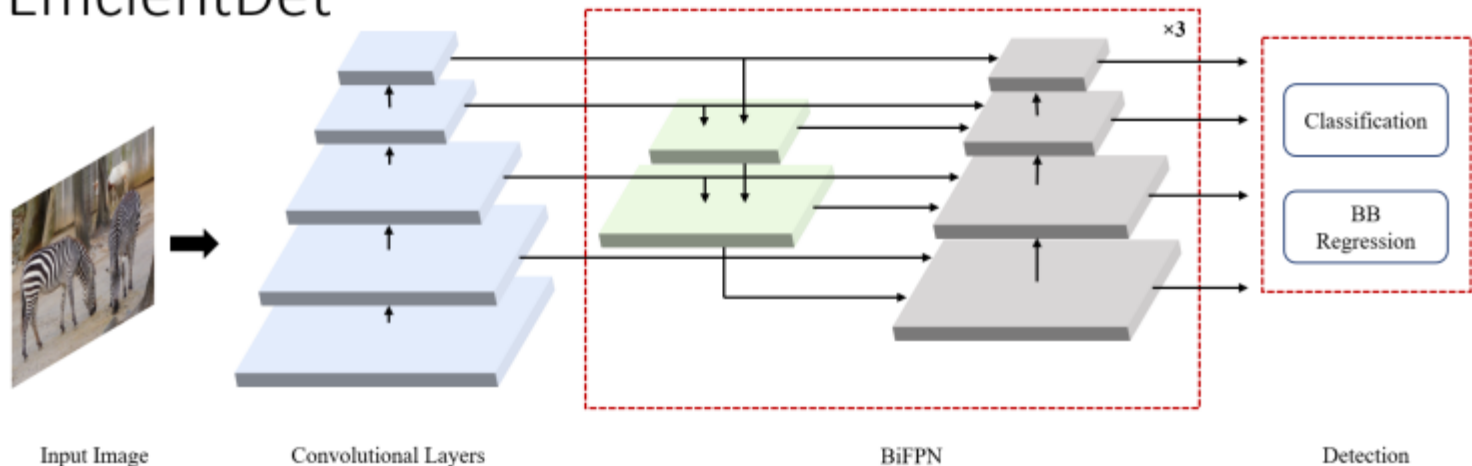


V. OBJECT DETECTORS

7) EfficientDet:

- EfficientDet enhances scalability, accuracy, and efficiency in object detection through efficient multi-scale features, BiFPN, and model scaling, utilizing EfficientNet as the backbone network and swish activation.
- superior efficiency and accuracy while being smaller and computationally cheaper, serving as the current state-of-the-art model for single-stage object detection.

EfficientDet

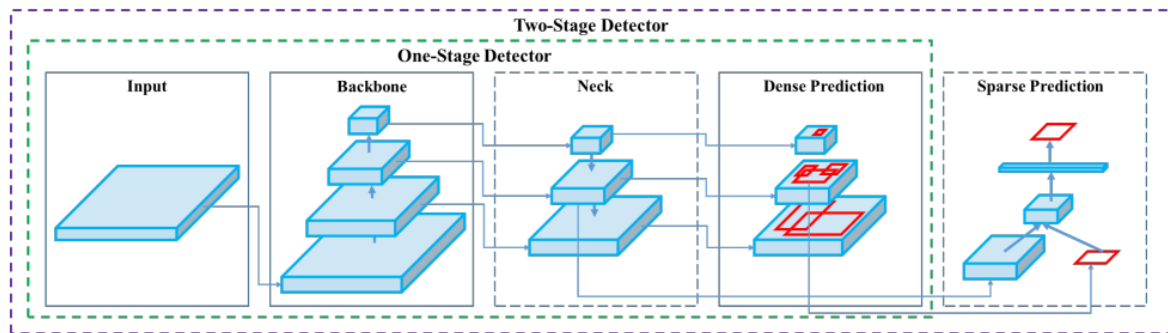




V. OBJECT DETECTORS

8) YOLOv4

- a range of innovations to create a fast and easily trainable object detector suitable for production systems, utilizing both "bag of freebies" and "bag of specials" techniques to improve training and inference
- Bag of Freebies : refers to techniques such as data augmentation, loss function modifications, and regularization, which impact the training process by increasing training costs to improve accuracy.
- Bag of Specials : encompasses architectural techniques, including post-processing, which solely increase inference costs to enhance accuracy.
- With a CSPNetDarknet-53 backbone, SPP and PAN block neck, and YOLOv3 detection head ->
- state-of-the-art performance for real-time single-stage detectors
- outperforming EfficientDet in speed while maintaining comparable accuracy on a single GPU



