Machine Learning with Applications 6 (2021) 100134

Contents lists available at ScienceDirect

Machine Learning with Applications

journal homepage: www.elsevier.com/locate/mlwa

Deep learning in computer vision: A critical review of emerging techniques and application scenarios

Junyi Chai [a,b,*], Hao Zeng [a], Anming Li [c], Eric W.T. Ngai [c]

[a] Division of Business and Management, BNU-HKBU United International College, Zhuhai, China
[b] Centre for Evaluation Studies, Beijing Normal University, Zhuhai, China
[c] Department of Management and Marketing, The Hong Kong Polytechnic University, Hong Kong, China

ARTICLE INFO

Keywords:
Machine learning
Deep learning
Computer vision
Literature review

ABSTRACT

Deep learning has been overwhelmingly successful in computer vision (CV), natural language processing, and video/speech recognition. In this paper, our focus is on CV. We provide a critical review of recent achievements in terms of techniques and applications. We identify eight emerging techniques, investigate their origins and updates, and finally emphasize their applications in four key scenarios, including recognition, visual tracking, semantic segmentation, and image restoration. We recognize three development stages in the past decade and emphasize research trends for future works. The summarizations, knowledge accumulations, and creations could benefit researchers in the academia and participators in the CV industries.

Contents

Code metadata

Permanent link to reproducible Capsule: https://doi.org/10.24433/CO.0411648.v1.

1. Introduction

Deep learning (DL), a prevailing branch of artificial intelligence (AI), has been extended with diversified network structures. The features of big data could be captured by DL automatically and efficiently.

The code (and data) in this article has been certified as Reproducible by Code Ocean: (https://codeocean.com/). More information on the Reproducibility Badge Initiative is available at https://www.elsevier.com/physical-sciences-and-engineering/computer-science/journals.

* Corresponding author.
E-mail addresses: donjychai@uic.edu.cn (J. Chai), harveyhzeng@outlook.com (H. Zeng), amingli@polyu.edu.hk (A. Li), eric.ngai@polyu.edu.hk (E.W.T. Ngai).

https://doi.org/10.1016/j.mlwa.2021.100134
Received 17 March 2021; Received in revised form 6 August 2021; Accepted 6 August 2021
Available online 14 August 2021

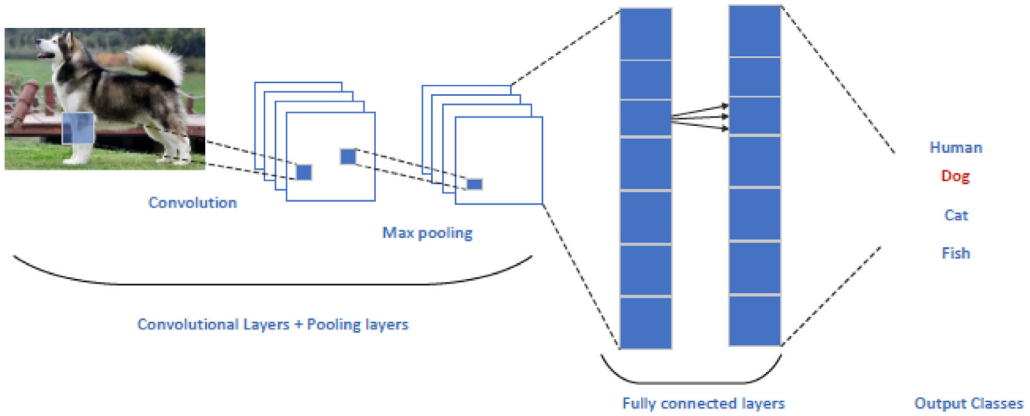

Fig. 1. CNN architecture for image classification.

The current applications of DL include computer vision (CV), natural language processing (NLP), video/speech recognition (V/SP), and finance and banking (F&B). Chai and Li (2019) provided a survey of DL on NLP and the advances on V/SP. The survey emphasized the milestones in DL development in these application domains. Their investigations showed that DL is deeply penetrated to the domains of NLP and V/SP. Huang et al. (2020) deliberated the cutting-edge developments of DL in F&B. Adamopoulou and Moussiades (2020) presented the history, technologies, and applications of natural dialog systems implemented through pattern matching approach. Muzammel et al. (2020) proposed a V/SP-based application called AudVowelConsNet, for clinical depression recognition and assessment from speech. With the development of DL, some scholars begin to explore the direction of industrial application. For example, Altan et al. (2021) developed a new hybrid wind speed forecasting (WSF) model for speed forecasting to efficient exploitation of wind power based on long short-term memory (LSTM) network and decomposition methods with gray wolf optimizer (GWO).

In the early stage of CV development, the DL approach faces difficult due to limitations of computer memory, CPU, and GPU. Most scholars thus are researching the application of ML in CV. Meanwhile, many methods for CV have been proposed, such as K-means, Naive Bayes classifier, Decision Tree, Boosting, Random Forest, Haar Classifier, Expectation–Maximization (EM), K-Nearest Neighbor (KNN), and Support Vector Machine (SVM). Based on Adaboost algorithm, Viola and Jones (2001) used the Haar-like wavelet feature and integral graph method for face detection. They are not the first to propose wavelet features, but they have designed more useful face detection features and cascaded the strong classifier trained by Adaboost. The proposed algorithm is called the Viola–Jones detector. Later, Lienhart and Maydt (2002) extended this detector by rotated Haar-like features and finally formed the Haar classifier that OpenCV now has.

The DL developments in past decades are rather rapid, which can be broadly separated into ten categories in terms of algorithm and architecture: Convolutional Neural Networks (CNNs), Long Short-Term Memory Networks (LSTMs), Recurrent Neural Networks (RNNs), Generative Adversarial Networks (GANs), Radial Basis Function Networks (RBFNs), Multilayer Perceptrons (MLPs), Self-Organizing Maps (SOMs), Deep Belief Networks (DBNs), Restricted Boltzmann Machines (RBMs), and Autoencoders. Guo et al. (2016) compared the literature and their respective performance on different CV tasks, including image classification, object detection, image retrieval, semantic segmentation, and human pose estimation. By comparing CNN, RBM, Autoencoder, and Sparse Coding, they finally concluded that CNN was the most suitable architecture for CV. Due to the limitation of precisions and model sizes at that time, nevertheless, there were several challenges in practical application. They included (a) no clear understanding of which architectures should perform better than others; (b) training with limited data; (c) hard to achieve real-time applications; (d) need more powerful models.

Moreover, Xu et al. (2020) summarized the potential for CV to assist on-site managerial tasks based on the articles published since 2014. More recently, Alzubaidi et al. (2021) introduced the structure, developments, hardware, and technological challenges of CNN and analyze the backbones of CNN from 2012 to 2018. They paid more attention on the development of CNN. Differently, our paper will focus on the development of DL in the CV field and makes a progressive summary in the way of the timeline.

The DL applied in information processing has more accumulation in the literature. Thus, we implemented the following conditions to limit our collection of articles. We only selected articles published on machine learning (ML), artificial intelligence, computer science, pattern recognition, business management because these articles are most possibly in accordance with the focus of this survey. Second, were searched the reviewed articles from academic databases, including Science Direct, Springer-Link Journal, IEEE Xplore, Emerald, JSTOR, World Scientific Net, and Google scholar. Third, we limited the period of publications to between 2014 and 2020, where few exceptions exist due to their significance.

Since the outstanding performance in the ImageNet competition, CNN have become the most notable DL approaches (Guo et al., 2016). One of the most important and basic field for CV applications is Image classification. Fig. 1 shows the CNN architecture for image classification. The CNN consists of convolutional layers, pooling layers, and fully connected layers. In the convolutional layers, a CNN uses various kernels to convolve the whole image and the intermediate feature maps, generating various feature maps. The pooling layers are used to reduce the dimensions of feature maps and network parameters. For the fully connected layers, it is generally at the end of each CNN architecture and works as a CNN classifier. After the fully connected layers, the output can be used for Image classification as shown in Fig. 1, or can transfer the output to the next Deep Neural Networks (DNN) as shown in Fig. 2. Therefore, Szegedy et al. (2016) argue that the research achievement gains in the classification performance tend to transfer to significant quality gains in a wide variety of application fields.

Specifically, we summarize recent developments in DL by looking into the eight *emerging* techniques which become the basic models in many CV application fields, including AlexNet, VGGNet, GoogLeNet & Inception, ResNet, DenseNet, MobileNets, EfficientNet, and RegNet. The broad applications of DL are boiled down to four main application scenarios including recognition, visual tracking, semantic segmentation, and image restoration. We analyze the recent literature and reconsider them as three development stages. Finally, we put forward the future research trends in application sides and the future works.

This paper aimed to identify the emerging techniques of DL and the recent achievements of application scenarios in the CV domain.

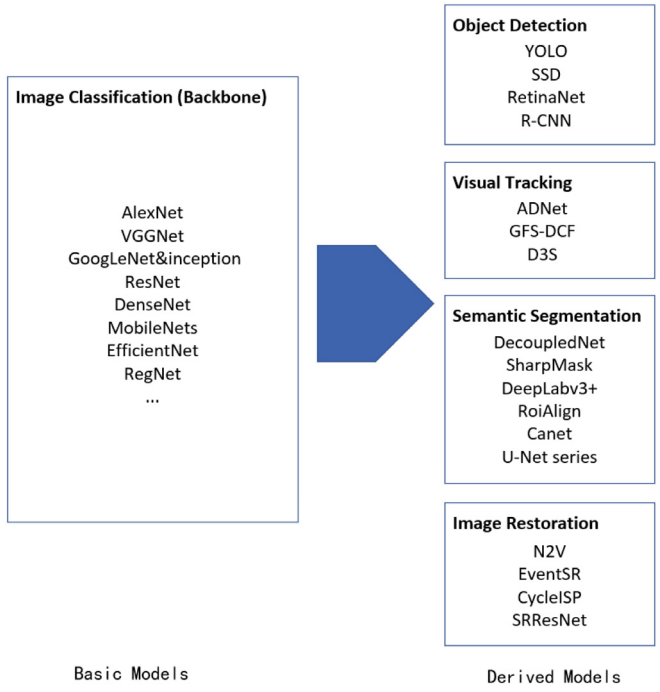

Fig. 2. CNN basic models and derived models.

Thus, we reviewed eight techniques and four applications in detail on the basis of quality publications between 2014 and 2020. We categorized the recent development for the past decade into three stages, namely, the early stage (2012–2016), the middle stage (2016–2019), and the recent stage (from 2019). We identified three research trends on the application side and two directions on the technical side. Our summarizations, knowledge accumulations, and creations could benefit researchers in the academia and participators in the CV industries.

