##### Classify abstract

\* LeNet,1998

Multilayer neural networks trained with the back-propagation algorithm constitute the best example of a successful gradient based learning technique. Given an appropriate network architecture, gradient-based learning algorithms can be used to synthesize a complex decision surface that can classify high-dimensional patterns, such as handwritten characters, with minimal preprocessing. This paper reviews various methods applied to handwritten character recognition and compares them on a standard handwritten digit recognition task. Convolutional neural networks, which are specifically designed to deal with the variability of 2D shapes, are shown to outperform all other techniques. Real-life document recognition systems are composed of multiple modules including field extraction, segmentation recognition, and language modeling. A new learning paradigm, called graph transformer networks (GTN), allows such multimodule systems to be trained globally using gradient-based methods so as to minimize an overall performance measure. Two systems for online handwriting recognition are described. Experiments demonstrate the advantage of global training, and the flexibility of graph transformer networks.A graph transformer network for reading a bank cheque is also described. It uses convolutional neural network character recognizers combined with global training techniques to provide record accuracy on business and personal cheques. It is deployed commercially and reads several million cheques per day.

\* AlexNet,2012

We trained a large, deep convolutional neural network to classify the 1.2 million

high-resolution images in the ImageNet LSVRC-2010 contest into the 1000 different classes. On the test data, we achieved top-1 and top-5 error rates of 37.5% and 17.0% which is considerably better than the previous state-of-the-art. The neural network, which has 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. To make training faster, we used non-saturating neurons and a very efficient GPU implementation of the convolution operation. To reduce overfitting in the fully-connected layers we employed a recently-developed regularization method called “dropout” that proved to be very effective. We also entered a variant of this model in the ILSVRC-2012 competition and achieved a winning top-5 test error rate of 15.3%, compared to 26.2% achieved by the second-best entry.

\* VGG,2015

In this work we investigate the effect of the convolutional network depth on its accuracy in the large-scale image recognition setting. Our main contribution is a thorough evaluation of networks of increasing depth using an architecture  
with very small (3×3) convolution filters, which shows that a significant improvement on the prior-art configurations can be achieved by pushing the depth to 16–19 weight layers.These findings were the basis of our ImageNet Challenge 2014 submission, where our team secured the first and the second places in the localisation and classification tracks respectively. We also show that our representations generalise well to other datasets, where they achieve state-of-the-art results. We have made our two best-performing ConvNet models publicly available to facilitate further research on the use of deep visual representations in computer vision.

\* GoogLeNet,2015

We propose a deep convolutional neural network architecture codenamed Inception that achieves the new state of the art for classification and detection in the ImageNet Large-Scale Visual Recognition Challenge 2014 (ILSVRC14). The main hallmark of this architecture is the improved utilization of the computing resources inside the network. By a carefully crafted design, we increased the depth and width of the network while keeping the computational budget constant. To optimize quality, the architectural decisions were based on the Hebbian principle and the intuition of multi-scale processing. One particular incarnation used in our submission for ILSVRC14 is called GoogLeNet, a 22 layers deep network, the quality of which is assessed in the context of classification and detection.

\* ResNet,2015

Deeper neural networks are more difficult to train. We present a residual learning framework to ease the training of networks that are substantially deeper than those used previously. We explicitly reformulate the layers as learning residual functions with reference to the layer inputs, in stead of learning unreferenced functions. We provide comprehensive empirical evidence showing that these residual networks are easier to optimize, and can gain accuracy from considerably increased depth. On the ImageNet dataset we evaluate residual nets with a depth of up to 152 layers — 8×

deeper than VGG nets but still having lower complexity. An ensemble of these residual nets achieves 3.57% error on the ImageNet test set. This result won the 1st place on the ILSVRC 2015 classification task. We also present analysis on CIFAR-10 with 100 and 1000 layers.The depth of representations is of central importance

for many visual recognition tasks. Solely due to our extremely deep representations, we obtain a 28% relative improvement on the COCO object detection dataset. Deep

residual nets are foundations of our submissions to ILSVRC & COCO 2015 competitions1 , where we also won the 1st places on the tasks of ImageNet detection, ImageNet localization, COCO detection, and COCO segmentation.