Input: { Image, Patches, Image Pyramid, ... }

Backbone: { VGG16 [68], ResNet-50 [26], ResNeXt-101 [86], Darknet53 [63], ... }

Neck: { FPN [44], PANet [49], Bi-FPN [77], ... }

Head:

Dense Prediction: { RPN [64], YOLO [61, 62, 63], SSD [50], RetinaNet [45], FCOS [78], ... }

Sparse Prediction: { Faster R-CNN [64], R-FCN [9], ... }

Figure 2: Object detector.

A. Bochkovski, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal speed and accuracy of object detection." [Online]. Available: <http://arxiv.org/abs/2004.10934>



V. OBJECT DETECTORS

9) Swin Transformer:

- Transformers, initially popularized in NLP, have shown remarkable success with models like BERT, GPT, and T5, owing to their ability to establish dependencies in sequences and attend to longer contexts.
- introduces a novel approach by providing a transformer-based backbone for computer vision tasks, splitting input images into non-overlapping patches and applying local multi-headed self-attention modules in successive blocks to maintain hierarchical representation.
- state-of-the-art results on the MS COCO dataset
- higher parameter usage compared to convolutional models remains a consideration

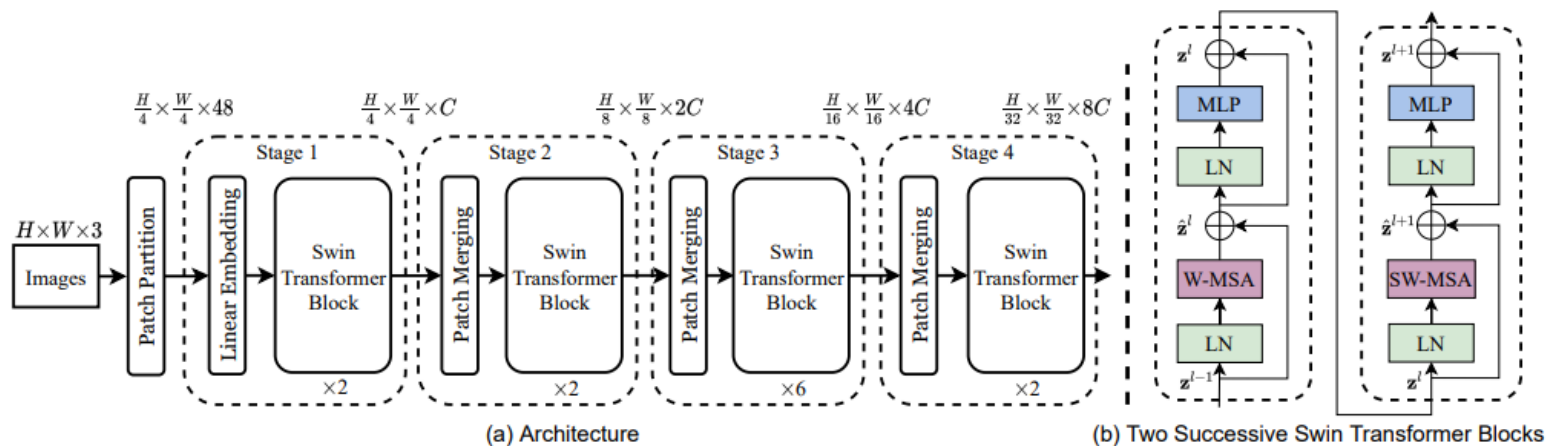


Figure 3. (a) The architecture of a Swin Transformer (Swin-T); (b) two successive Swin Transformer Blocks (notation presented with Eq. (3)). W-MSA and SW-MSA are multi-head self attention modules with regular and shifted windowing configurations, respectively.

Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," 2021.



VI. LIGHTWEIGHT NETWORKS

Recent research has focused on designing small and efficient networks for resource-constrained environments, with implications for object detection models as well.

While many object detection models offer high accuracy and real-time inference, they often require excessive resources, limiting their deployment on edge devices.

Various techniques such as efficient components, pruning, quantization, and distillation are explored to enhance the efficiency of deep learning models.