The rest of the paper is organized as below. We will first review the emerging techniques in Section 2 and then review four application scenarios in Sections 3–6. We analyze the recent developments and outline future research trends in Section 7. Section 8 concludes this paper.

2. Recent developments on deep network architectures and evolvement

In the past decade, the mainstream technology in the field of CV is CNN. Numerous CNN-based network structures have emerged after the success of AlexNet on various tasks of image classification. The following is a description of recent developments in deep network architectures and evolvement:

(1) AlexNet: Proposed by Krizhevsky et al. (2012), Alexnet consists of five convolutional layers, followed by three connected layers. Each convolutional layer is followed by a rectified linear unit (ReLU) used to "activate" outputs of convolutional layers. AlexNet's original model is trained on two GPUs. One may consider the existence of a CNN-based network structure called Caffenet, which has a similar structure to Alexnet. The difference is that Caffenet applies pooling before local response normalization at the first two convolutional layers, whereas *Alexnet* does the opposite.

(2) VGGNet: Proposed by Simonyan and Zisserman (2015), VGGNet increases the depth of the network through adding more convolutional layers by using small convolution filters (3×3) while other parameters are fixed. By pushing the depth to 16 and 19 weight layers, a significant improvement on the prior-art configurations could be achieved, generally called VGG-16 and VGG-19. Although VGG-16 and VGG-19 endorse that increasing the depth of the network could affect the final performances of the networks to a certain extent, VGGNet is superior to other methods of the same period because it has a large parameter space. VGGNet's final model has more than 500 M, while AlexNet has only 200 M. Accordingly, it usually takes a longer time to train a VGG model than AlexNet.

(3) GoogLeNet & Inception: Lin et al. (2014) introduced Network-in-network (NIN), which consists of a stack of *mlpconv* layers. It replaces convolution filters with a general nonlinear function *approximator*. Another feature of NIN is that it uses global average pooling to replace fully connected layers. It averages each feature map and feeds the resulted vector directly to *softmax* layer. Experiments on several image datasets showed that NIN achieved comparable or better classification accuracy with much fewer parameters.

Szegedy et al. (2015) proposed a new CNN architecture called Inception v1 as Fig. 3 shows. For the time, increasing the size of architecture is safe to improve the performance. Nevertheless, they argued that it could result in two bottlenecks: (a) the larger number of parameters and (b) the increasing use of the computational resource. To solve these problems, they introduced the inception — the layers of CNN architecture. It manages to increase the depth and width of the network while keeping the computing budget constant. The inception layers are repeated multiple times and formed GoogLeNet, a 22-layer deep model. GoogLeNet utilizes two ideas in NIN: the 1×1 Convolution and global average pooling.

Afterward, Szegedy et al. (2016) introduced a set of tricks to improve the efficacy of the original design of Inception v1. It points out that convolutions with larger filters tend to be disproportionally expensive in terms of computation. It suggests replacing filters with the size of 5×5 (7×7) with two stacked 3×3 filters. This design calls Inception v2. The authors also mentioned a batch normalization (BN) auxiliary, which used normalization within each mini-batch data to normalize the output to a normal distribution of N (0,1), reducing changes in the distribution of internal neurons. They refer to this design as Inception v3. Inspired by ResNet, Szegedy et al. (2017) introduced Inception v4 as a simplified version of Inception v3. They combined Inception architecture with residual connections and create a new architecture called Inception-ResNet. Finally, Chollet (2017) proposed Xception to improve Inception v3's performance, replacing Inception modules with depthwise separable convolutions. Xception marginally outperforms Inception v3 on the ImageNet dataset and is far superior on the JFT dataset.

(4) ResNet: He et al. (2016) argued that learning a residual function concerning layer input was more efficient than learning layer parameters without referring to inputs. They proposed a residual network called ResNet with 152 layers, which was eight times deeper than VGG Nets. The residual network utilized multiple parameter layers to learn the representation of residuals between input and output, rather than using parameter layers to directly learn the mapping between input and output in general CNN networks (e.g., AlexNet, VGG). As the direct connections are increased, it achieves the purpose of promoting the vanishing gradient problem, strengthening feature propagation, encouraging feature reuse, and substantially reducing the number of parameters.

(5) DenseNet: Based on the observation that convolutional networks are more accurate and faster, Huang et al. (2017) introduced *DenseNet* that connects all layers directly with each other. Using the feature maps from all preceding layers as inputs, the DenseNet can create L(L+1)/2 connections rather than L connections of traditional convolution networks. As a result, it has four advantages: (a) alleviating the vanishing-gradient problem, (b) strengthening feature propagation, (c) encouraging feature reuse, (d) substantially reducing the number of parameters.

(6) MobileNets: Howard et al. (2017) presented a class of efficient models called MobileNets (v1), which used two simple global hyperparameters that efficiently trade-off between latency and accuracy. In the

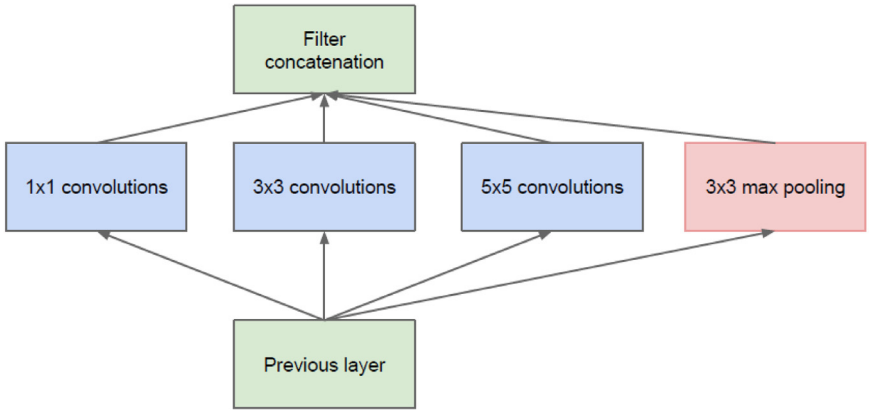

Fig. 3. Inception module, naive version (Szegedy et al., 2015).

ImageNet dataset, for example, the MobileNet (v1) parameter is only 4.2 million, while VGG16 uses 138 million, which results in a longitude difference of only 0.9%. On the downside, the structure of MobileNet V1 is similar to VGG. Compared with ResNet, Densenet and other structures are relatively low-cost performance. The depthwise Convolution greatly reduces the computational cost. The $N \times N$ depthwise $+1 \times 1$ pointwise structure can be close to the $N \times N$ Conv in performance. In actual use, the kernel of the depthwise part is easy to be abandoned by training. Nevertheless, many kernels trained by depthwise are empty.

Sandler et al. (2018) introduced a neural network called MobileNetV2 to improve the MobileNetV1 in two main aspects. First, they introduce a linear bottleneck that uses Linear activation instead of ReLU to prevent the nonlinear layer from losing some information. Inspired by ResNet, they presented Inverted Residuals to Improve gradient propagation between layers, with higher memory efficiency.

Howard et al. (2019) made MobileNetV3 come in two versions, MobileNetv3-Small and MobileNetv3-Large, which have lower and higher computation and storage requirements. MobileNetV3 applies network architecture search (NAS) and NetAdapt algorithm to improve the performance. MobileNetv3-Large improves accuracy by approximately 3.2% over MobileNetV2 in ImageNet classification tasks but reduces the time by 20%. Compare with MobileNetv3-Small, the accuracy of the ImageNet classification task is improved by 6.6% with comparable latency. The MobileNetv3-Large achieves the same accuracy on COCO detection and is 25% faster than MobileNetV2.

(7) EfficientNet: Tan and Le (2019) found that make the balance of network depth, width, and resolution can lead to better performance, and thus introduce *EfficientNet*. Overall, EfficientNet-B7 achieves the state-of-the-art 84.4% top-1 and 97.1% top-5 accuracy on ImageNet, while being 8.4x smaller and 6.1x faster on the inference than GPipe (Huang et al., 2019). EfficientNets also transfer well and achieve state-of-the-art accuracy on CIFAR-100 (91.7%), Flowers (98.8%), and three other transfer learning datasets, with an order of magnitude fewer parameters.

(8) RegNet: Radosavovic et al. (2020) presented a new network design paradigm called *RegNet* that combines the advantages of manual design and NAS. The RegNet design space can work perfectly across a wide range of flop regimes as it is a simple and fast network. Under similar training settings and flops, the RegNet model outperforms the popular EfficientNet model, which is up to 5-times faster on the GPU. Although the precision of RegNet is not a great improvement compare with EfficientNet, it proposes new ideas in the direction of design network design spaces. At present, RNN and LSTM are more frequently used in NLP and audio recognitions. An increasing trend is the intersection of the fields of CV and NLP. Ye et al. (2020) proposed a dual convolutional LSTM (ConvLSTM) network that consists of an encoder network and a decoder network for capturing spatial and sequential information in the input image.

Apart from CNN and RNN, Restricted Boltzmann Machine (RBM) is a two-layer shallow neural network that learns the joint probability of visible inputs and hidden units. RBM learns the probability of a hidden unit, a given input x, and the weight is p(a| x; w). A deep belief network can be regarded as a stack of RBMs. Autoencoder will try to learn a set of parameters for the reconstruction of x after being given a set of inputs x. It normally contains two modules: encoder and decoder. For Sparse Coding, Elad and Aharon (2006) provided the classic K-SVD algorithm. Xie et al. (2012) developed a training scheme for denoising auto-encoder, which can catch up with the performance of the classic one (see Table 1).

3. Recognition

3.1. Image classification

Image classification aims to assign a pre-defined label to an input. All network architecture mentioned hereinbefore can be used for image classification. Zeiler and Fergus (2014) proposed to use a multi-layered deconvolutional network (*deconvnet*) to project feature activations back to the input pixel space. Instead of mapping pixels to features, *deconvnet* conducted the opposite. To achieve visualization of pixel space, they run the previous results through the next three processes:

(a) Unpooling: by recording the maxima locations within each pooling region to place the reconstructions from the layer above into appropriate locations.

(b) Rectification: reconstruct signal through a ReLU non-linearity.

(c) Filtering: uses learned filters to convolve the feature maps from the previous layer.

