Deep Learning–Based Multimedia Analytics: A Review

WEI ZHANG and TING YAO, JD AI Research, China
SHIAI ZHU, Ant Financial Group, China
ABDULMOTALEB EL SADDIK, University of Ottawa, Canada

The multimedia community has witnessed the rise of deep learning–based techniques in analyzing multimedia content more effectively. In the past decade, the convergence of deep-learning and multimedia analytics has boosted the performance of several traditional tasks, such as classification, detection, and regression, and has also fundamentally changed the landscape of several relatively new areas, such as semantic segmentation, captioning, and content generation. This article aims to review the development path of major tasks in multimedia analytics and take a look into future directions. We start by summarizing the fundamental deep techniques related to multimedia analytics, especially in the visual domain, and then review representative high-level tasks powered by recent advances. Moreover, the performance review of popular benchmarks gives a pathway to technology advancement and helps identify both milestone works and future directions.

CCS Concepts: • **General and reference → Surveys and overviews**; • **Computing methodologies → Artificial intelligence**; **Neural networks**;

Additional Key Words and Phrases: Multimedia analytics, deep learning, neural networks

ACM Reference format:
Wei Zhang, Ting Yao, Shiai Zhu, and Abdulmotaleb El Saddik. 2019. Deep Learning–Based Multimedia Analytics: A Review. *ACM Trans. Multimedia Comput. Commun. Appl.* 15, 1s, Article 2 (January 2019), 26 pages.
https://doi.org/10.1145/3279952

1 INTRODUCTION

Due to the advancement of multimodal sensors, today's digital content is inherently multimedia, for example, text, images, audio, and video. The multimedia data of interest covers a wide spectrum, ranging from text, audio, images, click-through logs, Web videos, EEG signals, to surveillance videos. Visual content—that is, images and video, in particular—is becoming a new way of communicating among Internet users with the proliferation of sensor-rich mobile devices. Accelerated by a tremendous increase in Internet bandwidth and storage space, multimedia data has been generated, published, and spread explosively, becoming an indispensable part of today's big data.

Such large-scale multimedia data has generated challenges and opportunities for intelligent multimedia analysis, for example, management, retrieval, recognition, categorization, visualization, and generation. Meanwhile, with recent advances in deep-learning techniques, we are now able to boost the intelligence of multimedia analysis significantly and initiate new research directions

Authors' addresses: W. Zhang, JD AI Research, 8 Beichen West Road, Beijing, China; email: wzhang.cu@gmail.com; T. Yao, JD AI Research, 8 Beichen West Road, Beijing, China; email: tingyao.ustc@gmail.com; S. Zhu (corresponding author), Ant Financial Group, Hangzhou, China; email: zshiai@gmail.com; A. El Saddik, University of Ottawa, 800 King Edward, Ottawa, Canada; email: elsaddik@uottawa.ca.

1551-6857/2019/01-ART2 $15.00
https://doi.org/10.1145/3279952

to analyze multimedia content. For instance, convolutional neural networks have demonstrated high capability in image and video recognition, recurrent neural networks are widely exploited in modeling temporal dynamics in videos, and generative adversarial networks are capable of generating realistic images on demand. Therefore, deep learning for intelligent multimedia analysis is becoming an emerging research area in the field of multimedia and computer vision.

Many multimedia tasks can be viewed as mapping multimedia content to a set of outputs with different capacities, ranging from several bits to kilobytes. Multimedia analytics can therefore be categorized as classification, detection, captioning, and segmentation, according to different capacities of their outputs. After years of research, the output has evolved from single prediction (e.g., class label) to structured output (e.g., bounding box, sentence and image). Generally speaking, a larger output usually corresponds to a more challenging task.

This article reviews recent advances in deep learning–based multimedia analytics. The goal is to review state-of-the-art deep-learning components and network architectures, to identify typical scenarios and challenges emerging in multimedia analysis, and to discuss real-world datasets and benchmarks for future directions. In Section 2, we start with the core building blocks shared by different architectures. In Section 3, we review several representative high-level tasks, including classification, detection, captioning, and semantic segmentation. Standard benchmarks and the corresponding state-of-the-art are summarized in Section 4 to show a clear roadmap in terms of performance. Conclusions and future directions are discussed in Section 5.

2 PRELIMINARY

In this section, we first review the basic building blocks of deep learning layers in Section 2.1, and then discuss several widely adopted network architectures in Section 2.2.

2.1 Building Blocks

Most popular deep-learning frameworks are highly modularized, such that deep networks can be easily constructed by a collection of interacting layers.

The convolution layer [56] applies the convolution operation over an input signal, which is especially critical for multimedia visual data. Transposed convolution[1] goes in the opposite direction [125] and is widely adopted for upsampling in deblurring [115], image matting [133], super resolution [58], image generation [100], and restoration [79].

The fully connected layer defines a linear transformation between nodes, where neurons in one layer are connected to all neurons in another layer. A typical functionality is high-level reasoning [56] after several convolution layers.

Activation [18, 33, 34, 77, 82] and **pooling layers** [56, 111] introduce nonlinearity into the networks, which have demonstrated their superior performance in multimedia analytics. In principle, activation defines the response mechanism for neuron outputs and pooling combines multiple outputs at one layer into a single output in the next layer, which introduces a form of non-linear downsampling.

The normalization layer [45, 121] is critical in stabilizing the training process and accelerating convergence speed. For example, Batch Normalization [45] reduces the internal covariate shift in the neural networks, leading to faster convergence.

The loss layer defines various loss functions for diverse purposes, ranging from *L1*, *MSE*, *Cross Entropy*, *Negative log likelihood* losses to *KL-divergence* and *Triplet Margin* losses.

Optimizing deep networks is generally difficult. On one hand, different initializations [29, 38] have a major impact on network convergence. On the other hand, the optimization algorithm

[1] In some literature, it is also referred to as fractionally strided convolution or deconvolution.

is also important for convergence. The fundamental optimization technique is based on backpropagation [108] via SGD [95, 110], Adam [52], LBFGS, Rprop, or RMSprop [31].