- distillation : Use of trained large network to train smaller models



VI. LIGHTWEIGHT NETWORKS

A. SqueezeNet

- a compact neural network architecture aimed at reducing parameters while preserving performance.
- employing smaller filters
- reducing the number of input channels to 3x3 filters
- strategically placing downsampling layers
- The architecture consists of fire modules, comprising squeeze and expand layers with ReLU activation, stacked between convolution layers
- can be enhanced with residual connections for improved accuracy



VI. LIGHTWEIGHT NETWORKS

B. MobileNets

- adopts an efficient network architecture using depthwise separable convolution, reducing computation cost and model size
- With 28 convolutional layers, batch normalization, and ReLU activation, it introduces width and resolution multipliers to enhance speed and reduce model size.
- achieves comparable accuracy to larger models across various applications like object detection and face attribution



VI. LIGHTWEIGHT NETWORKS

C. ShuffleNet

- optimized for mobile devices, offering enhanced computational efficiency.
- By integrating channel shuffle and group convolution techniques, ShuffleNet effectively mitigates scalability issues observed in efficient networks
- superior performance with significantly reduced model size compared to contemporaneous models
- innovative design marks a significant advancement in efficient network architecture.

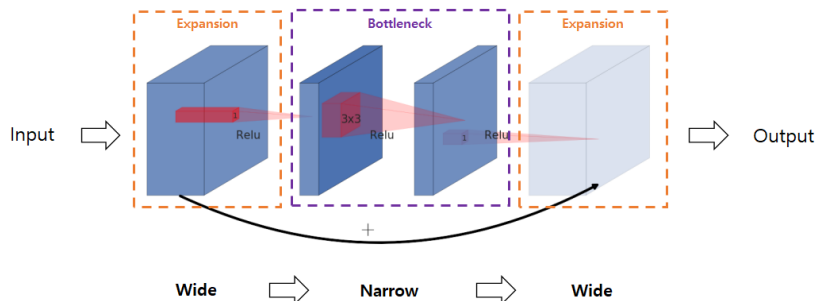


VI. LIGHTWEIGHT NETWORKS

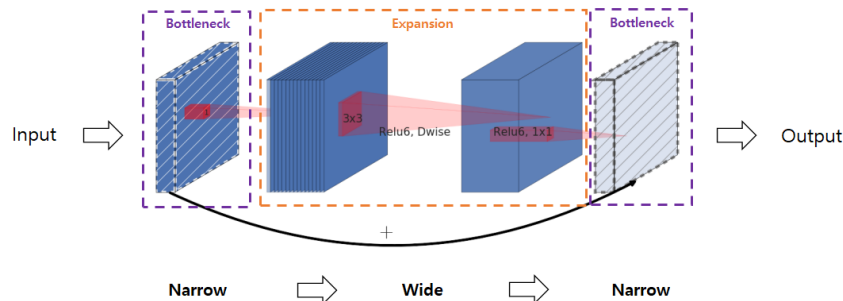
D. MobileNetv2

- an improvement over MobileNetv1
- featuring inverted residual with linear bottleneck modules to enhance efficiency and accuracy
- Unlike conventional residual blocks, MobileNetv2 expands the input dimension, applies depthwise convolution, and then compresses it back, optimizing computation.
- employs ReLU6 activation
- serves as the feature extractor for SSDLite, a variant of SSD, offering competitive accuracy with significantly fewer parameters.

(a) Residual block



(b) Inverted residual block





VI. LIGHTWEIGHT NETWORKS

E. PeleeNet

- a novel lightweight deep learning architecture inspired by DenseNet
- focus on efficient implementation using conventional convolution techniques
- PeleeNet incorporates two-way dense layers, stem block, dynamic channel adjustment, and conventional post activation to reduce computation while maintaining speed.
- PeleeNet serves as the backbone for Pelee, a real-time object detection system based on SSD, showcasing improved performance on mobile and edge devices through simple design choices