Projections of different layers of trained *deconvnet* show the hierarchical nature of features in the *Alexnet*. It concluded that a smaller receptive window size and smaller stride step of the first layer could improve the performance of *AlexNet* for image classification.

Other algorithms are worth mention here. To solve non-transparency problems in DNN, Lapuschkin et al. (2016) pointed out that the LRP can better explain the classifier and help people gain more scientific insights. For example, when distinguishing illustrations from photographs, handcrafted features and outline detection features perform poorly due to the absence of dark outline colors. In contrast, fine-tuning DCNN has extremely high accuracy (96.8%), which outperforms the other models, including that the custom CNN models that were trained from scratch (Gando et al., 2016).

For high-resolution synthetic aperture radar image classification, a discriminant deep belief network (DisDBN) is introduced to learn discriminant and robust features. After training a set of weak classifiers, performing discriminative projection, and learning high-level discriminative features, the author proves the effectiveness of DisDBN through three experiments. However, due to the neighbor selection strategy of weak classifiers, a large deviation in pseudo-labeling may exist (Zhao et al., 2017).

Table 1
A detailed summary about deep network architectures and evolvement.

Architecture	Year	Model complexity	SPECIAL optimization tricks	Test dataset	Limitation
AlexNet	2012	60 million parameters and 650,000 neurons, consists of five convolutional layers, some of which are followed by max-pooling layers, and three fully-connected layers with a final 1000-way softmax.	Two GPUs, Rectified Linear Units (ReLUs), Local Response Normalization (LRN), Overlapping Pooling, Data augmentation, Dropout	LSVRC-2010	Space for accuracy improvement
VGGNet	2014	Convolutional layers use very small (3×3) convolution filters in all layers to increase the depth of the network. Several networks are proposed, notably VGG16 (138 million parameters) and VGG19 (144 million parameters). 13/19 convolutional layers with three fully connected layers.	Build depth network by reusing simple convolution blocks (small (3×3) convolution filters), Dense evaluation, Multi-crop evaluation	LSVRC-2010	Evaluating the network requires a lot of computation
GoogLeNet/Inception v1	2014	9 Inception modules, 22 layers deep (21 convolutional layers and one fully connected layer) when counting only layers with parameters (or 27 layers if we also count pooling) and 12 times fewer parameters than AlexNet.	Inception module, Global Average Pooling, Auxiliary classifiers, Dropout, Multi-crop evaluation	ILSVRC-2012	The complexity of the Inception architecture makes it more difficult to make changes to the network. Makes it much harder to adapt it to new use-cases while maintaining its efficiency.
Inceptionv2	2016	Total ten inception modules with factorizing the traditional 7×7 Convolution into three 3×3 convolution s.network is 42 layers deep. Our computation cost is only about 2.5 higher than that of GoogLeNet, and it is still much more efficient than VGGNet.	Batch Normalization (BN), Three types inception module, Grid Size Reduction	ILSVRC-2012	
Inceptionv3	2016	Based on inceptionV2 architecture, re-add some special optimization tricks with less than 25 million parameters.	BN-auxiliary, Multi-crop evaluation, Label Smoothing Regularization (LSR), RMSProp	ILSVRC-2012	Relatively complex and partitioning the model for distributed training using disbelief
Inceptionv4	2017	Total 14 inception modules with one stem module.	Stem modules, New inception modules	ILSVRC-2012	More complex and takes longer to run than V3
ResNet	2016	152 layers, which is eight times deeper than VGG Nets, and with 1.7 million parameters.	Residual Block, Shortcut connection, Zero-padding, Projection shortcut	ILSVRC-2012, PASCAL VOC 2007/2012	Suffer from overfitting
DenseNet	2017	Three dense blocks and Transition Layers. The number of parameters increases from 1.0M to 27.2M while not suffering from overfitting or residual networks' optimization difficulties.	Dense blocks, Batch Normalization(BN)	CIFAR-10 (C10), CIFAR100, Street View House Numbers (SVHN), ILSVRC-2012	
MobileNets (v1)	2017	Two simple global hyperparameters efficiently trade-off between latency and accuracy. The parameter is only 4.2million.	Depthwise convolutions, Pointwise convolution, Batch Normalization (BN)	YFCC100M, Stanford Dogs, Im2GPS, COCO, ImageNet	Space for accuracy improvement
MobileNetV2	2018	Factorizing traditional convolutions to 1×1 convolutions. The parameter is just 3.4 million.	Inverted residual structure, Linear Bottleneck	COCO, PASCAL VOC 2012	Space for accuracy improvement
MobileNetsV3	2019	V3-Large has 5.4 million parameters, and V3-Small has 2.5 million parameters.	h-wish, Squeeze-and-excitation (SE), NetAdapt, Neural Architecture Search (NAS)	COCO, Cityscapes	Improves the accuracy but reduces the speed
EfficientNet	2019	Compound scaling the network. Have eight kinds of network, EfficientNet-B0 only has 5.3 million parameters, and even EfficientNet-B7 only has 66 million parameters.	Compound Model Scaling	CIFAR-100, CIFAR-10, Flowers, Birdsnap, Stanford Cars, FGVC Aircraft, Oxford-IIIT Pets, Food-101	Search cost for grid search is high.
RegNet	2020	Combines the advantages of manual design and NAS.REGENCY-400MF only uses 4.3 million parameters to reach a similar result of EfficientNet-B0.	Design space design	LSVRC-2010	

3.2. Object detection

The image classification describes the image, while object detection aims to detect the location of a set of target objects. The detection task consists of two sub-tasks, one is the category information and probability of the target, and it is also a classification task. The other is the specific location of the target by utilizing bounding boxes with labels, which is a positioning task.

The current mainstream methods are mainly divided into a one-stage approach (e.g., SSD, YOLO) and a two-stage approach (e.g., R-CNN series). The two-stage approach firstly generates a sparse set of the bounding box from the image. It then makes corrections based on the bounding box region to improve the final detection results. In contrast, the single-stage approach directly calculates the image and generates detection results. The single-stage detection speed is faster, but the detection accuracy is lower. In contrast, the two-stage approach is completely the opposite. Object detector components and classifications are shown in Table 2.

3.2.1. One-stage detectors

(1) YOLO

Redmon et al. (2016) proposed YOLO that frames object detection problem as a regression problem instead of classification. One notable feature of this method is fast detection speed. As claimed by the authors, YOLO can achieve 45 frames per second, and the fast version has higher efficiency. That is, 155 frames per second doubles mAP (mean of Average Precision) compare with other real-time systems. Note that YOLO still lags in critical detection systems in terms of accuracy.

Redmon and Farhadi (2017) introduced YOLO9000 (also known as YOLOv2), which made various improvements on YOLO and can detect over 9000 object categories. Compare with YOLO, YOLOv2 made the following changes, including batch normalization, use high resolution training images, dimension cluster, and convolutional with anchor box, which means predicting offsets instead of coordinates of bounding boxes. At a speed of 40 FPS (Frame Per Second), YOLOv2 achieves 78.6 mAP on the VOC 2007 dataset, which outperforms the critical detection algorithm, faster R-CNN with ResNet and SSD. They also proposed a joint training method that can predict locations of object classes without labeled detection data. YOLOv3 is later proposed by Redmon and Farhadi (2018). The backbone of YOLOv3 has evolved from Darknet-19 in YOLOv2 to Darknet-53, which deepens the number of network layers and introducing the cross-layer add operation in ResNet. Although the Darknet-53 processes 78 images per second, which is considerably slower than Darknet-19 (171 FPS), it is still considerably faster than ResNet-152(37 FPS) and ResNet-101(53 FPS). 320 x 320 YOLOv3 runs in 22 ms at 28.2 mAP, as accuracy as SSD, yet three times faster. When we look at the old .5 IOU (Intersection over Union) mAP detection metric YOLOv3 is quite good. It achieves 57.9 AP_{50} in 51 ms on a Titan X, compared to 57.5 AP_{50} in 198 ms by RetinaNet, which has similar performance but 3.8 times faster. Although YOLOv3's accuracy is not significantly better than other networks, it has a higher speed than other competitors.

Redmon and Farhadi (2018) believed CV is already being put to questionable usages and have stopped the research on YOLO after the completion of YOLOv3. The subsequent versions of YOLO after YOLOv3 are all improvements based on V3. So far, the YOLO series has spawned several offshoots. Bochkovskiy et al. (2020) improved YOLOv4 significantly, such as weighted-residual-connection (WRC), cross-stage-partial-connections (SCP), cross mini-batch Normalization (CmBN), self-adversarial- training (SAT), and mish-activation. They also apply tricks including Mosaic data augmentation, DropBlock regularization, and CIoU (Complete-IOU) loss. The result of YOLOv4 is 43.5% AP (Average Precision) (65.7% AP_{50}) for the MS COCO dataset at a real-time speed of ~65 FPS on Tesla V100.

PP-YOLO (Long et al., 2020) have got improvements based on YOLOv3 after YOLOv4, which uses Resnet50-V instead of Darknet53 as the backbone. And, BatchSize changed from 64 to 196. In addition, by adding IOU Loss, Grid Sensitive, and IOU Aware, limited the use of DropBlock, PP-YOLO increases the mAP on COCO from 43.5% to 45.2%, with FPS improved from 62 to 72.9 compared with YOLOv4.

(2) SSD

Liu et al. (2016) presented a method that eliminates the process of generating bounding boxes. Their method first processes six feature maps. Each of the anchor boxes on each feature map generates a different length of the anchor boxes on the original input. Therefore, it can function feature maps from different resolutions to handle the various size of objects. The detection speed is up to 59 FPS when the input size is 300 × 300. Changing the input size to 512 × 512 achieves 76.9% mAP on VOC 2007 dataset, which outperforms the critical detection algorithm, a faster R-CNN.

Based on the SSD, Fu et al. (2017) attempted to change the base network from VGG into Residual-101 (He et al., 2016), while the accuracy rate drops from 77.5% to 76.4%. Inspired by MS-CNN (Cai et al., 2016), they added a prediction module to improve the sub-network of each task for improving accuracy. Although the final accuracy is not far from the accuracy of SSD513, which network is also Residual-101, Deconvolutional Single Shot Detector (DSSD) can better monitor small objects of an image.

(3) RetinaNet

Lin, Goyal et al. (2017) thought that the low accuracy of the one-stage approach was caused by the class imbalance and propose a new structure, RetinaNet, using Focal Loss. RetinaNet used ResNet and Feature Pyramid Network (FPN) as the backbone. It used single-level target recognition with focal loss, which can apply a modulating term to the cross-entropy loss. This is for focusing learning on hard examples and down-weighting the numerous easy negatives. This structure reaches 39.1 mAP higher than 36.2 mAP that Faster R-CNN on FPN (Lin, Dollar et al., 2017) got based on the challenging COCO datasets.