2.2 Network Architectures

2.2.1 Convolutional Neural Networks (CNN). The convolutional neural network was first introduced decades ago [59] for recognizing ZIP codes, which was rather primitive at that time. Recent advancements in computational hardware (GPU[2]) and massive training data [109] bring CNN architectures [56] to a wide range of fields, such as object/scene/action detection, recognition, and regression.

2.2.2 Recurrent Neural Networks (RNN). In addition to the feed-forward architecture, another important branch is based on the recurrent structure RNNs to model temporal dynamics. Note that this is essentially helpful for sequential signals, such as video, text, and audio. However, the RNN suffers from the vanishing and exploding gradient problem. To address this problem, Long Short-Term Memory (LSTM) captures the long-term dependencies with the cell state, while GRU further combines the forget and input gates, and mergers the cell state and hidden state.

2.2.3 Generative Adversarial Networks (GAN). Generative Adversarial Networks (GAN) [30] have gained great attention in recent years, which are usually composed of a generator and a discriminator. A GAN defines a high-level objective, real or fake, rather than a specified loss function. Following this principle, many variants, for example, wGAN [3] and DCGAN [100], have improved the original framework for generating more realistic and robust images. The Laplacian Pyramid of Adversarial Networks [21] extends the GAN for progressively generating images with higher resolutions. A more recent work by NVIDIA [49] further generates celebrity photos in impressive quality. A conditional GAN [80] takes extra input for generating images based on a constraint. Pix2pix [46] translates an image to another representation with a conditional GAN. CycleGAN [154], DiscoGAN [51] and DualGAN [142] share the same idea for image translation between different domains, where unpaired data is adopted in a self-supervised way. Moreover, a conditional GAN is also applied for generating images conditioning on text descriptions [83, 102, 148].

3 HIGH-LEVEL TASKS

This section reviews high-level tasks in multimedia analytics built on deep techniques. This survey covers a wide range of tasks in multimedia analytics, especially in the visual domain.

Figure 1 shows the overall landscape of multimedia analytics, which maps multimedia data (blue) into diverse outputs (yellow). Below, we briefly summarize the high-level tasks according to their output capacities (quantity of information).

- Image/Video → Label. This category maps multimedia content to a set of predefined labels, which are mostly represented in several bits. Typical tasks include object/scene/action classification.
- Image/Video → Region. Region-level multimedia understanding corresponds to object/action detection, where one or several bounding boxes (10~100b) are required to locate the target object/action.
- Image/Video → Sentence. Describing multimedia content with natural sentences or a paragraph (0.1~1KB) has been a fast-developing area in recent years. Typical tasks include image/video captioning.

[2] Graphics Processing Unit.

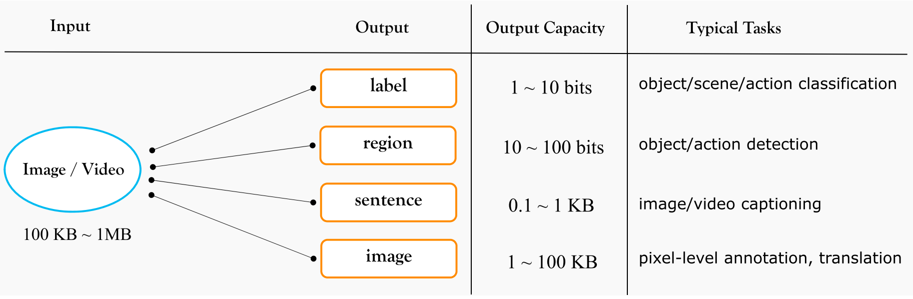

Fig. 1. A landscape of deep learning–based multimedia analytics.

- Image/Video → Image. Mapping visual content to an image is more challenging since it requires pixel-level annotation/translation, where the structured output has much more capacity. Typical tasks include semantic segmentation and image translation.

Generally speaking, larger output capacity leads to more difficulty. Image classification aims to map an input image/video to one or multiple labels, while image captioning parses an input image to a sentence. The difficulty of each task increases as the output capacity grows. Moreover, reverse mapping (from left to right) is also valid for all of the above cases owing to recent advances in GANs. In the following, we will review the four most representative tasks in multimedia analytics: classification, detection, captioning, and semantic segmentation.

3.1 Classification

Classification is the most fundamental task in multimedia analytics. The work of LeCun in [59], known as LeNet-5, serves as the basis for modern frameworks in the family of deep-learning techniques. The structure of a CNN typically consists of stacked convolutional layers that are optionally followed by normalization or pooling layers. LeNet-5 and its variants achieved state-of-the-art performance on several simple visual classification tasks, such as character recognition. However, owing to the limited number of training instances and computational resources, LetNet-5 did not perform well on more complex visual tasks. Recently, several new network structures have been proposed to address the problem. In the following, we will briefly discuss the milestone CNN architectures.

The first breakthrough structure, AlexNet, was proposed by Krizhevsky [56], which significantly boosts the performance of large-scale image classification in the ImageNet Large Scale Visual Recognition Competition (ILSVRC) of 2012. AlexNet contains eight learned layers: five convolutional layers followed by three fully connected layers. Compared to LeNet-5, AlexNet improves CNN learning in several ways: (1) data augmentation, (2) dropout, (3) ReLU nonlinearity, (4) local response normalization and (5) overlapping pooling. Two novel components widely used in the following works are ReLU and dropout. ReLU is defined as $f(x) = \max(0, x)$, which turns the negative input into zero. It has been demonstrated that networks using ReLU activation can be trained several times faster than the non-saturating functions. During the learning of AlexNet, the dropout technique is introduced to randomly set output of hidden neurons to zero with probability 0.5. In this way, the dropped neurons will not contribute to the forward and backward propagation. With dropout, substantial overfitting can be alleviated.

The success of AlexNet attributes to its powerful representation ability through multiple layers of nonlinear transformation. Inspired by AlexNet, Network-in-Network (NIN) is proposed in [68] to enhance the learned representation in neural networks. There are two major contributions in this CNN architecture: multiple linear perceptron convolution (MLP) and global average pooling. Specifically, the standard linear convolution filter is replaced by a mini-network, including two additional fully connected layers with nonlinear activation function. Each fully connected layer is also equivalent to a convolution layer with a 1 × 1 convolution kernel. In this way, the interactions across channels can be learned. In addition, deeper structure gives more capability approximating more abstract representations of the latent concepts. Another contribution in NIN is the utility of global average pooling to replace the fully connected layers for classification. The output of the pooling layer is used as confidence of categories and directly fed into the softmax. Compared to a fully connected layer, the new design contains less parameters to be learned and is less prone to overfitting.