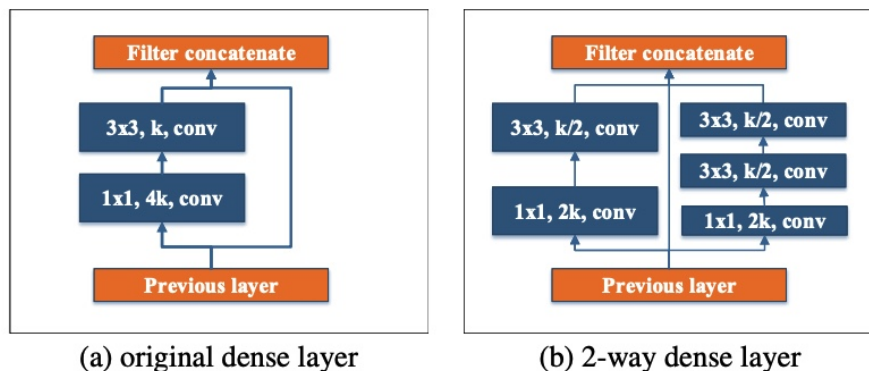


Figure 1: Structure of 2-way dense layer

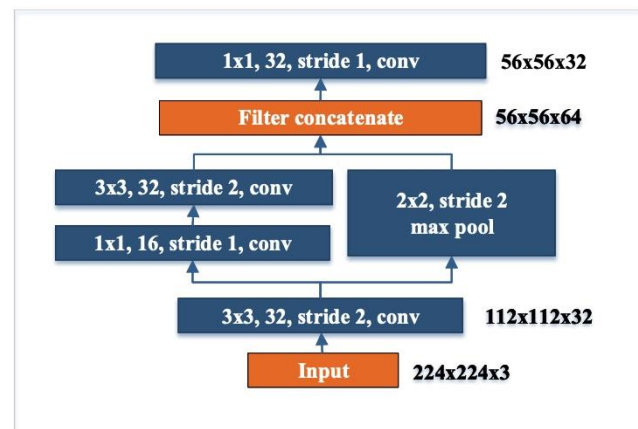


Figure 2: Structure of stem block



VI. LIGHTWEIGHT NETWORKS

F. ShuffleNetv2

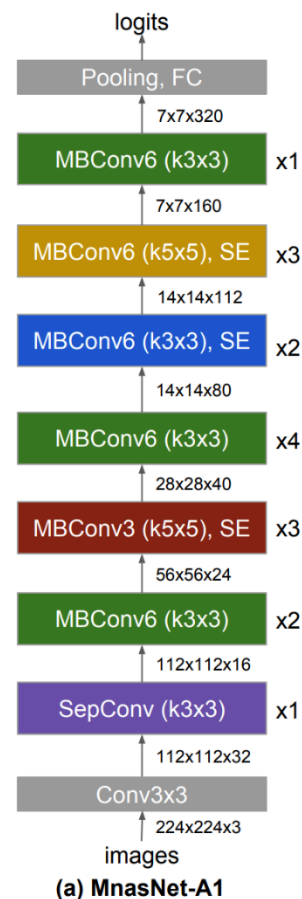
- efficient network design guidelines emphasizing direct metrics like speed over indirect ones like FLOPs
- novel building block splitting inputs, utilizing group convolution, and employing multi-path structures.
- This results in a highly efficient architecture with superior accuracy compared to state-of-the-art models at similar complexity levels.



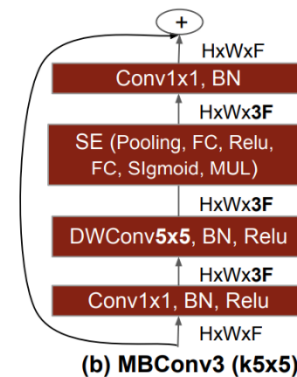
VI. LIGHTWEIGHT NETWORKS

G. MnasNet

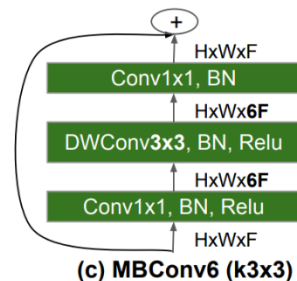
- utilizing an automated neural architecture search (NAS) approach to balance high accuracy and low latency.
- By partitioning the CNN into unique blocks and employing RNN-based reinforcement learning, they achieved diverse block designs, yielding faster inference and improved accuracy compared to MobileNetv2.
- computational demands for the search process remain significant



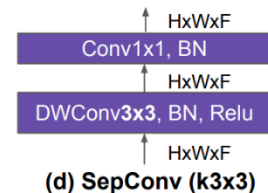
(a) MnasNet-A1



(b) MBConv3 (k5x5)



(c) MBConv6 (k3x3)



(d) SepConv (k3x3)



VI. LIGHTWEIGHT NETWORKS

H. MobileNetv3

- derived from MnasNet
- utilizes a platform-aware neural architecture search optimized with NetAdapt, achieving efficiency improvements
- By trimming redundant components and incorporating hard swish activation, Howard et al. crafted two variants tailored for diverse resource constraints, enhancing both speed and accuracy in feature detection for SSDLite with reduced latency.