3.2.2. Two-stage detectors: R-CNN series

Inspired by the great success of CNN for image classification, Girshick et al. (2014) proposed a three-module method to utilize CNN on object detection. It firstly generated a set of object-independent object proposals in the form of regions in the input image. It then extracts fixed-length deep features using CNN from processed regions of the image. Finally, it feed these features to a set of linear SVMs (Support Vector Machine) that identify the type of objects. Authors named this method R-CNN.

He et al. (2015) re-examined the requirement of image inputs and propose a new pooling strategy. A fully connected layer only takes fixed-size input that results in fixed-size image input. They argued that it might lead to poor performance on tasks like object detection. They developed a network structure called SPP-net, which took an image of arbitrary size as inputs and reached 63.1 mAP on VOC 2007 test dataset. They claimed its advantage in handling object deformations. This paper profoundly impacts the later work regarding object detection because it enables fast feature extraction from different regions of an input image.

Inspired by He et al.'s (2015) work, Girshick (2015) designed a network that could classify objects and predicted the location of bounding boxes simultaneously, namely, fast R-CNN. Firstly, it generated a set of object proposals for an image using the algorithm called selective search. And then, it extracted deep features from each region with the aid of the *RoI* pooling layer. The extracted features and coordinates of their corresponding bounding box were fed into a *softmax* classifier and a bounding box *regressor,* respectively. Compared to R-CNN, this method has merged the last two modules and thus reduce training (18.3 x faster) and testing time (169 x faster) in the S group.

Ren et al. (2017) extended Girshick's (2015) work to achieve the goal of real-time object detection. Their method aimed at solving the bottleneck of generating object proposals. A Region Proposal Network (RPN) was introduced to predict object bounds and objectness scores.

Table 2
Object detector components and classification.

Input			Image, Patches, ImagePyramid
Backbone	GPU platform CPU Platform		VGG, ResNet, CSPResNeXt, CSPDarknet53, DenseNet, EfficientNet-B0/B7,GhostNet,RegNet SqueezeNet, MobileNet(V1–V3+), ShuffleNet(V1–V2),
Neck	Additional blocks Path-aggregation Blocks		SPP, ASPP, RFB, SAM FPN, PAN, NAS-FPN, NAS-FPN, Fully-connected FPN, BiFPN, ASFF, SFAM
Heads	One-stage	Anchor Anchor-free	RPN, SSD, YOLO(V2–V4), RetinaNet YOLOV1, CornerNet, CenterNet, MatrixNet, FCOS, ATSS, PAA,
	Two-stage	Anchor Anchor-free	Faster R-CNN, R-FCN, Mask R-CNN, Libra R-CNN RepPoints

This method merges all steps into a single neural network. Thus, they call this method a faster R-CNN. Therein, the computation is not shared on the whole page, resulting in extra computation and time. To overcome this shortage, Dai et al. (2016) introduced R-FCN, a fully convolutional region-based detector with a position-sensitive score map. The *RoI* pooling place between Faster R-CNN layers can affect translation invariance and make R-FCN get similar accuracy with 19x less time.

Lin, Dollar et al. (2017) proposed FPN. This architecture leverages the pyramidal shape of a ConvNet's feature hierarchy and builds high-level semantic feature maps at all scales. Using FPN in the Faster R-CNN model, their method improves the average precision significantly, achieving better performance than many heavily-engineered single-model entries of competition winners such as G-RMI and Faster R-CNN+++. He et al. (2017) extended Ren et al.'s (2017) work by inserting a parallel branch for predicting an object's mask to do the semantic segmentation. The work also introduces *RoI* Align to replace traditional *RoI* Pooling, which is not the pixel-to-pixel alignment, enhancing the accuracy from 10% to 50%. This method shows superior results than other models in all tracks of the COCO suite of challenges. This method is named Mask R-CNN, and reaches 62.3% AP_{50} on the MS COCO dataset. Apart from R-CNN and its variants, other significant methods also exist, improving the speed of detection or accuracy of detection.

R-CNN series, YOLO series, SSD, and RetinaNet mentioned above are all based on Anchor's target detection algorithm. Law and Deng (2018) presented a new Anchor Free model called CornerNet. CornerNet achieves the goal using heatmaps, embeddings, offsets, and 40.5% AP, 56.5% AP_{50}on MS COCO dataset. This can solve two main problems: (1) class Imbalance that tried Focal Loss to solve it (Lin, Goyal et al., 2017), and (2) the introduction of more hyperparameters, such as the number, the size, and the aspect ratio of anchors. Inspired by CornerNet, some researchers present plenty of Anchor Free models. Such as CenterNet (Zhou et al., 2019), FCOS (Tian et al., 2019), and RepPoints (Yang et al., 2019).

4. Visual tracking

Visual tracking is one of the most challenging topics in the CV field. In the real world, visual tracking is affected by external factors, including pose variations, illumination variations, full or partial occlusions, and noise in the video. Researchers pay further attention to multi-cue methods. Walia and Kapoor (2016) categorized multi-cue tracking methods into single modal and multi-modal. Differently, Kumar et al. (2020) categorized multi-cue object tracking into traditional architecture and DL-based trackers.

Wang and Yeung (2013) considered the object tracking problem as a problem of learning feature representation. They proposed to use stacked denoising autoencoders that learned a generic feature representation of images offline on auxiliary image data. For online tracking, they attached a classification layer to the encoder part. Both the classification layer and the encoder are then fine-tuned to adapt to changes in the appearance of an object. This method is named a deep learning tracker (DLT).

Wang, Liu et al. (2015) argued that generic features cannot capture temporal invariance, and DLT cannot transfer from offline learning to Online Tracking. To resolve both issues, they propose a two-layered CNN that learns features from offline auxiliary video sequences. The learned feature is then adapted online for a given target video sequence through an adaptation module. Authors claim that learned features are robust to both motion transformation and appearance changes. Concerning the discriminative power of generic features and the time-consuming training process, Zhang et al. (2016) proposed a CNN-based online training method that utilized a lightweight convolutional network structure. It consists of two layers: a simple layer and a complex layer. The simple layer contains fixed filters generated from the target region and its surrounding regions, while the complex layer is used to handle the location ambiguity problem. And, this convolutional network-based tracker (CNT) algorithm achieves the AUC of 0.545, which outperforms the DLT method by 10.9%. Wang, Li et al. (2015) conducted an in-depth study on features of different CNN layers to inspire a more effective feature extractor of a CNN-based tracker. The top layer of CNN has more discriminative power that distinguishes the target from other classes and is more tolerant to object deformation. On the other hand, a lower layer handles the distractor better than a top layer. Thus, they design a mechanism that can switch between features from the lower layer and features from the top layer depending on the presence of distractors. Using these improvements improves the AUC metric of the overlap rate curve from 0.529 to 0.602 for the open benchmark (see Fig. 4).

Similarly, Qi et al. (2016) proposed a method that forms a stronger tracker by hedging a set of weak trackers generate from several layers of a pre-trained CNN. They argued that features from a single layer cannot fully utilize the power of CNN. They concluded that an online decision-theoretical hedge algorithm was used to weigh each weak tracker and prove the effectiveness of the proposed hedged deep tracking algorithm (see Fig. 5).

While many existing trackers use deep networks, Yun et al. (2017) designed a tracker to achieve both light computation and satisfactory tracking accuracy, the tracker tracks the target by repetitive actions controlled by the action-decision network (ADNet), which is well pre-trained by supervised learning and reinforcement learning. Later, in online adaptation in Tracking, the tracking algorithm will be more robust against deformation. In the visual tracking experiment, ADNet had similar precision and success rate (64.6% AUC, Area Under Curve) with MDNet and C–COT.

Song et al. (2018) pointed out that trackers with current deep classification networks have two drawbacks: (a) positive samples are highly overlapped, and (b) positive and negative samples are highly imbalanced. A VITAL algorithm was proposed to solve the problem through adversarial learning. To handle the first problem, they use their network to identify the mask with robust features when they use a generative network to generate masks randomly. To handle the second problem, a high-order cost-sensitive loss is proposed to diminish the negative influences. The expected average overlap (EAO) for VITAL reaches 0.323 with high accuracy rank (Ar) and robustness rank (Rr), which are 1.63 and 2.17, respectively. Xu et al. (2019) proposed a new Group Feature Selection method for Discriminative Correlation

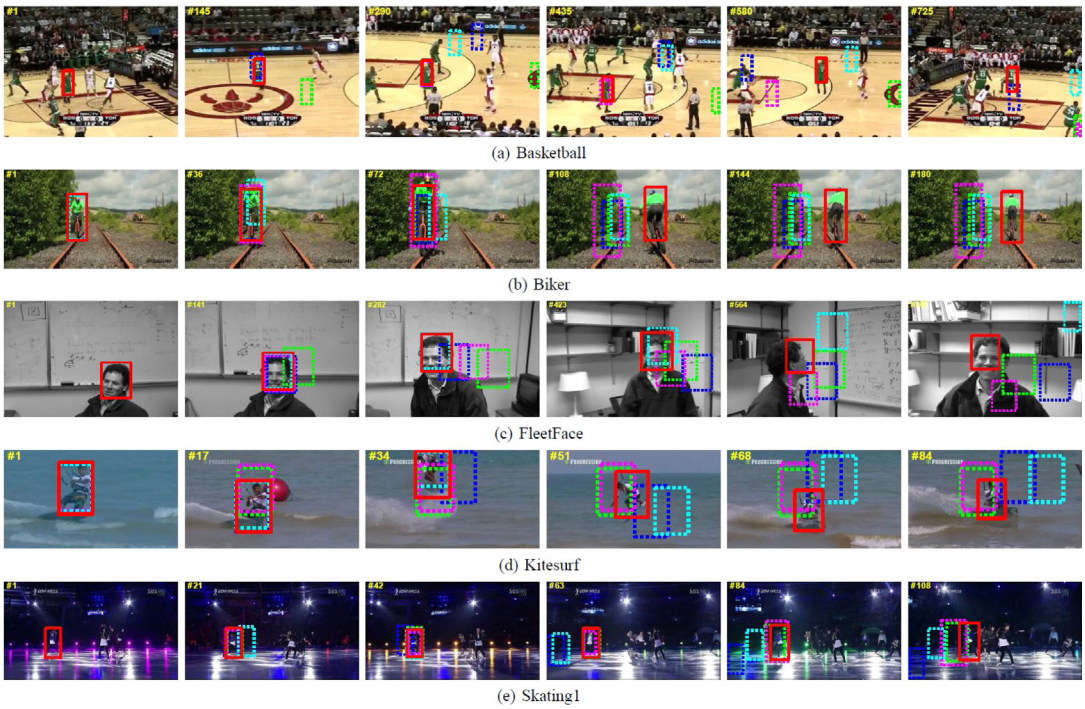

Fig. 4. Example of visual tracking (Wang, Li et al., 2015).