As a deeper network is able to improve discriminative ability, VGGNet [114] further pushes the depth of the CNN to 16 and 19 layers, which correspond to VGG16 and VGG19, respectively. The simple increase of depth is feasible owing to the utility of the 3×3 convolution kernel in all of the layers. In contrast to AlexNet, where the first convolution layer enjoys relatively large receptive fields using an 11 × 11 kernel, VGGNet adopts a stack of multiple 3 × 3 convolution layers to archive large receptive fields. Meanwhile, the decision function becomes more discriminative by increasing the non-linearity functions. Another advantage is that the number of parameters can be controlled at a reasonable level. VGGNet is very appealing as its uniform architecture and is frequently used for extracting features from images.

GoogleNet [118] is another powerful CNN structure consisting of up to 22 layers. The basic component in GoogleNet is a novel CNN structure referred to as the Inception module, which consists of several parallel convolution layers with different kernels sizes. The outputs of sub-branches are concatenated into a single output. Compared with previous structures, filter-level sparsity is intuitively introduced in the Inception module. Thus, the use of computational resources can be significantly reduced, and the enlarged network is less prone to overfitting. To further reduce the number of parameters, dimension reduction is performed before the costly 3×3 and 5×5 layers by adding a 1 × 1 convolution layer. Also, a max pooling layer is included to summarize the information from previous layer. By stacking several inception modules, both the depth and width of a CNN are increased while maintaining an affordable computational cost. There are several extensions upon the basic inception module. The 5 × 5 kernel is replaced by two 3 × 3 kernels in Inception-v2 [119]. Another improvement is the proposal of a batch normalization (BN) layer, which is widely used in the following networks. In Inception-v3 [119], the reduced parameters and increased depth are achieved by using stacked 1 × 3 and 3 × 1 kernels rather than a single 3×3 kernel.

A deeper network causes two major problems: vanishing/exploding gradients and accuracy degradation. The first problem has been largely addressed by using different kinds of normalization strategies or adding auxiliary loss in middle layers, as in GoogleNet. However, when the network starts converging, the accuracy gets saturated and degrades by adding more layers to a structure with suitable depth. ResNet [37] is proposed to address the degradation problem by explicitly learning residual functions with reference to the inputs rather than directly fitting the desired mapping from input to output. The core idea is the introduction of an identity shortcut connection. By simply stacking identity mappings, a deeper model should not cause a higher training error than the shallower counterparts. The original underlying mapping $H(x)$ is approximated as learning a residual function $F(x) = H(x) - x$, which would be easier to optimize. In addition, residual block does not introduce additional cost. As a result, the depth of a CNN can be up to 152 layers. ResNet is further extended to ResNext [130], where split-transform-merge is adopted. It is intuitively an Inception module using much more sub-branches to increase the dimension of cardinality. In this way, similar performance can be achieved using a shallower network.

Similarly, the residual connect is also introduced in the family of GoogleNet, known as Inception-v4 architecture [117].

The great success of the shortcut connection has attracted much attention. The most representative one is the DenseNet [43], which consists of several stacked dense blocks. Within the dense block, each layer is connected to every subsequent layer in the feed-forward fashion. In contrast to ResNet using summation, the feature maps of all preceding layers are concatenated with that of current layer. In contrast to previous proposed architectures, DenseNet adopts very narrow layers, for example, 12 feature maps. However, the concatenated input still contains many feature maps. To address this problem, a bottleneck layer using a 1×1 convolution is introduced to reduce the dimension before the 3×3 convolution. DenseNet enjoys several advantages, such as alleviating the problem of vanishing gradient, strengthening information propagation, encouraging feature reuse, and computational efficiency. In [143], backward update is further incorporated into DenseNet, where several previous layers are connected to update the next layer, and the newly updated layer is concatenated to the previous layer. In this way, the effectiveness of forward and backward information flow can be maximized. Recently, a novel architectural unit named Squeeze-and-Excitation (SE) block is proposed in [48], where interdependencies between channels are modeled by re-weighting channel-wise feature responses. It has been demonstrated that previous proposed network architectures can be further improved by integrating the SE unit.

Works on compact CNN architectures has also drawn great interests. SqueezeNet, proposed in [44], achieves a similar accuracy as AlexNet, with 50x fewer parameters. The basic component is the Fire module, comprised of a squeeze layer and an expanding layer. The squeeze layer use a 1×1 convolution to reduce the number of channels. The squeezed outputs further feed into the expanding layer, which includes a mix of parallel 1×1 and 3×3 filters. We can see that Fire block is fundamentally derived from the Inception module in GoogleNet. Another technique in model size reduction is the utility of depth-wise separable convolution. Specifically, convolution is first performed for each input feature map, followed by a standard 1×1 convolution to capture the correlations across different channels. This structure has been successfully adopted in MobileNet [40] and XCeption [17]. Instead of applying a filter on each channel, a more moderate way, named *group-wise convolution*, is adopted in ShuffleNet [149], where input feature maps are separated into several groups. The standard convolution is performed within each group. Cross-channel information is integrated by shuffling all of the channels between two group-wise convolution layers. A more efficient network, ShuffleNet-V2, using group convolution is proposed in [76] by further considering the memory access cost. In addition, several practical guidelines for efficient network design are introduced. Group convolution is also employed in CondenseNet [42], which is a more efficient DenseNet architecture. In contrast to ShuffleNet, where the groups are predefined, CondenseNet proposes to learn group convolution through a multiple training stage. In general, $1 * 1$ convolution and channel-wise separable convolution have been widely used in the design of computation-efficient CNN architectures.