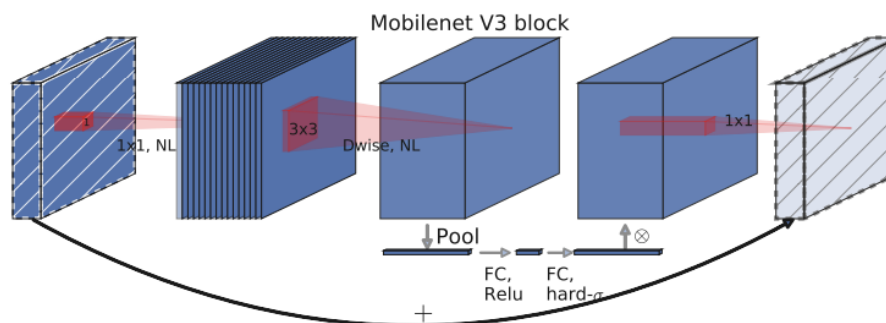


Figure 4. MobileNetV2 + Squeeze-and-Excite [20]. In contrast with [20] we apply the squeeze and excite in the residual layer. We use different nonlinearity depending on the layer, see section 5.2 for details.



VI. LIGHTWEIGHT NETWORKS

I. Once-For-All (OFA)

- decouples model training from neural architecture search (NAS), reducing computational costs.
- enables flexible sub-network creation by varying depth, width, kernel size, and dimension within a nested architecture, employing progressive shrinking for parameter reduction.
- state-of-the-art accuracy in ImageNet and winning the LPCVC
- demonstrates a novel approach to designing lightweight models for diverse hardware needs

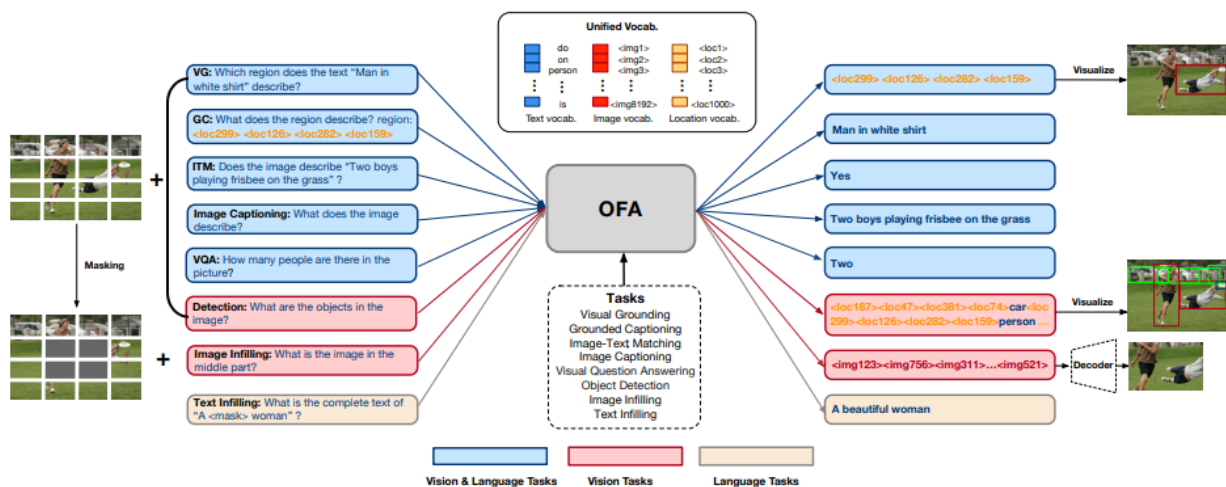


Figure 2: A demonstration of the pretraining tasks, including visual grounding, grounded captioning, image-text matching, image captioning, VQA, object detection, image infilling as well as text infilling.



VII. COMPARATIVE RESULTS

TABLE IV: Comparison of Lightweight models.