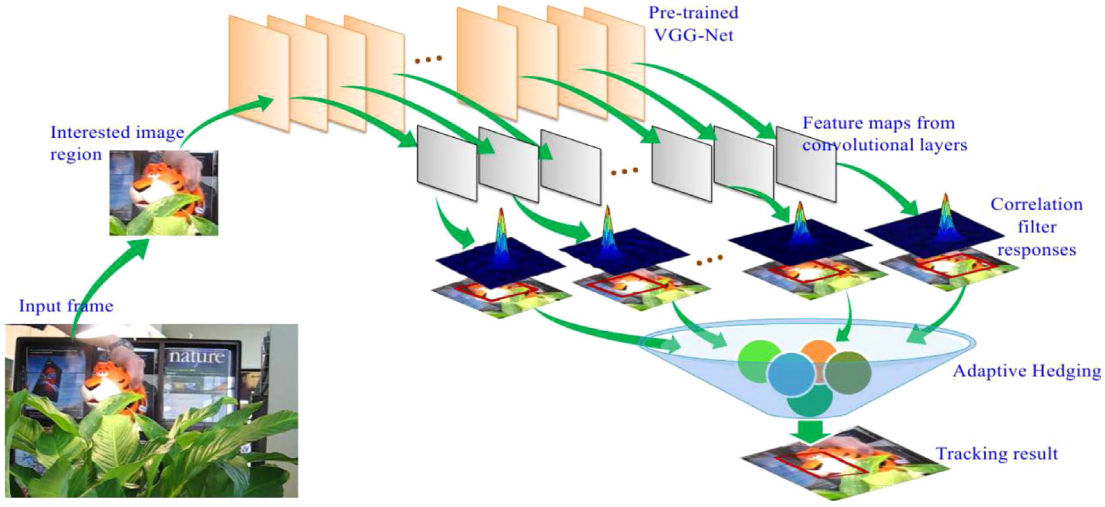

Fig. 5. Processes for tracking results by using DL backbone (Qi et al., 2016).

Filters (GFS-DCF) based on visual object tracking. GFS-DCF method can significantly improve the performance of a DCF tracker equipped with deep neural network features, with the AUC increasing from 55.49% to 63.07%. Lukezic et al. (2020) propose a discriminative single-shot segmentation tracker called D3S. They use those kinds of custom networks and construct two modules: GIM for segmentation and GEM for positioning to solve the problem. The AUC increases up to 72.8%. The Bounding Box's rough representation of the target will affect the performance, the background interference, and the long video robustness in the video segmentation task.

5. Semantic segmentation

CNN is still the dominant DL technique used in semantic segmentation. Long et al. (2015) proposed to use CNN to enable efficient end-to-end dense learning. They replace the last fully connected layer of vanilla CNN with a 1×1 convolutional layer to output a *heatmap*. Noh et al. (2015) proposed to use a *deconvolutional* network for semantic segmentation. The network contains a convolution network and a deconvolution network consisting of multiple deconvolutions and *uppooling* layers. Deconvolution is the reverse operation of Convolution, while *uppooling* is the reverse operation of pooling. The output of this network is a probability map that indicates classes are assigned to each pixel. They used VGG16 network architecture.

Hong et al. (2015) proposed to treat semantic segmentation tasks as two subtasks: classification and segmentation. For each task, they trained a separate CNN. Bridging layers were used to deliver class-specific information from classification networks to segmentation networks. It not only utilizes both image-level and pixel-level class labels but also improves segmentation efficiency since search space is reduced by learning class-specific activation maps. They firstly train classification networks using a vast number of images. After that, they fix the

classifier's parameters and jointly train bridging layers and segmentation networks using a small number of strongly annotated training data. They used VGG16 and named their method as *DecoupledNet*.

Face with the difficulty that the upper layers in feedforward networks are invariant to factors, Pinheiro et al. (2016) proposed a SharpMask method. It augmented feedforward nets through top-down refinement and combined itself with the DeepMask network. The experiments showed that the method performed outstandingly in both quality (i.e., 10%–20% improvement in recall accuracy averagely) and speed (i.e., 50% faster than the original DeepMask network, say below 0.8 s per image), and it can also be used in other pixel-labeling tasks.

Extending the success of faster R-CNN on object detection, He et al. (2017) applied it to the task of instance segmentation. In addition to the bounding box, Mask R-CNN generates a binary map that indicates whether a pixel belongs to an object or not. This is achieved by using a method called Realigning RoIPool (RoiAlign), which reaches 37.1AP and $60AP_{50}$ on COCO dataset. Chen et al. (2018) proposed DeepLabv3+ to combine the advantage of encoding multiscale contextual information of encoder–decoder structure and the advantage of capturing sharper object boundaries of deep neural networks. They also learned from the Xception model to achieve a faster and stronger encoder–decoder network. The test set performance of 89% in PASCAL VOC 2012 proved the method to be successful. Zhang et al. (2019) presented a class-agnostic segmentation network with few-shot learning called *Canet,* which reaches 49.9% meanIoU (intersection over union) on the COCO2014 dataset. The attention mechanism that solves the k-shot problem turns out to be more effective than non-learnable methods. An iterative optimization module that iteratively refines the predicted results is used (see Fig. 6).

Besides, it is worth mentioning that with the deepening of the above image segmentation researches, many medical image segmentation models have been proposed and applied in medical domains. Ronneberger et al. (2015) developed U-Net that was based on a fully convolutional network. By the strong use of data augmentation, they had effectively improved the accuracy of the results with a very limited training dataset. They also presented U-shaped architecture, which consisted of a contracting path to capture context and a symmetric expanding path that enabled precise localization to solve the problem of positioning medical images. Since then, researchers have proposed variants based on U-Net, such as V-Net (Milletari et al., 2016), UNet++ (Zhou et al., 2020). Rai and Chatterjee (2020) developed LU-Net by CNN with fewer layers to detect tumors in the brain.

6. Image restoration

Dong et al. (2016) developed a method for image super-resolution (SR), which could learn an end-to-end mapping between low-resolution (LR)/high-resolution (HR) images. This method is based on CNN, and it has three layers. The first layer extracts feature that maps from LR patches, and the second layer is used for mapping from these feature maps to HR feature maps. The last layer reconstructs HR by combining the predictions. This structure is lightweight and fast enough for online usage.

Burger et al. (2012) used a plain MLP to learn mapping from a noisy image to a noise-free image. This method keeps pace with other advanced denoising methods [such as Dabov et al. (2007)], and it is suitable for less extensively studied types of noise.

Lehtinen et al. (2018) proposed an innovative idea to recover images by merely looking at the corrupted examples without obtaining clean training targets. By using ML to map corrupted observations to clean signals and basic signal reconstruction algorithms, the researchers were able to reconstruct the signal from noise to clean. The result shows that the noisy targets have reached an average PSNR (Peak Signal to Noise Ratio) of 31.74 dB on the validation data. The network trained with clean targets has reached 31.77 dB, where the network works well in reconstructing the signal from the noise to the clean.

In the previous research, many scholars (Dosovitskiy & Brox, 2016; Lai et al., 2017; Ledig et al., 2017) have carried out a large number of pre-training with plenty of realistic images in order to improve the performance of deep convolutional network in image restoration. However, Ulyanov et al. (2018) showed that even without learning, the structure of a convolutional image generator could capture a large number of image statistics. Their method does not require a degradation modeling process and pre-training. However, it performs well in SR, inpainting, and denoising. As it requires a large number of iterations, the processing is relatively slow.

It faces difficulties in acquiring training datasets in some fields, for example, biomedical image data. Therefore, Krull et al. (2019) introduced NOISE2VOID (N2V), a training scheme that only requires single noisy acquisitions to train denoising CNNs. They proposed a blind-spot network, where the receptive field of each pixel excludes the pixel itself, thus preventing it from learning the identity. Thus, N2V could not remove the noise very well if the assumption of independence could not be satisfied; they show a new way for training the network to adapt to the fields that acquire limited or realistic low-resolution training datasets. Kim et al. (2020) designed a GAN-based joint SR and inverse tone-mapping (ITM) network (SR-ITM) called JSI-GAN. It consists of three task-specific subnets: an image reconstruction subnet, a detail restoration subnet, and a local contrast enhancement (LCE) subnet. When all subnet joint training is perfect, the quality of the predicted HR and high dynamic range result is increased, with a PSNR gain of at least 0.41 dB.

Event cameras perform better in sensing intensity changes than traditional cameras. However, reconstructing intensity images from event stream outputs is still in low resolution (LR), noisy, blurred, and unrealistic. Wang et al. (2020) proposed EventSR, an end-to-end pipeline that reconstructs LR images from event streams, enhances the image qualities, and up-samples the enhanced images. However, due to the lack of real GT images, their approach is largely unsupervised, deploying adversarial learning. However, based on the adversarial learning, EventSR increases PSNR to 47.68 dB on ESIM-RW dataset.

7. Analyses on recent developments and future research trends

The application of DL in CV showed significant development in the past decade. A clear evolution can be summarized, for which we considered three stages in general.

Early Stage (2012–2016)

With the advent of AlexNet (Krizhevsky et al., 2012), researchers used CNN for image classification in the early stage between 2012 and 2016. The neural network architecture is continually optimized by various cues to achieve increased accuracy. After VGG (Simonyan & Zisserman, 2015) was proposed, researchers branched their focus and started to explore basic application scenarios, including object detection, visual tracking, and semantic segmentation. Following this backbone, researchers developed various architectures. YOLOV1 (Redmon et al., 2016) utilizes GoogLeNet as the head for references in the object detection. Deeplab (Chen et al., 2015) modifies the semantic segmentation based on VGG16. By using ResNet (He et al., 2016), the accuracy in image classification has surpassed the recognition of human beings.