\* EffectNet,2019

Convolutional Neural Networks (ConvNets) are commonly developed at a fixed resource budget, and then scaled up for better accuracy if more resources are available. In this paper, we systematically study model scaling and identify that carefully balancing network depth, width, and resolution can lead to better performance. Based on this observation, we propose a new scaling method that uniformly scales all dimensions of depth/width/resolution using a simple yet highly effective compound coefficient. We demonstrate the effectiveness of this method on scaling up MobileNets and ResNet.To go even further, we use neural architecture search to design a new baseline network and scale it up to obtain a family of models, called EfficientNets, which achieve much better accuracy and efficiency than previous ConvNets. In particular, our EfficientNet-B7 achieves stateof-the-art 84.4% top-1 / 97.1% top-5 accuracy on ImageNet, while being 8.4x smaller and 6.1x faster on inference than the best existing ConvNet. Our EfficientNets also transfer well and achieve state-of-the-art accuracy on CIFAR-100 (91.7%), Flowers (98.8%), and 3 other transfer learning datasets, with an order of magnitude fewer parameters. Source code is at https://github.com/tensorflow/tpu/tree/master/models/official/efficientnet.

\* ResNest,2020

Abstract. While image classification models have recently continued to advance, most downstream applications such as object detection and semantic segmentation still employ ResNet variants as the backbone network due to their simple and modular structure. We present a modular Split-Attention block that enables attention across feature-map groups.By stacking these Split-Attention blocks ResNet-style, we obtain a new ResNet variant which we call ResNeSt. Our network preserves the overall ResNet structure to be used in downstream tasks straightforwardly without introducing additional computational costs.ResNeSt models outperform other networks with similar model complexities. For example, ResNeSt-50 achieves 81.13% top-1 accuracy on ImageNet using a single crop-size of 224 × 224, outperforming previous best ResNet variant by more than 1% accuracy. This improvement also helps downstream tasks including object detection, instance segmentation and semantic segmentation. For example, by simply replace the

ResNet-50 backbone with ResNeSt-50, we improve the mAP of FasterRCNN on MS-COCO from 39.3% to 42.3% and the mIoU for DeeplabV3 on ADE20K from 42.1% to 45.1%.

##### Detect abstract

\* R-CNN

Object detection performance, as measured on the canonical PASCAL VOC dataset, has plateaued in the last few years. The best-performing methods are complex ensemble systems that typically combine multiple low-level image features with high-level context. In this paper, we propose a simple and scalable detection algorithm that improves mean average precision (mAP) by more than 30%

relative to the previous best result on VOC 2012—achieving a mAP of 53.3%. Our approach combines two key insights: (1) one can apply high-capacity convolutional neural networks (CNNs) to bottom-up region proposals in order to localize and segment objects and (2) when labeled training data is scarce, supervised pre-training for an auxiliary task, followed by domain-specific fine-tuning, yields a significant

performance boost. Since we combine region proposals with CNNs, we call our method R-CNN: Regions with CNN features. Source code for the complete system is available at [http://www.cs.berkeley.edu/rbg/rcnn.](http://www.cs.berkeley.edu/˜rbg/rcnn.)

\* Fast RCNN,2015

This paper proposes a Fast Region-based Convolutional Network method (Fast R-CNN) for object detection. Fast R-CNN builds on previous work to efficiently classify object proposals using deep convolutional networks. Compared to previous work, Fast R-CNN employs several innovations to improve training and testing speed while also increasing detection accuracy. Fast R-CNN trains the very deep VGG16 network 9× faster than R-CNN, is 213× faster at test-time, and achieves a higher mAP on PASCAL VOC 2012. Compared to SPPnet, Fast R-CNN trains VGG16 3×faster, tests 10× faster, and is more accurate. Fast R-CNN is implemented in Python and C++ (using Caffe) and is available under the open-source MIT License at <https://github.com/rbgirshick/fast-rcnn.>