Video classification has also attracted intensive research interest in recent years. For example, Simonyan and Zisserman [113] design two-stream ConvNets, which contains spatial and temporal nets to capture the discriminative appearance feature and motion feature, respectively. Qiu et al. [97] design a novel end-to-end deep quantization architecture by incorporating the Fisher Vector encoding strategy into deep generative models. Later, in [120] and [98], 3D ConvNets are employed to learn spatio-temporal video descriptors to capture appearance features and motion features in a unified network. In general, the inputs of these architectures are often frames or short video clips, making it difficult for the networks to capture long-term temporal information. To overcome this drawback, the LSTM networks are exploited in [128, 129]. Li et al. [62, 63] propose a multi-granular deep architecture and employ LSTM to incorporate long-term temporal dynamics based on multiple granularity features. More recently, several techniques have been proposed to boost action recognition by incorporating an attention mechanism [61], human detector [60], or temporal coherence [84]. By combining different components, the authors of [47, 96, 137] present multi-model systems with higher classification accuracy for video classification.

3.2 Detection

In this section, we will review the most recent deep-learning methods on general object detection, where both location and category information of the appeared objects should be provided. The existing works can be roughly categorized into two groups: two-stage approach and one-stage approach. The former uses a CNN classifier on several regions of interest (RoIs), and suspected candidates are further merged as final results. In contrast, end-to-end approaches predict the location and category in a single CNN network.

R-CNN [28] is one of the preliminary works on CNN-based object detection. In R-CNN, RoI proposals are generated by selective search, which groups the adjacent pixels into candidate object regions based on the texture, color, or intensity information. The category of each generated RoI is determined by warping it to a standard CNN classifier. In R-CNN, the CNN is used only for feature extraction. A linear support vector machine (SVM) is adopted for training the classifier. Finally, the bounding boxes are refined using a regression model. R-CNN provides the first practical solution for object detection using a CNN. However, the training is expensive both in the use of computational resource and storage. Another major drawback is the slow inference, as it requires a forward pass of the CNN for every RoI proposal.

The region proposals are actually invariably overlapped, causing computation waste in the repeated forward pass of the CNN. To address this problem, SPPNet [36] proposes spatial pyramid pooling (SPP) layer on the last convolution layer of the CNN. The representation of each region proposal is obtained by applying the SPP to pool the corresponding window of the feature maps into a fixed-length feature vector. The SPP layer contains several predefined grid separations (e.g., 1×1, 2×2, and 3×3) on any arbitrary windows. Feature pooling is applied within each grid. In this way, the feature maps are extracted from the entire image only once. SPPNet accelerated the R-CNN by 24 to 100x of the inference time. Following the multi-stage pipeline of R-CNN, the convolution layers cannot be updated during fine-tuning. To further improve speed and accuracy, Fast R-CNN [27] adopts RoI pooling layer and multi-task learning techniques. RoI pooling using only one pyramid layer is simply a special case of the SPP layer. In contrast to the SPPNet and R-CNN, Fast R-CNN uses two learning tasks on top of the CNN that jointly optimizes a softmax classifier and bounding box regression. Both detection quality and resource consuming are benefits of single network training and inference.

The major bottleneck in the Fast R-CNN is the region proposal, which is not only time consuming but also generates around 2000 RoIs. To address this problem, the Faster R-CNN [104] introduces a Region Proposal Network (RPN), which can directly regress the RoI bounding boxes. Specifically, a sliding window is applied on the feature map. At each location, k region proposals are predicted simultaneously by k binary objectiveness classifiers and k coordinate regressors. The k proposals and loss function are parameterized relative to k reference anchors with difference scale and aspect ratio. The top 300 RoIs receiving highest confidence scores are further used as input to the following Fast R-CNN object detection network. The advantage of the Faster R-CNN is the shared convolution layers for the RPN and detection network. As such, region proposals are nearly computationally cost free. To unify the RPN with the detection network, the two tasks are trained alternatively by fixing the shared layers and fine-tuning the layers unique to each network in turn. The two-stage design is further leveraged in the Mask R-CNN [35], where more accurate spatial pooling is achieved by using RoIAlign, rather than RoI pooling used in the Faster R-CNN.

In addition, more accurate localization of arbitrary shapes of objects is offered by introducing a new branch for predicting an object mask.

In the Faster R-CNN, the inference can be implemented in an end-to-end fashion. Nevertheless, the RPN and detection networks should be trained separately, even though they share full image convolution features. In contrast, one-stage object detection approaches target for end-to-end model training and testing. In YOLO [101], object detection is formulated as a single regression problem that directly predicts the bounding boxes and categories from full images. Specifically, the input image is divided into $S \times S$ grids. Each grid cell predicts B bounding boxes and confidence scores, indicating the probability of an object appealing in the predicted bounding box. In addition, each cell also predicts C conditional probabilities for C categories. Thus, the dimension of network output is $S \times S \times (B \times 5 + C)$. In contrast to the Faster R-CNN, both the bounding boxes and categories are predicted using the features from the entire image. This would result in a significant number of localization errors, as the spatial information is neglected. In YOLO-v2 [102], anchor boxes proposed in the Faster R-CNN are employed. Each anchor box predicts the bounding boxes and categories using features in the corresponding sub-region. As such, the number of final predicted outputs is significantly increased to $S \times S \times B \times (5 + C)$. YOLO-v2 exhibits better recall and localization ability while causing a small decrease in accuracy.

SSD [73] is another efficient one-stage object detection architecture, which conveys the concept of end-to-end regression in YOLO and anchor mechanism in the Faster R-CNN. For each anchor box, both the bounding box offsets and confidences of all categories are predicted. The prediction is performed on different resolutions of feature maps extracted from the backbone CNN network. As different feature maps correspond to different minimum receptive fields on the input image, the objects of various sizes and shapes are naturally handled by carefully designing the scale and ratio of the anchor boxes. In addition, small convolution kernels are applied on the feature maps. The computation cost is saved compared to the fully connected layers used in YOLO. For an $M \times N$ feature map, SSD yields $(C + 4) \times B \times M \times N$ outputs corresponding to C categories and B predefined anchors. While the generated boxes are larger than those produced by YOLO and Faster RCNN, most boxes can be filtered out by a confidence threshold. The remaining ones are refined via non-maximum suppression (NMS).