Middle Stage (2016–2019)

In the middle stage, a branch of researchers began to pursue the lightweight of parameters and neural networks with sufficient precision, such as MobileNet V1–V3 and ShuffleNet V1-2. This direction has been continued to the third stage, such as RegNet and GhostNet (Han et al., 2020). Another branch of researchers started to penetrate emerging techniques to various application scenarios. For example, in visual tracking, SiamRPN (Li et al., 2018) is based on the RPN idea in faster RCNN, which was developed for object detection. Besides, it could improve the model's accuracy by optimizing the previous models

Fig. 6. Example of Semantic Segmentation (He et al., 2017).

or upgrading the physical hardware. For example, He et al. (2016) compared plain and residual networks, and the philosophy of VGG nets mainly inspires the plain baselines they used.

Latest Stage (2019–now) and Research Trends

On the basis of the analysis of literature trends in the past 2 years, we summarized four research trends for future works.

(1) **Exploration of network types and architecture**: The types of networks tend to be enriched. More types, such as Siamese neural network (SNN), Recurrent neural network (RNN), Generative adversarial network (GAN), and Custom networks, came out. Semi-supervised learning has gradually been centered in recent studies for CV scenarios. In other words, we witnessed an evolution process from supervised learning to semi-supervised learning.

(2) **Enhancement of more specific scenarios of application**: As the techniques for CV have gradually matured, application scenarios become more specific, such as the application samples of GAN in 3D semantic segmentation, face recognition, action recognition, stylization, and machine creation. As a result, researchers tend to refine the techniques and cues developed in conventional CV and improve their performance for a more specific subdivision.

(3) **Combinatorial application of CV with other ML domains**: The studies in CV tend to incorporate with other ML domains for combinatorial applications apart from just circumstancing in their fields. For example, chatbots use more NLP techniques to improve the accuracy of responses (Adamopoulou & Moussiades, 2020), where detecting what is happening in the conversation is difficult. A big challenge of chatbots is to simulate real communication by understanding the user's inner activities at the next level. Through motion analyses and facial expression recognition, chatbots could understand users' emotions from micro-expressions and even analyze them in combination of psychology theories.

(4) **Penetrating CV to broader application domains**: Studies on crossover applications are enriched. CV application studies have been extended to the medical domain, such as cancer detection by semantic segmentation (Mehrotra et al., 2020); industrial predictions, such as forecasting petroleum production (Al-Shabandar et al., 2021); and archaeology, such as recovering historical letters (Dambrogio et al., 2021).

In terms of technical sides, we considered two directions for future works.

(1) **Model Visualization and Interpretability**: ML, including DL, is usually regarded as a black box. Conventional designs of techniques for CV could not process the large size of the dataset efficiently. The DL-based end-to-end learning mechanism offers the opportunity to be less concerned about the large size of the dataset. The DL model could be trained in very large datasets and then used in sensitive or unrepeatable application scenarios, e.g., medical surgery. Therefore, visualizing and interpreting the DL model are necessary for outsiders (e.g., medical doctors or surgeons) to understand the technical basis for the determination. The first direction is to reinforce the visualization and interpretability of DL models.

(2) **Model Scalability**: Many DL models have been reported at present, and their structures are becoming complicated. Besides, training the DL model is time consuming. Thus, whether a model is scalable easily becomes a criterion for evaluating this model. With limited resources in time and data, a model could be trained with a simple structure. The complicity of the model could be extended for fulfilling more needs, with an increased degree of accuracy. For instance, Facebook (RegNet) and Google (EfficientNet) have proposed their scalable models and method for space design. This direction is worthy of further exploration in the future.

Despite the promising and in some cases, impressive results of CV, challenges do remain in using DL for CV. Many people are concerned that ethical and privacy issues will also become more prominent. Redmon and Farhadi (2018) stopped developing YOLO, as they thought their work might include possible misuses. They commented in their paper "computer vision is already being put to questionable use and as researchers, we have a responsibility to at least consider the harm our work might be doing and think of ways to mitigate it. We owe the world that much". There is growing alarm over the use of altered videos online, especially those known as Deepfakes (AI-generated videos). Bloomberg Quicktake (2018) uploaded a video on YouTube that showed how the fake face process used CV technology which the people feel concerned and worried. In fact, with open-source code and the increasing power of personal computers, Deepfakes can mimic biometric data and can potentially trick systems that reply on face, voice, or vein recognition.

Nevertheless, the protection of privacy is an important issue in modern information society. Young and Quan-Haase (2013) worked on information revelation and privacy protection strategies. They found that users disclose information despite their privacy concerns because they have made a conscious effort to protect themselves against potential violations by establishing who has access to their data. On the other hand, academic efforts are trying to enhance the development of technologies, and several scholars have begun research in this area (Agarwal, Farid, El-Gaaly et al., 2020; Agarwal, Farid, Fried et al., 2020; Hsu et al., 2020). Meanwhile, Facebook launched contest of Deepfake Detection Challenge (DFDC) in partnership with Microsoft and academic in 2019. They believe the DFDC results contribute to this effort and build a robust response to the emergent threat Deepfakes pose globally (Kaggle, 2019).

8. Concluding remarks

For the application scenario of CV, we identified eight emerging DL techniques in this paper, including AlexNet, VGGNet, GoogLeNet & Inception, ResNet, DenseNet, MobileNets, EfficientNet, and RegNet. We investigated their origins and provided a critical review of representative research outputs from 2014. We focused on four key tasks of CV, including recognition, visual tracking, semantic segmentation, and image restoration. We also investigated and emphasized the performance of these techniques in each scenario. We summarized the recent developments into three stages and depicted future research directions in terms of applications and techniques.

CRediT authorship contribution statement

Junyi Chai: Conceptualization, Methodology, Validation, Resources, Writing – original draft, Writing - review & editing, Supervision, Project administration, Funding acquisition. **Hao Zeng:** Methodology, Writing – original draft. **Anming Li:** Conceptualization, Resources, Writing – original draft. **Eric W.T. Ngai:** Writing – review & editing, Validation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

We thank the Editor-in-Chief, the Associate Editor, and three anonymous reviewers for their constructive comments that have helped to improve the paper significantly. This study is supported financially in part by College Research Grant of BNU-HKBU United International College, China, and Centre for Evaluation Studies with Beijing Normal University at Zhuhai.

References

Adamopoulou, E., & Moussiades, L. (2020). Chatbots: History, technology, and applications. *Machine Learning with Applications*, *2*, Article 100006. http://dx.doi.org/10.1016/j.mlwa.2020.100006.

Agarwal, S., Farid, H., El-Gaaly, T., & Lim, S. N. (2020). Detecting deep-fake videos from appearance and behavior. In *Proceedings of the 2020 ieee international workshop on information forensics and security (wifs)* (pp. 1–6). USA: http://dx.doi.org/10.1109/WIFS49906.2020.9360904.

Agarwal, S., Farid, H., Fried, O., & Agrawala, M. (2020). Detecting deep-fake videos from phoneme-viseme mismatches. In *Proceedings of the 2020 ieee/cvf conference on computer vision and pattern recognition workshops (cvprw)* (pp. 2814–2822). USA: http://dx.doi.org/10.1109/CVPRW50498.2020.00338.

Al-Shabandar, R., Jaddoa, A., Liatsis, P., & Hussain, A. J. (2021). A deep gated recurrent neural network for petroleum production forecasting. *Machine Learning with Applications*, *3*, Article 100013. http://dx.doi.org/10.1016/j.mlwa.2020.100013.

Altan, A., Karasu, S., & Zio, E. (2021). A new hybrid model for wind speed forecasting combining long short-term memory neural network, decomposition methods and grey wolf optimizer. *Applied Soft Computing*, *100*, Article 106996. http://dx.doi.org/10.1016/j.asoc.2020.106996.

Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., Santamaría, J., Fadhel, M. A., Al-Amidie, M., & Farhan, L. (2021). Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions. *Journal of Big Data*, *8*(1), 53. http://dx.doi.org/10.1186/s40537-021-00444-8.

Bloomberg Quicktake (2018). It's getting harder to spot a deep fake video. Retrieved from https://www.youtube.com/watch?v=gLoI9hAX9dw (Accessed July 10, 2021).

Bochkovskiy, A., Wang, C. Y., & Liao, H.-Y. M. (2020). YOLOV4: Optimal speed and accuracy of object detection. arxiv preprint arXiv:2004.10934 [Cs, Eess]. http://arxiv.org/abs/2004.10934.

Burger, H. C., Schuler, C. J., & Harmeling, S. (2012). Image denoising: Can plain neural networks compete with BM3D?. In *Proceedings of the 2012 ieee conference on computer vision and pattern recognition* (pp. 2392–2399). Rhode Island: http://dx.doi.org/10.1109/CVPR.2012.6247952.

Cai, Z., Fan, Q., Feris, R. S., & Vasconcelos, N. (2016). A unified multiscale deep convolutional neural network for fast object detection. In *Proceedings of the 14th european conference on computer vision (eccv)* (pp. 354–370). Netherlands: http://dx.doi.org/10.1007/978-3-319-46493-0_22.

Chai, J., & Li, A. (2019). Deep learning in natural language processing: A state-of-the-art survey. In *Proceedings of the 2019 international conference on machine learning and cybernetics (icmlc)* (pp. 1–6). Japan: http://dx.doi.org/10.1109/ICMLC48188.2019.8949185.

Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., & Yuille, A. L. (2015). Semantic image segmentation with deep convolutional nets and fully connected CRFs. arxiv preprint arXiv:1412.7062 [Cs]. http://arxiv.org/abs/1412.7062.

Chen, L. C., Zhu, Y., Papandreou, G., Schroff, F., & Adam, H. (2018). Encoder–decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the 15th european conference on computer vision (eccv)* (pp. 801–818). Germany: http://dx.doi.org/10.1007/978-3-030-01234-2_49.

Chollet, F. (2017). Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the 2017 ieee conference on computer vision and pattern recognition (cvpr)* (pp. 1251–1258). USA: http://dx.doi.org/10.1109/CVPR.2017.195.

Dabov, K., Foi, A., Katkovnik, V., & Egiazarian, K. (2007). Image denoising by sparse 3-D transform-domain collaborative filtering. *IEEE Transactions on Image Processing*, *16*(8), 2080–2095. http://dx.doi.org/10.1109/TIP.2007.901238.

Dai, J., Li, Y., He, K., & Sun, J. (2016). R-FCN: Object detection via region-based fully convolutional networks. In *Proceedings of the 30th international conference on neural information processing systems (nips)*, Spain, (pp. 379–387).