\* SSD,2016

We present a method for detecting objects in images using a single deep neural network. Our approach, named SSD, discretizes the output space of bounding boxes into a set of default boxes over different aspect ratios and scales per feature map location. At prediction time, the network generates scores for the presence of each object category in each default box and produces adjustments to the box to better match the object shape. Additionally, the network combines predictions from multiple feature maps with different resolutions to naturally handle objects of various sizes. SSD is simple relative to methods that require object proposals because it completely eliminates proposal generation and subsequent pixel or feature resampling stages and encapsulates all computation in a single network. This makes SSD easy to train and straightforward to integrate into systems that require a detection component. Experimental results on the PASCAL VOC, COCO, and ILSVRC datasets confirm that SSD has competitive accuracy to methods that utilize an additional object proposal step and is much faster, while providing a unified framework for both training and inference. For 300 × 300 input, SSD achieves 74.3% mAP1 on VOC2007 test at 59 FPS on a Nvidia Titan X and for 512 × 512 input, SSD achieves 76.9% mAP, outperforming a comparable state-of-the-art Faster R-CNN model. Compared to other single stage methods, SSD has much better accuracy even with a smaller input image size. Code is available at: https://github.com/weiliu89/caffe/tree/ssd .

\* YOLO\_v1,2016

We present YOLO, a new approach to object detection.Prior work on object detection repurposes classifiers to perform detection. Instead, we frame object detection as a regression problem to spatially separated bounding boxes and associated class probabilities. A single neural network predicts bounding boxes and class probabilities directly from full images in one evaluation. Since the whole detection pipeline is a single network, it can be optimized end-to-end directly on detection performance.Our unified architecture is extremely fast. Our baseYOLO model processes images in real-time at 45 frames per second. A smaller version of the network, Fast YOLO, processes an astounding 155 frames per second while still achieving double the mAP of other real-time detectors. Compared to state-of-the-art detection systems, YOLO makes more localization errors but is less likely to predict false positives on background. Finally, YOLO learns very general representations of objects. It outperforms other detection methods, including DPM and R-CNN, when generalizing from natural images to other domains like artwork.

\* YOLO\_v2,2016

We introduce YOLO9000, a state-of-the-art, real-time object detection system that can detect over 9000 object categories. First we propose various improvements to the

YOLO detection method, both novel and drawn from prior work. The improved model, YOLOv2, is state-of-the-art on standard detection tasks like PASCAL VOC and COCO. Using a novel, multi-scale training method the same YOLOv2 model can run at varying sizes, offering an easy tradeoff between speed and accuracy. At 67 FPS, YOLOv2 gets 76.8 mAP on VOC 2007. At 40 FPS, YOLOv2 gets 78.6 mAP, outperforming state-of-the-art methods like Faster RCNN with ResNet and SSD while still running significantly faster. Finally we propose a method to jointly train on object detection and classification. Using this method we train YOLO9000 simultaneously on the COCO detection dataset and the ImageNet classification dataset. Our joint training allows YOLO9000 to predict detections for object classes

that don’t have labelled detection data. We validate our approach on the ImageNet detection task. YOLO9000 gets 19.7 mAP on the ImageNet detection validation set despite only having detection data for 44 of the 200 classes. On the 156 classes not in COCO, YOLO9000 gets 16.0 mAP. But YOLO can detect more than just 200 classes; it predicts detections for more than 9000 different object categories. And it still runs in real-time.

\* Faster RCNN,2016

State-of-the-art object detection networks depend on region proposal algorithms to hypothesize object locations. Advances like SPPnet and Fast R-CNN have reduced the running time of these detection networks, exposing region proposal computation as a bottleneck. In this work, we introduce a Region Proposal Network (RPN) that shares full-image convolutional features with the detection network, thus enabling nearly cost-free region proposals. An RPN is a fully convolutional network that simultaneously predicts object bounds and objectness scores at each position. The RPN is trained end-to-end to generate high-quality region proposals, which are used by Fast R-CNN for detection. We further merge RPN and Fast R-CNN into a single network by sharing their convolutional features—using the recently popular terminology of neural networks with “attention” mechanisms, the RPN component tells the unified network where to look. For the very deep VGG-16 model, our detection system has a frame rate of 5fps (including all steps) on a GPU, while achieving state-of-the-art object detection accuracy on PASCAL VOC 2007, 2012, and MS COCO datasets with only 300 proposals per image. In ILSVRC and COCO 2015 competitions, Faster R-CNN and RPN are the foundations of the 1st-place winning entries in several tracks. Code has been made publicly available.