One-stage approaches using dense sampling of RoI regions tend to be more efficient; nevertheless, they have trailed the accuracy of two-stage detectors. The performance decrease contributes to the extreme foreground-background class imbalance. In You Only Look Once (YOLO) and Single Shot MultiBox Detector (SSD), hard negative sampling strategies are applied to address this problem. In contrast, focal loss [70], designed for one-stage detectors, alleviates the imbalance problem by down-weighting the well-classified examples and, in turn, enhancing the importance of misclassified examples. Some other works try to combine the advantages from different detection networks. For example, the feature pyramid network (FPN) [69] adopts pyramid networks for two-stage object detection. The prediction is performed on multiple feature maps rather than the last layer, as in the Faster R-CNN. In addition, the feature maps at the top layer, which is spatially coarser and semantically stronger, is upsampled and merged into its nearest bottom layer. A similar idea is used in the Reverse connection with Objectness prior Networks (RON) [54], where the RPN is replaced by a learned objectness prior network. In [112], the Deeply Supervised Object Detector (DSOD) is proposed by introducing the Dense Block in DenseNet to the Single Shot MultiBox Detector (SSD) architecture. Each feature map in the front-end structure is downsampled and concatenated into the low-resolution feature maps. In this way, the DSON is able to achieve a similar performance by learning from scratch with the model fine-tuned from a pretrained backbone network. This is essential in real-world applications, where the backbone networks are expensive to be learned owing to limited training instances and computational resources. All milestone architectures rely on detecting objects individually, ignoring the appearance relationships between instances. In [41], a relation network is proposed to process the target objects simultaneously by modeling their relations following the attention mechanism. The lightweight relation network is further used to replace the NMS, resulting in an end-to-end architecture. Recently, we have observed several other improvements and modifications of the above state-of-the-art architectures, such as the adaptive IOU thresholds in [152] and the scale invariance approach in [9].

3.3 Captioning

The research on image/video captioning has proceeded along three different dimensions: template-based methods [53, 57, 81, 106, 134], search-based approaches [22, 25], and sequence learning models [23, 78, 85, 86, 88, 123, 124, 126, 132, 136, 141, 144].

3.3.1 Template-Based Captioning. The template-based paradigm for captioning, in which each sentence fragment (e.g., subject, verb, and object) is first aligned with detected words from image content and then the sentence is generated with predefined language templates, has a long history. One of the earliest successes for image captioning is the work of Yang et al., who use Hidden Markov Model (HMM) to select the best objects, scenes, verbs, and prepositions with the highest log-likelihood ratio for template-based sentence generation in [134]. Similarly, in [57], Kulkarni et al. employ the Conditional Random Field (CRF) model to predict labeling based on the detected objects, attributes, and prepositions and then generate a sentence with a template by filling in slots with the most likely labeling. For video captioning, [53] builds a concept hierarchy of actions for natural-language description of human activities. Later, in [106], Rohrbach et al. teach a CRF to model the relationships between different components of the input video and generate descriptions for video based on the predefined template. While the template-based approach is the leading captioning paradigm in the earlier stage, most of them highly rely on the templates of sentences and invariably generate sentences with the same syntactical structure.

3.3.2 Search-Based Captioning. To achieve human-level descriptions, several search-based works attempt to “generate” sentences for an image/video by directly copying human-generated sentences from other visually similar images/videos. The significant drawbacks of these models have been that they cannot generate novel descriptions and the need to collect human-generated sentences also makes the sentence pool hard to be scaled up. For instance, in [25], an intermediate meaning space based on the triplet of object, action, and scene is proposed to measure the similarity between image and sentence, where the top sentences are regarded as the generated sentences for the target image. Recently, a simple k-nearest neighbor retrieval model was used in [22] and the best or consensus caption was selected from the returned candidate captions, which even performed as well as several state-of-the-art language-based models. Most recently, the search-based paradigm was also widely leveraged in image/video commenting [11, 64] to produce diverse comments depending on both visual content and emotional reaction.

3.3.3 Sequence Learning Captioning. The dominant paradigm in modern image/video captioning is the sequence learning method, which uses the CNN plus RNN architectures to generate novel sentences with more flexible syntactical structures. Here, the RNN architecture is employed to model the probability of generating a word given previous words and images across the word sequence. In particular, Vinyals et al. propose an end-to-end neural network architecture by using LSTM to generate a sentence for an image in [126], which is further incorporated with soft/hard attention mechanism in [132] to automatically focus on salient objects when generating corresponding words. Moreover, in [127, 141], semantic attributes are shown to clearly boost image captioning when injected into the existing state-of-the-art CNN plus RNN model; such attributes can be further leveraged as semantic attention [144] to enhance image captioning. In another work by Yao et al. [139], attribute detectors are used in image captioning to describe novel objects. Most recently, visual relationships have been integrated into image encoders to produce relation-aware, region-level features in image captioning [140]. For video captioning, Venugopalan et al. present an LSTM-based model to generate video descriptions with the mean pooling representation over all frames in [124]. The framework is then extended by inputting both frames and optical flow images into an encoder-decoder LSTM in [123]. Compared to mean pooling, Yao et al. propose using the temporal attention mechanism to exploit temporal structure for video captioning [136]. Furthermore, inspired by the idea of learning visual-semantic embedding space in search [89, 138], Pan et al. also consider the relevance between sentence semantics and video content as a regularizer in LSTM-based architecture [85]. Taking the inspiration from the use of semantic attributes in image captioning, an LSTM with a transferred semantic attributes model is designed in [88] to incorporate the transferred semantic attributes learned from both images and videos into the CNN plus RNN framework for video captioning. Instead of describing video with a single sentence, a hierarchical RNN in [147] is devised to further capture the inter-sentence dependency, targeting for describing a long video with a paragraph consisting of multiple sentences. In contrast to the video paragraph captioning with non-overlapping and annotated temporal intervals, a more challenging task—named *dense video captioning*—was recently studied in [55, 65, 137], which explore both detecting and describing multiple events in a video. Most recently, as a brave extension of video captioning (video to text), a novel temporal GANs-based architecture was developed in [87] to enable text-to-video synthesis of real-world cooking videos from human-written descriptions.

3.3.4 A Typical Architecture for Image/Video Captioning. In this section, we introduce the typical solution for image/video captioning, a CNN plus RNN scheme, which is mainly inspired by sequence learning models in machine translation [5, 116]. This CNN plus RNN scheme first encodes visual content (image/video) into a fixed dimensional vector via the CNN (2D/3D CNN) and then decodes it to the output target sentence through the RNN.