Dambrogio, J., Ghassaei, A., Smith, D. S., Jackson, H., Demaine, M. L., Davis, G., Mills, D., Ahrendt, R., Akkerman, N., van der Linden, D., & Demaine, E. D. (2021). Unlocking history through automated virtual unfolding of sealed documents imaged by X-ray microtomography. *Nature Communications*, *12*(1), 1184. http://dx.doi.org/10.1038/s41467-021-21326-w.

Dong, C., Loy, C. C., He, K., & Tang, X. (2016). Image super-resolution using deep convolutional networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *38*(2), 295–307. http://dx.doi.org/10.1109/TPAMI.2015.2439281.

Dosovitskiy, A., & Brox, T. (2016). Inverting visual representations with convolutional networks. In *Proceedings of the 2016 ieee conference on computer vision and pattern recognition (cvpr)* (pp. 4829–4837). USA: http://dx.doi.org/10.1109/CVPR.2016.522.

Elad, M., & Aharon, M. (2006). Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Transactions on Image Processing*, *15*(12), 3736–3745. http://dx.doi.org/10.1109/TIP.2006.881969.

Fu, C.-Y., Liu, W., Ranga, A., Tyagi, A., & Berg, A. C. (2017). DSSD: Deconvolutional single shot detector. arxiv preprint arxiv:1701.06659 [Cs]. http://arxiv.org/abs/1701.06659.

Gando, G., Yamada, T., Sato, H., Oyama, S., & Kurihara, M. (2016). Fine-tuning deep convolutional neural networks for distinguishing illustrations from photographs. *Expert Systems with Applications*, *66*, 295–301. http://dx.doi.org/10.1016/j.eswa.2016.08.057.

Girshick, R. (2015). Fast r-cnn. In *Proceedings of the ieee international conference on computer vision (iccv)* (pp. 1440–1448). Chile: http://dx.doi.org/10.1109/iccv.2015.169.

Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the 2014 ieee conference on computer vision and pattern recognition* (pp. 580–587). USA: http://dx.doi.org/10.1109/CVPR.2014.81.

Guo, Y., Liu, Y., Oerlemans, A., Lao, S., Wu, S., & Lew, M. S. (2016). Deep learning for visual understanding: A review. *Neurocomputing*, *187*, 27–48. http://dx.doi.org/10.1016/j.neucom.2015.09.116.

Han, K., Wang, Y., Tian, Q., Guo, J., Xu, C., & Xu, C. (2020). GhostNet: More features from cheap operations. In *Proceedings of the 2020 ieee/cvf conference on computer vision and pattern recognition (cvpr)* (pp. 1577–1586). USA: http://dx.doi.org/10.1109/CVPR42600.2020.00165.

He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask r-cnn. In *Proceedings of the 2017 ieee international conference on computer vision (iccv)* (pp. 2961–2969). Italy: http://dx.doi.org/10.1109/ICCV.2017.322.

He, K., Zhang, X., Ren, S., & Sun, J. (2015). Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *37*(9), 1904–1916. http://dx.doi.org/10.1109/TPAMI.2015.2389824.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition resnet. In *Proceedings of the 2016 ieee conference on computer vision and pattern recognition (cvpr)* (pp. 770–778). USA: http://dx.doi.org/10.1109/CVPR.2016.90.

Hong, S., Noh, H., & Han, B. (2015). Decoupled deep neural network for semi-supervised semantic segmentation. In *Proceedings of the 28th international conference on neural information processing systems (nips)*, Canada (pp. 1495-1503).

Howard, A., Sandler, M., Chu, G., Chen, L.-C., Chen, B., Tan, M., Wang, W., Zhu, Y., Pang, R., Vasudevan, V., Le, Q. V., & Adam, H. (2019). Searching for MobileNetV3. In *Proceedings of the 2019 ieee/cvf international conference on computer vision (iccv)* (pp. 1314–1324). Korea (South): http://dx.doi.org/10.1109/ICCV.2019.00140.

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., & Adam, H. (2017). MobileNets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861 [Cs]. http://arxiv.org/abs/1704.04861.

Hsu, C.-C., Zhuang, Y.-X., & Lee, C.-Y. (2020). Deep fake image detection based on pairwise learning. *Applied Sciences*, *10*(1), 370. http://dx.doi.org/10.3390/app10010370.

Huang, J., Chai, J., & Cho, S. (2020). Deep learning in finance and banking: A literature review and classification. *Frontiers of Business Research in China*, *14*, 1–24. http://dx.doi.org/10.1186/s11782-020-00082-6.

Huang, Y., Cheng, Y., Bapna, A., Firat, O., Chen, D., Chen, M., Lee, H., Ngiam, J., Le, Q. V., & Wu, Y. (2019). Gpipe: Efficient training of giant neural networks using pipeline parallelism. In *Proceedings of the 33rd conference on neural information processing systems (neurips), Vol. 32*, Canada, (pp. 103-112).

Huang, G., Liu, Z., van der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. In *Proceedings of the 2017 ieee conference on computer vision and pattern recognition (cvpr)* (pp. 2261–2269). USA: http://dx.doi.org/10.1109/CVPR.2017.243.

Kaggle (2019). Deepfake detection challenge | kaggle. Retrieved from https://www.kaggle.com/c/deepfake-detection-challenge (Accessed July 10, 2021).

Kim, S. Y., Oh, J., & Kim, M. (2020). Jsi-gan: Gan-based joint super-resolution and inverse tone-mapping with pixel-wise task-specific filters for uhd hdr video. *Proceedings of the AAAI Conference on Artificial Intelligence, USA*, *34*(07), 11287–11295. http://dx.doi.org/10.1609/aaai.v34i07.6789.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Proceedings of the 25th international conference on neural information processing systems (nips), Vol. 1*, USA (pp. 1097-1105).

Krull, A., Buchholz, T.-O., & Jug, F. (2019). Noise2Void—LEarning denoising from single noisy images. In *Proceedings of the 2019 ieee/cvf conference on computer vision and pattern recognition (cvpr)* (pp. 2124–2132). USA: http://dx.doi.org/10.1109/CVPR.2019.00223.

Kumar, A., Walia, G. S., & Sharma, K. (2020). Recent trends in multicue based visual tracking: A review. *Expert Systems with Applications*, *162*, Article 113711. http://dx.doi.org/10.1016/j.eswa.2020.113711.

Lai, W.-S., Huang, J. B., Ahuja, N., & Yang, M. H. (2017). Deep laplacian pyramid networks for fast and accurate super-resolution. In *Proceedings of the 2017 ieee conference on computer vision and pattern recognition (cvpr)* (pp. 5835–5843). USA: http://dx.doi.org/10.1109/CVPR.2017.618.

Lapuschkin, S., Binder, A., Montavon, G., Muller, K.-R., & Samek, W. (2016). The LRP toolbox for artificial neural networks. *Journal of Machine Learning Research*, *17*(114), 1–5, http://jmlr.org/papers/v17/15-618.html.

Law, H., & Deng, J. (2018). CornerNet: Detecting objects as paired keypoints. In *Proceedings of the 15th european conference on computer vision (eccv), Vol. 11218* (pp. 734–750). Germany: http://dx.doi.org/10.1007/978-3-030-01264-9_45.

Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., & Wang, Z. (2017). Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the 2017 ieee conference on computer vision and pattern recognition (cvpr)* (pp. 105–114). USA: http://dx.doi.org/10.1109/CVPR.2017.19.

Lehtinen, J., Munkberg, J., Hasselgren, J., Laine, S., Karras, T., Aittala, M., & Aila, T. (2018). Noise2Noise: Learning image restoration without clean data. In *Proceedings of the 35th international conference on machine learning (icml), Vol. 80*, (Stockholm SWEDEN) (pp. 2965-2974).

Li, B., Yan, J., Wu, W., Zhu, Z., & Hu, X. (2018). High performance visual tracking with siamese region proposal network. In *Proceedings of the 2018 ieee/cvf conference on computer vision and pattern recognition (cvpr)* (pp. 8971–8980). USA: http://dx.doi.org/10.1109/CVPR.2018.00935.

Lienhart, R., & Maydt, J. (2002). An extended set of haar-like features for rapid object detection. In *Proceedings. Proceedings of the international conference on image processing, Vol. 1* (pp. I–900–I–903). USA: http://dx.doi.org/10.1109/ICIP.2002.1038171.

Lin, M., Chen, Q., & Yan, S. (2014). Network in network. In *Proceedings of the 2014 international conference on learning representations (iclr)*, Canada.

Lin, T.-Y., Dollar, P., Girshick, R., He, K., Hariharan, B., & Belongie, S. (2017). Feature pyramid networks for object detection. In *Proceedings of the 2017 ieee conference on computer vision and pattern recognition (cvpr)* (pp. 936–944). USA: http://dx.doi.org/10.1109/CVPR.2017.106.

Lin, T. Y., Goyal, P., Girshick, R., He, K., & Dollár, P. (2017). Focal loss for dense object detection. In *Proceedings of the 2017 ieee international conference on computer vision (iccv)* (pp. 2999–3007). Italy: http://dx.doi.org/10.1109/iccv.2017.324.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C. (2016). SSD: Single shot multibox detector. In *Proceedings of the european conference on computer vision (eccv), Vol. 9905* (pp. 21–37). Netherlands: http://dx.doi.org/10.1007/978-3-319-46448-0_2.

Long, X., Deng, K., Wang, G., Zhang, Y., Dang, Q., Gao, Y., Shen, H., Ren, J., Han, S., Ding, E., & Wen, S. (2020). PP-YOLO: An effective and efficient implementation of object detector. arXiv preprint arXiv:2007.12099 [Cs]. http://arxiv.org/abs/2007.12099.

Long, J., Shelhamer, E., & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In *Proceedings of the 2015 ieee conference on computer vision and pattern recognition (cvpr)* (pp. 3431–3440). USA: http://dx.doi.org/10.1109/cvpr.2015.7298965.

Lukezic, A., Matas, J., & Kristan, M. (2020). D3S – a discriminative single shot segmentation tracker. In *Proceedings of the 2020 ieee/cvf conference on computer vision and pattern recognition (cvpr)* (pp. 7131–7140). USA: http://dx.doi.org/10.1109/CVPR42600.2020.00716.

Mehrotra, R., Ansari, M. A., Agrawal, R., & Anand, R. S. (2020). A transfer learning approach for AI-based classification of brain tumors. *Machine Learning with Applications*, *2*, Article 100003. http://dx.doi.org/10.1016/j.mlwa.2020.100003.