\* R2CNN,2017

In this paper, we propose a novel method called Rotational Region CNN (R2CNN) for detecting arbitrary-oriented texts in natural scene images. The framework is based on Faster R-CNN architecture. First, we use the Region Proposal Network (RPN) to generate axis-aligned bounding boxes that enclose the texts with different orientations. Second, for each axis-aligned text box proposed by RPN, we extract its pooled features with different pooled sizes and the concatenated features are used to simultaneously predict the text/non-text score,axis-aligned box and inclined minimum area box. At last, we use an inclined non-maximum suppression to get the detection results. Our approach achieves competitive results on text detection benchmarks: ICDAR 2015 and ICDAR 2013.

\* MobileNet,2017

We present a class of efficient models called MobileNets for mobile and embedded vision applications. MobileNets are based on a streamlined architecture that uses depthwise separable convolutions to build light weight deep neural networks. We introduce two simple global hyperparameters that efficiently trade off between latency and accuracy. These hyper-parameters allow the model builder to choose the right sized model for their application based on the constraints of the problem. We present extensive experiments on resource and accuracy tradeoffs and show strong performance compared to other popular models on ImageNet classification. We then demonstrate the effectiveness of MobileNets across a wide range of applications and use cases including object detection, finegrain classification, face attributes and large scale geo-localization.

\* YOLO\_v3,2018

We present some updates to YOLO! We made a bunch of little design changes to make it better. We also trained this new network that’s pretty swell. It’s a little bigger than last time but more accurate. It’s still fast though, don’t worry. At 320 × 320 YOLOv3 runs in 22 ms at 28.2 mAP, as accurate as SSD but three times faster. When we look at the old .5 IOU mAP detection metric YOLOv3 is quite good. It achieves 57.9 AP50 in 51 ms on a Titan X, compared to 57.5 AP50 in 198 ms by RetinaNet, similar performance but 3.8× faster. As always, all the code is online at <https://pjreddie.com/yolo/.>

\* CornerNet,2018

We propose CornerNet, a new approach to object detection where we detect an object bounding box as a pair of keypoints, the top-left corner and the bottom-right corner, using a single convolution neural network. By detecting objects as paired keypoints, we eliminate the need for designing a set of anchor boxes commonly used in prior

single-stage detectors. In addition to our novel formulation, we introduce corner pooling, a new type of pooling layer that helps the network better localize corners. Experiments show that CornerNet achieves a 42.1% AP on MS COCO, outperforming all existing one-stage detectors.

\* CenterNet,2019

In object detection, keypoint-based approaches often suffer a large number of incorrect object bounding boxes, arguably due to the lack of an additional look into the cropped regions. This paper presents an efficient solution which explores the visual patterns within each cropped region with minimal costs. We build our framework upon a representative one-stage keypoint-based detector named CornerNet. Our approach, named CenterNet, detects each object as a triplet, rather than a pair, of keypoints, which improves both precision and recall. Accordingly, we design two customized modules named cascade corner pooling and center pooling, which play the roles of enriching information collected by both top-left and bottom-right corners and providing more recognizable information at the central regions, respectively. On the MS-COCO dataset, CenterNet achieves an AP of 47.0%, which outperforms all existing one-stage detectors by at least 4.9%. Meanwhile,with a faster inference speed, CenterNet demonstrates quite comparable performance to the top-ranked two-stage detectors. Code is available at https://github.com/Duankaiwen/CenterNet.

\* SCRDet,2019

Object detection has been a building block in computer vision. Though considerable progress has been made, there still exist challenges for objects with small size, arbitrary direction, and dense distribution. Apart from natural images, such issues are especially pronounced for aerial images of great importance. This paper presents a novel multicategory rotation detector for small, cluttered and rotated objects, namely SCRDet. Specifically, a sampling fusion network is devised which fuses multi-layer feature with effective anchor sampling, to improve the sensitivity to small objects. Meanwhile, the supervised pixel attention network and the channel attention network are jointly explored for small and cluttered object detection by suppressing the noise and highlighting the objects feature. For more accurate rotation estimation, the IoU constant factor is added to the smooth L1 loss to address the boundary problem for

the rotating bounding box. Extensive experiments on two remote sensing public datasets DOTA, NWPU VHR-10 as well as natural image datasets COCO, VOC2007 and scene text data ICDAR2015 show the state-of-the-art performance of our detector. The code and models will be available at <https://github.com/DetectionTeamUCAS.>