Model	Year	Top-1 Acc%	Latency (ms)	Parameters (Million)	FLOPs (Million)
SqueezeNet	2016	60.5	-	3.2	833
MobileNet	2017	70.6	113	4.2	569
ShuffleNet	2017	73.3	108	5.4	524
MobileNetv2	2018	74.7	143	6.9	300
PeleeNet	2018	72.6	-	2.8	508
ShuffleNetv2	2018	75.4	178	7.4	597
MnasNet	2018	76.7	103	5.2	403
MobileNetv3	2019	75.2	58	5.4	219
OFA	2020	80.0	58	7.7	595

evaluate performance of models based on the results from their papers

models are compared on average precision (AP) and processed frames per second (FPS) at inference time

Lightweight models are compared in table IV where we compare them on ImageNet Top-1 classification accuracy, latency, number of parameters and complexity in MFLOPs

Models with MFLOPs lesser than 600 are expected to perform adequately on mobile devices



VII. COMPARATIVE RESULTS

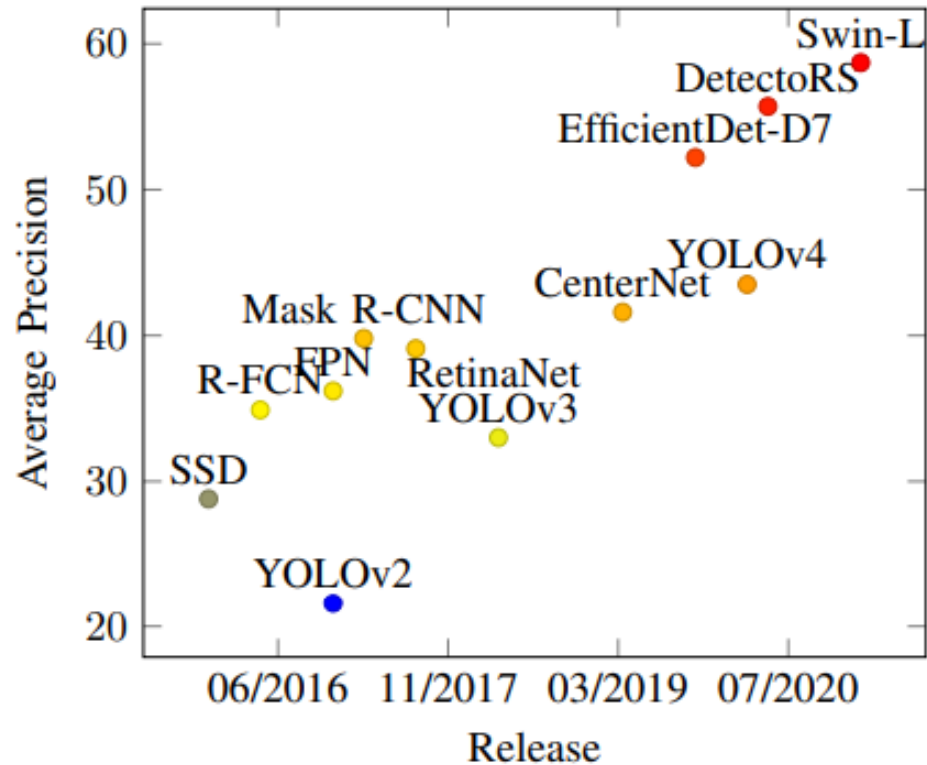


Fig. 10: Performance of Object Detectors on MS COCO dataset.



VIII. FUTURE TRENDS

open problem in the field of object detection

- AutoML: Using automatic neural architecture search (NAS) for object detector design is a growing area, yet resource-intensive and complex.
- Lightweight detectors: While lightweight networks excel in classification errors, they still lack in detection accuracy, increasing demand for mobile and embedded systems.
- Weakly supervised/few shot detection: Recent object detection models require extensive data annotation efforts, whereas Weakly supervised/few shot detection explores utilizing limited labeled data to reduce the cost and time associated with annotation.
- Domain transfer: Transferring models trained on labeled images to related tasks reduces reliance on large datasets, promoting model reuse and high accuracy.
- 3D object detection: Crucial for autonomous driving, achieving high accuracy in 3D object detection is paramount to address safety concerns.
- Object detection in video: Identifying objects in video by considering spatial and temporal relationships between frames remains a challenging problem.



IX. CONCLUSION

- still have room for improvement
- growing demand for lightweight models suitable for mobile and embedded systems
- single-stage detectors have become faster while maintaining high accuracy
- Swin Transformer emerging as one of the most accurate detectors to date
- increasing anticipation for more accurate and faster detectors