Milletari, F., Navab, N., & Ahmadi, S.-A. (2016). V-NEt: Fully convolutional neural networks for volumetric medical image segmentation. In *Proceedings of the 2016 fourth international conference on 3d vision (3dv)* (pp. 565–571). USA: http://dx.doi.org/10.1109/3DV.2016.79.

Muzammel, M., Salam, H., Hoffmann, Y., Chetouani, M., & Othmani, A. (2020). AudVowelConsNet: A phoneme-level based deep CNN architecture for clinical depression diagnosis. *Machine Learning with Applications*, *2*, Article 100005. http://dx.doi.org/10.1016/j.mlwa.2020.100005.

Noh, H., Hong, S., & Han, B. (2015). Learning deconvolution network for semantic segmentation. In *Proceedings of the 2015 ieee international conference on computer vision (iccv)* (pp. 1520–1528). Chile: http://dx.doi.org/10.1109/iccv.2015.178.

Pinheiro, P. O., Lin, T.-Y., Collobert, R., & Dollàr, P. (2016). Learning to refine object segments. (pp. 75–91). Netherlands: http://dx.doi.org/10.1007/978-3-319-46448-0_5.

Qi, Y., Zhang, S., Qin, L., Yao, H., Huang, Q., Lim, J., & Yang, M. H. (2016). Hedged deep tracking. In *Proceedings of the ieee conference on computer vision and pattern recognition (cvpr)* (pp. 4303–4311). USA: http://dx.doi.org/10.1109/CVPR.2016.466.

Radosavovic, I., Kosaraju, R. P., Girshick, R., He, K., & Dollár, P. (2020). Designing network design spaces. In *Proceedings of the 2020 ieee/cvf conference on computer vision and pattern recognition (cvpr)* (pp. 10425–10433). USA: http://dx.doi.org/10.1109/CVPR42600.2020.01044.

Rai, H. M., & Chatterjee, K. (2020). Detection of brain abnormality by a novel lu-net deep neural CNN model from MR images. *Machine Learning with Applications*, *2*, Article 100004. http://dx.doi.org/10.1016/j.mlwa.2020.100004.

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *Proceedings of the 2016 ieee conference on computer vision and pattern recognition (cvpr)* (pp. 779–788). USA: http://dx.doi.org/10.1109/CVPR.2016.91.

Redmon, J., & Farhadi, A. (2017). YOLO9000: better, faster, stronger. *Proceedings of the 2017 ieee conference on computer vision and pattern recognition (cvpr)*, 6517–6525. http://dx.doi.org/10.1109/CVPR.2017.690.

Redmon, J., & Farhadi, A. (2018). Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767. http://arxiv.org/abs/1804.02767.

Ren, S., He, K., Girshick, R., & Sun, J. (2017). Faster r-cnn: Towards real-time object detection with region proposal networks. In *Proceedings of the 2017 ieee conference on computer vision and pattern recognition (cvpr)* (pp. 6517–6525). USA: http://dx.doi.org/10.1109/TPAMI.2016.2577031.

Ronneberger, O., Fischer, P., & Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. *Proceedings of the 18th international conference on medical image computing and computer-assisted intervention, Vol. 9351*, 234–241. http://dx.doi.org/10.1007/978-3-319-24574-4_28.

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L. C. (2018). Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the 2018 ieee conference on computer vision and pattern recognition (cvpr)* (pp. 4510–4520). USA: http://dx.doi.org/10.1109/CVPR.2018.00474.

Simonyan, K., & Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. In *Proceedings of the 3rd international conference on learning representations (iclr2015)*, USA.

Song, Y., Ma, C., Wu, X., Gong, L., Bao, L., Zuo, W., Shen, C., Lau, R. W., & Yang, M. H. (2018). Vital: Visual tracking via adversarial learning. In *Proceedings of the ieee conference on computer vision and pattern recognition (cvpr)* (pp. 8990–8999). USA: http://dx.doi.org/10.1109/CVPR.2018.00937.

Szegedy, C., Ioffe, S., Vanhoucke, V., & Alemi, A. 2017. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Proceedings of the thirty-first aaai conference on artificial intelligence*, USA (pp. 4278–4284).

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., & Rabinovich, A. (2015). Going deeper with convolutions. In *Proceedings of the 2015 ieee conference on computer vision and pattern recognition (cvpr)* (pp. 1–9). USA: http://dx.doi.org/10.1109/CVPR.2015.7298594.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In *Proceedings of the 2016 IEEE conference on computer vision and pattern recognition (cvpr)* (pp. 2818–2826). USA: http://dx.doi.org/10.1109/CVPR.2016.308.

Tan, M., & Le, Q. (2019). Efficientnet: Rethinking model scaling for convolutional neural networks. In *Proceedings of the 36thinternational conference on machine learning (icml), Vol. 97*, USA (pp. 6105-6114).

Tian, Z., Shen, C., Chen, H., & He, T. (2019). Fcos: Fully convolutional one-stage object detection. In *Proceedings of the 2019 ieee/cvf international conference on computer vision (cvpr)* (pp. 9626–9635). USA: http://dx.doi.org/10.1109/iccv.2019.00972.

Ulyanov, D., Vedaldi, A., & Lempitsky, V. (2018). Deep image prior. *International Journal of Computer Vision*, *128*(7), 1867–1888. http://dx.doi.org/10.1007/s11263-020-01303-4.

Viola, P., & Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. In *Proceedings of the 2001 ieee computer society conference on computer vision and pattern recognition (cvpr)*. USA: http://dx.doi.org/10.1109/CVPR.2001.990517.

Walia, G. S., & Kapoor, R. (2016). Recent advances on multicue object tracking: A survey. *The Artificial Intelligence Review*, *46*(1), 1–39. http://dx.doi.org/10.1007/s10462-015-9454-6.

Wang, L., Kim, T. K., & Yoon, K. J. (2020). Eventsr: From asynchronous events to image reconstruction, restoration, and super-resolution via end-to-end adversarial learning. In *Proceedings of the 2020 ieee/cvf conference on computer vision and pattern recognition (cvpr)* (pp. 8312–8322). USA: http://dx.doi.org/10.1109/cvpr42600.2020.00834.

Wang, N., Li, S., Gupta, A., & Yeung, D. Y. (2015). Transferring rich feature hierarchies for robust visual tracking. arXiv preprint arXiv:1501.04587 http://arxiv.org/abs/1501.04587.

Wang, L., Liu, T., Wang, G., Chan, K. L., & Yang, Q. (2015). Video tracking using learned hierarchical features. *IEEE Transactions on Image Processing*, *24*(4), 1424–1435. http://dx.doi.org/10.1109/TIP.2015.2403231.

Wang, N., & Yeung, D. Y. (2013). Learning a deep compact image representation for visual Tracking. In *Proceedings of the 27th annual conference on neural information processing systems, Vol. 1*, USA (pp. 809-817).

Xie, J., Xu, L., & Chen, E. (2012). Image denoising and inpainting with deep neural networks. *Advances in Neural Information Processing Systems*, *25*, 341–349.

Xu, T., Feng, Z.-H., Wu, X.-J., & Kittler, J. (2019). Joint group feature selection and discriminative filter learning for robust visual object tracking. In *Proceedings of the 2019 ieee/cvf international conference on computer vision (iccv)* (pp. 7949–7959). Korea (South): http://dx.doi.org/10.1109/ICCV.2019.00804.

Xu, S., Wang, J., Shou, W., Ngo, T., Sadick, A.-M., & Wang, X. (2020). Computer vision techniques in construction: A critical review. *Archives of Computational Methods in Engineering*, http://dx.doi.org/10.1007/s11831-020-09504-3, 2020.

Yang, Z., Liu, S., Hu, H., Wang, L., & Lin, S. (2019). RepPoints: Point set representation for object detection. In *Proceedings of the 2019 ieee/cvf international conference on computer vision (iccv)* (pp. 9656–9665). Korea (South): http://dx.doi.org/10.1109/ICCV.2019.00975.

Ye, L., Liu, Z., & Wang, Y. (2020). Dual convolutional LSTM network for referring image segmentation. *IEEE Transactions on Multimedia*, *22*(12), 3224–3235. http://dx.doi.org/10.1109/TMM.2020.2971171.

Young, A. L., & Quan-Haase, A. (2013). Privacy protection strategies on facebook: The internet privacy paradox revisited. *Information, Communication & Society*, *16*(4), 479–500. http://dx.doi.org/10.1080/1369118X.2013.777757.

Yun, S., Choi, J., Yoo, Y., Yun, K., & Choi, J. Y. (2017). Action-decision networks for visual tracking with deep reinforcement learning. In *Proceedings of the 2017 ieee conference on computer vision and pattern recognition (cvpr)* (pp. 1349–1358). USA: http://dx.doi.org/10.1109/CVPR.2017.148.

Zeiler, M. D., & Fergus, R. (2014). Visualizing and understanding convolutional networks. In *Proceedings of the 13th european conference on computer vision (eccv), Vol. 8689* (pp. 818–833). Switzerland: http://dx.doi.org/10.1007/978-3-319-10590-1_53.

Zhang, C., Lin, G., Liu, F., Yao, R., & Shen, C. (2019). Canet: Class-agnostic segmentation networks with iterative refinement and attentive few-shot learning. In *Proceedings of the 2019 ieee/cvf conference on computer vision and pattern recognition (cvpr)* (pp. 5212–5221). USA: http://dx.doi.org/10.1109/cvpr.2019.00536.

Zhang, K., Liu, Q., Wu, Y., & Yang, M. H. (2016). Robust visual tracking via convolutional networks without training. *IEEE Transactions on Image Processing*, *25*(4), 1779–1792. http://dx.doi.org/10.1109/TIP.2016.2531283.

Zhao, Z., Jiao, L., Zhao, J., Gu, J., & Zhao, J. (2017). Discriminant deep belief network for high-resolution SAR image classification. *Pattern Recognition*, *61*, 686–701. http://dx.doi.org/10.1016/j.patcog.2016.05.028.

Zhou, Z., Siddiquee, M. M. R., Tajbakhsh, N., & Liang, J. (2020). Unet++: Redesigning skip connections to exploit multiscale features in image segmentation. *IEEE Transactions on Medical Imaging*, *39*(6), 1856–1867. http://dx.doi.org/10.1109/TMI.2019.2959609.

Zhou, X., Wang, D., & Krähenbühl, P. (2019). Objects as points. arXiv preprint arXiv:1904.07850 [Cs]. http://arxiv.org/abs/1904.07850.