Specifically, suppose that we have an image I or video V with N_v sample frames/clips (uniform sampling) to be described by a textual sentence $\mathcal{S}$, where $\mathcal{S} = \{w_1, w_2, \ldots, w_{N_s}\}$ consisting of N_s words. Let $\mathbf{V}_i \in \mathbb{R}^{D_I}$, $\mathbf{V}_v \in \mathbb{R}^{D_v}$, and $\mathbf{w}_t \in \mathbb{R}^{D_w}$ denote the D_I-dimensional image representations of the image I, the D_v-dimensional video representations of the video V, and the D_w-dimensional textual features of the tth word in sentence $\mathcal{S}$, respectively. As a sentence consists of a sequence of words, a sentence can be represented by a $D_w \times N_s$ matrix $\mathbf{W} \equiv [\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_{N_s}]$, with each word in the sentence as its column vector. Hence, the image/video sentence generation problem that we exploit here can be generally formulated by minimizing the following energy loss function as

$$E(\mathbf{V}, \mathcal{S}) = -\log \Pr(\mathcal{S}|\mathbf{V}), \tag{1}$$

which is the negative log probability of the correct textual sentence given the image/video content. Note that we use $\mathbf{V} \in \{\mathbf{V}_i, \mathbf{V}_v\}$ for simplicity, that is, $\mathbf{V}$ denotes either image representations $\mathbf{V}_i$ or video representations $\mathbf{V}_v$ in the image/video captioning task, respectively. Since the CNN plus RNN scheme produces one word in the sentence at decoding stage, it is natural to apply a chain rule to model the joint probability over the sequential words. Thus, the log probability of the sentence is given by the sum of the log probabilities over each word, which can be expressed as

$$\log \Pr(\mathcal{S}|\mathbf{V}) = \sum_{t=1}^{N_s} \log \Pr(\mathbf{w}_t|\mathbf{V}, \mathbf{w}_0, \ldots, \mathbf{w}_{t-1}). \tag{2}$$

In the CNN plus RNN scheme, the above parametric distribution $\Pr(\mathbf{w}_t|\mathbf{V}, \mathbf{w}_0, \ldots, \mathbf{w}_{t-1})$ in Equation (2) is commonly modeled with the LSTM network, which is a widely used type of RNN and can capture long-term information in the sequential data by mapping sequences to sequences.

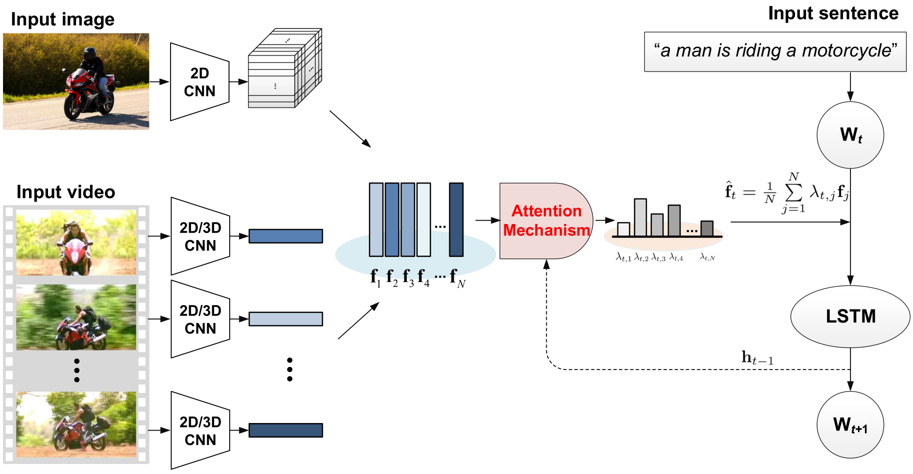
Input image
2D CNN
Input video
2D/3D CNN
2D/3D CNN
2D/3D CNN
$\mathbf{f}_1\ \mathbf{f}_2\ \mathbf{f}_3\ \mathbf{f}_4 \cdots \mathbf{f}_N$
Attention Mechanism
$\lambda_{t,1}\ \lambda_{t,2}\ \lambda_{t,3}\ \lambda_{t,4}\ \lambda_{t,N}$
$\hat{\mathbf{f}}_t = \frac{1}{N}\sum_{j=1}^{N}\lambda_{t,j}\mathbf{f}_j$
Input sentence
"a man is riding a motorcycle"
$\mathbf{W}_t$
LSTM
$\mathbf{h}_{t-1}$
$\mathbf{W}_{t+1}$

Fig. 2. The common architecture of the CNN plus RNN scheme with attention mechanism for image/video captioning (better viewed in color). A 2D CNN is employed to generate convolutional image representations for the input image. For the input video, the corresponding sequence of video representations is produced by using a 2D/3D CNN to extract visual features for sampled frames/clips. Both of the convolutional image representations and the sequence of frame/clip representations can be treated as a context set containing a number of fixed-dimensional context vectors. Each context vector corresponds to a certain spatial or temporal location in the input image/video. An attention mechanism is devised to measure a normalized attention distribution over all context vectors and thus achieve the attended image/video feature by aggregating all of the context vectors weighted with attention. The attended image/video feature is also injected into LSTM for boosting image/video captioning.