\* R3Det,2020

Rotation detection is a challenging task due to the difficulties of locating the multi-angle objects and separating them accurately and quickly from the background. Though considerable progress has been made, for practical settings,there still exist challenges for rotating objects with large aspect ratio, dense distribution and category extremely imbalance. In this paper, we propose an end-to-end refined single-stage rotation detector for fast and accurate positioning objects. Considering the shortcoming of feature misalignment in existing refined single-stage detector, we

design a feature refinement module to improve detection performance by getting more accurate features. The key idea of feature refinement module is to re-encode the position information of the current refined bounding box to the corresponding feature points through feature interpolation to realize feature reconstruction and alignment. Extensive experiments on two remote sensing public datasets DOTA, HRSC2016 as well as scene text data ICDAR2015 show the state-of-the-art accuracy and speed of our detector. Code is available at <https://github.com/> Thinklab-SJTU/R3Det\_Tensorflow.

\* YOLO\_v4,2020

There are a huge number of features which are said to improve Convolutional Neural Network (CNN) accuracy. Practical testing of combinations of such features on large

datasets, and theoretical justification of the result, is required. Some features operate on certain models exclusively and for certain problems exclusively, or only for small-scale datasets; while some features, such as batch-normalization and residual-connections, are applicable to the majority of models, tasks, and datasets. We assume that such universal features include Weighted-Residual-Connections (WRC),

Cross-Stage-Partial-connections (CSP), Cross mini-Batch Normalization (CmBN), Self-adversarial-training (SAT) and Mish-activation. We use new features: WRC, CSP, CmBN, SAT, Mish activation, Mosaic data augmentation, CmBN, DropBlock regularization, and CIoU loss, and combine some of them to achieve state-of-the-art results: 43.5% AP (65.7% AP50) for the MS COCO dataset at a realtime speed of ∼65 FPS on Tesla V100. Source code is at <https://github.com/AlexeyAB/darknet.>

\* EfficientDet,2020

Model efficiency has become increasingly important in computer vision. In this paper, we systematically study neural network architecture design choices for object detection and propose several key optimizations to improve efficiency.

First, we propose a weighted bi-directional feature pyramid network (BiFPN), which allows easy and fast multiscale feature fusion; Second, we propose a compound scaling method that uniformly scales the resolution, depth, and width for all backbone, feature network, and box/class prediction networks at the same time. Based on these optimizations and better backbones, we have developed a new family of object detectors, called EfficientDet, which consistently achieve much better efficiency than prior art across a wide spectrum of resource constraints. In particular, with singlemodel and single-scale, our EfficientDet-D7 achieves stateof-the-art 55.1 AP on COCO test-dev with 77M parameters and 410B FLOPs1, being 4x – 9x smaller and using 13x – 42x fewer FLOPs than previous detectors. Code is available at <https://github.com/google/automl/tree/master/efficientdet.>

\* YOLO\_x,2021

In this report, we present some experienced improvements to YOLO series, forming a new high-performance detector — YOLOX. We switch the YOLO detector to an

anchor-free manner and conduct other advanced detection techniques, i.e., a decoupled head and the leading label assignment strategy SimOTA to achieve state-of-the-art results across a large scale range of models: For YOLONano with only 0.91M parameters and 1.08G FLOPs, we get 25.3% AP on COCO, surpassing NanoDet by 1.8% AP; for YOLOv3, one of the most widely used detectors in industry, we boost it to 47.3% AP on COCO, outperforming the current best practice by 3.0% AP; for YOLOX-L with roughly the same amount of parameters as YOLOv4-CSP, YOLOv5-L, we achieve 50.0% AP on COCO at a speed of 68.9 FPS on Tesla V100, exceeding YOLOv5-L by 1.8% AP. Further, we won the 1st Place on Streaming Perception Challenge (Workshop on Autonomous Driving at CVPR 2021) using a single YOLOX-L model. We hope this report can provide useful experience for developers and researchers in practical scenes, and we also provide deploy versions with ONNX, TensorRT, NCNN, and Openvino supported. Source code is at <https://github.com/>Megvii-BaseDetection/YOLOX.