Taking inspiration from the attention mechanism in machine translation [5], a visual/temporal attention mechanism tailored to image/video captioning [132, 136] was recently incorporated into the CNN plus RNN scheme. To better summarize the architecture of the CNN plus RNN scheme with the attention mechanism for image/video captioning, we depict its common architecture in Figure 2. Unlike the CNN plus RNN scheme without an attention mechanism that takes the outputs of a fully connected layer as image representation, the convolutional image features, that is, the output feature map of the convolutional layer that contains more spatial information, is used here to represent input image. Suppose that the dimensions of the convolutional image features are $K \times K \times D_i$, where $K \times K$ is the number of regions in the feature map and D_i represents the dimension of the feature vector for each region. The local descriptor of each image region is denoted as $\mathbf{f}_j^i \in \mathbb{R}^{D_i}$, $j \in [1, K^2]$, where j is the index of each region. Therefore, the whole image feature map consisting of K^2 D_i-dimensional local descriptors for image I is represented as $\mathbf{F}_I = [\mathbf{f}_1^i, \mathbf{f}_2^i, \ldots, \mathbf{f}_{K^2}^i] \in \mathbb{R}^{D_i \times K^2}$. Each local descriptor slices the feature map into different overlapping regions in the raw image. We refer to these local descriptors as the feature cube of input image in Figure 2. For input video V, a 2D/3D CNN is adopted to extract visual features for each sampled frame/clip, resulting in a temporal sequence of visual features $\mathbf{F}_V = [\mathbf{f}_1^v, \mathbf{f}_2^v, \ldots, \mathbf{f}_{N_v}^v] \in \mathbb{R}^{D_v \times N_v}$. $\mathbf{f}_j^v \in \mathbb{R}^{D_v}$ denotes the representation of the jth sampled frame/clip. Therefore, both of the convolutional image representations $\mathbf{F}_I$ of the input image and the sequence of frame/clip representations $\mathbf{F}_V$ of the input video can be treated as one kind of context set containing a number of fixed-dimensional context vectors, which is denoted as $\mathbf{F} = [\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_N] \in \mathbb{R}^{D \times N} \in \{\mathbf{F}_I, \mathbf{F}_V\}$ in general. Each context vector $\mathbf{f}_j$ corresponds to a certain spatial or temporal location in the input image/video.

In many cases, the output word at decoding stage only relates to some regions of the input image or several frames/clips of the input video. As a result, using one encoder to compress the whole image or video into a global feature vector may lead to sub-optimal results owing to the noises introduced from regions or frames/clips that are irrelevant to the output word. To dynamically pinpoint the regions or frames/clips that are highly relevant to the output word at each timestep and further incorporate the contributions of different regions or frames/clips into producing image/video representation that would be fed into the LSTM, an attention mechanism is employed over the context set for boosting image/video captioning. Most specifically, at each timestep t, the attention mechanism first generates a normalized attention distribution over all of the context vectors $\mathbf{F} = [\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_N]$ depending on the previous output of LSTM $\mathbf{h}_{t-1}$:

$$a_{t,j} = \mathbf{W}_a[\tanh(\mathbf{W}_f \mathbf{f}_j + \mathbf{W}_h \mathbf{h}_{t-1})], \;\; \lambda_t = softmax(\mathbf{a}_t), \tag{3}$$

where $a_{t,j}$ is the jth element of $\mathbf{a}_t$, and $\mathbf{W}_a \in \mathbb{R}^{1\times D_a}$, $\mathbf{W}_f \in \mathbb{R}^{D_a\times D}$, and $\mathbf{W}_h \in \mathbb{R}^{D_a\times D_h}$ are transformation matrices. $\lambda_t \in \mathbb{R}^K$ denotes the normalized attention distribution and its jth element $\lambda_{t,j}$ is the attention probability of $\mathbf{f}_j$. Based on the attention distribution, we calculate the attended image/video feature $\hat{\mathbf{f}}_t = \frac{1}{N}\sum_{j=1}^{N} \lambda_{t,j}\mathbf{f}_j$ by aggregating all of the context vectors weighted with attention. We further concatenate the attended image/video feature $\hat{\mathbf{f}}_t$ with the input word w_t and feed them into the LSTM, whose updating procedure is given as:

$$\mathbf{h}_t = f([\hat{\mathbf{f}}_t, \mathbf{W}_s \mathbf{w}_t]), \tag{4}$$

where f is the updating function within the LSTM and $\mathbf{W}_s \in \mathbb{R}^{D_s\times D_w}$ is the transformation matric for input word w_t. The output of the LSTM $\mathbf{h}_t$ is leveraged to predict the next word w_{t+1} through a softmax layer.

In the training stage, the LSTM in the decoder is typically optimized with cross-entropy loss, which inevitably results in the discrepancy of evaluation between training and inference. Accordingly, to further boost the image/video captioning model by amending the discrepancy, the authors of [72, 105] devise a policy gradient optimization approach to directly optimize the LSTM with expected sentence-level reward loss as

$$E_p(\mathbf{V}, \mathcal{S}) = -\mathbb{E}_{\mathcal{S}\sim p_\theta}[r(\mathcal{S})], \tag{5}$$

where θ denotes the parameters of the LSTM that schedule a policy p_θ for generating a sentence. $r(\mathcal{S})$ is the reward measured by comparing the generated sentence $\mathcal{S}$ to ground-truth sentences over a non-differentiable evaluation metric.

3.4 Semantic Segmentation

Semantic segmentation is one of the most challenging tasks in multimedia and vision, which attempts to understand the semantic meaning of every pixel. It assigns each pixel of an image a semantic label. Compared to classification and detection, semantic segmentation requires dense pixel-wise predictions and, thus, is much more challenging.

3.4.1 Traditional Approaches. Techniques developed before deep learning mostly rely on CRF and operate on pixels or superpixels, where local evidence is incorporated in unary potential, and label interactions are encoded with binary potentials.

3.4.2 FCN. A milestone for deep-based semantic segmentation is the Fully Convolutional Network (FCN) [74], which introduces a full CNN architecture. It removes all fully connected layers; thus, the input can be of arbitrary sizes and the output is structured for dense prediction. FCN employs deconvolution for upsampling after a series of convolutions, making it an efficient and end-to-end architecture for per-pixel tasks such as semantic segmentation. Semantic segmentation based on the FCN has been widely adopted in the literature and also extended to video semantic segmentation, for example, [99].

3.4.3 Encoder-Decoder. Following the FCN, various techniques have been proposed for further improvement. An encoder-decoder network is featured by its bottleneck and symmetric structure, where the encoder gradually reduces the spatial dimension and the decoder recovers the details with increasing spatial dimension. The most representative work is SegNet [4], where the upsampling in the decoder is performed by using the pooling indices computed in the max pooling step of the corresponding encoder. This makes learning and inference more efficient. SegNet is further extended to Bayesian SegNet [50] by adding a DropOut layer. Moreover, shortcuts (e.g., U-Net [107]) between encoder and decoder are usually adopted to better recover the details.

3.4.4 Dilated/Atrous Convolution. Dilated convolution [146] keeps the output resolution and avoids upsampling, which uses dilated convolutions to aggregate multi-scale contextual information without losing resolution.

DeepLab is a big branch for semantic segmentation. The main contribution includes the dilated convolutions, atrous spatial pyramid pooling (ASPP), and fully connected CRF. DeepLabv1 [12] uses atrous convolution to explicitly control the resolution at which feature responses are computed within deep CNNs. DeepLabv2 [13] uses ASPP to robustly segment objects at multiple scales with filters at multiple sampling rates and effective fields of views. DeepLabv3 [14] augments the ASPP module with an image-level feature to capture longer-range information. Atrous convolution is adopted to extract output features at different output strides during training and evaluation, which efficiently enables training BN at output stride = 16 and attains a high performance at output stride = 8 during evaluation. DeepLabv3+ [15] extends [14] to include a simple yet effective decoder module to refine the segmentation, especially along object boundaries. Furthermore, in this encoder-decoder structure, one can arbitrarily control the resolution of extracted encoder features by atrous convolution to trade off precision and runtime.

3.4.5 Feature Ensemble. Another category is based on feature ensembling, which jointly considers multi-scale or multi-level features in the network. Typical networks include RefineNet [67] and PSPNet [151]. RefineNet adopts multi-path refinements by repeatedly using residual connections between upsampled multi-resolution feature maps. In contrast, PSPNet employs a pyramid pooling module to generate different resolution feature maps, which are further concatenated after several convolution lays and an upsampling lay. Both approaches achieve significant performance improvements on several benchmark datasets.

3.4.6 Real-Time Semantic Segmentation. Due to the intensive pixel-level prediction, most of the semantic segmentation approaches are time consuming and lack scalability. Some recent works focus on more efficient semantic segmentation algorithms, which are essential for real-time applications such as autonomous driving. Most of the works rely on the design principles proposed in efficient classification networks. For example, the architecture proposed in [91] consists of a large encoder and a small decoder. In addition, the bottleneck module and small convolution kernels are frequently used to reduce the number of parameters. A more straightforward way adopted in ShuffleSeg [26] is based on grouped convolution and channel shuffling, which are derived from ShuffleNet. In [145], a spatial path and a context path are introduced to preserve spatial information with a small stride and obtain sufficient receptive field with a fast downsampling strategy. In addition, a lightweight network based on Xception is used as the backbone.

Table 1. Top-5 Errors on ImageNet 2012 Validation Set

Model	Top-1 error	Top-5 error	Year
AlexNet [56]	42.8	19.7	2012
VGG16 [114]	28.5	9.1	2014
VGG19 [114]	24.7	7.5	2014
GoogleNet [118]	29	9.2	2014
Inception V2 [119]	23.4	-	2015
Inception V3 [119]	21.8	5.9	2015
ResNet50 [37]	24.7	7.8	2015
ResNet152 [37]	23	6.7	2015
Xception [17]	21	5.5	2016
DenseNet-264 [43]	22.15	6.12	2017
SqueezeNet [44]	42.5	19.7	2017
MobileNet [40]	29.4	10.5	2017
ShuffleNet [149]	29.1	10.2	2017
CondenseNet [42]	26.2	8.3	2017

All values are reported as percentage (%).

3.4.7 GAN. It is also worth noting another interesting branch [92, 155] based on the recent GAN framework, where segmentation is modeled in an adversarial learning manner. A convolutional segmentation network (generator) is trained along with an adversarial network that discriminates segmentation maps coming either from the ground truth or from the generator. In this way, the adversarial loss penalizes higher-order inconsistencies between ground truth segmentation maps and the ones produced by the generator. By the end of training, the generated masks are indistinguishable from ground-truth masks.

3.4.8 Weak Supervision. As most semantic segmentation methods rely heavily on the pixel-level annotations that require expensive labeling, many researchers also exploit alternative weak supervision, such as instance-level bounding boxes [20], image-level tags [94], and cross-domain annotations [150] for semantic segmentation. To achieve this target, techniques such as multiple instance learning, EM algorithm, constrained CNN, and transfer learning are adopted in the literature.

4 BENCHMARKS

This section reviews several popular benchmarks on multimedia analytics and state-of-the-art advancements on their benchmarks.

4.1 Classification

Several benchmark datasets have been proposed for evaluating the performance of classification approaches. The most popular one is ImageNet 2012 [109] which provides 1.28 million training images with annotations for 1,000 classes. It has been the standard dataset for demonstrating different methods. Performance is reported by using the top-N error rate, which indicates the percentage of misclassified testing samples. A given image is misclassified when any of its top-N highest-confidence output labels cannot match its ground-truth class. In Table 1, we summarize the top-1 errors reported in the literature. We can see that the performance has been improved significantly using deeper and more complex networks. On the other hand, increasing attention is paid to the design of efficient networks while keeping an acceptable classification accuracy (e.g., MobileNet,

Table 2. VOC2007 Test Detection Results

Model	Training data	**mAP**	**Year**
R-CNN [28]	07	66	2013
SPPNet [36]	07	68.1	2014
Fast R-CNN [27]	07	66.9	2015
Fast R-CNN [27]	07+12	70	2015
Faster R-CNN [104]	07	69.9	2017
Faster R-CNN [104]	07+12	73.2	2017
SSD300 [73]	07	68	2015
SSD300 [73]	07+12	74.3	2015
YOLO [101]	07+12	63.4	2015
YOLOv2 [102]	07+12	73.4	2017
DSOD300 [112]	07+12	76.3	2017

All values are reported as percentage (%).

ShuffleNet). In addition, some efficient versions of classical networks are proposed by using the basic components (depth-wise convolution) of the compact networks. Actually, compact networks have been proven to be effective and efficient in many applications, where the number of categories are much less than 1,000, as in the ImageNet benchmark.

4.2 Detection

The most popular benchmarks for object detection are VOC2007 [24], VOC2012 [24], and COCO [71], each of which includes bunches of images for several object categories. In VOC2007, all of the annotations for 20 objects of the training, validation, and testing sets are released. We have observed much more complete performance evaluations on this dataset. Thus, we will summarize only the results on the VOC2007 testing set by using different training data from VOC2007 or VOC2012 as reported in the literature. The conclusion is similar across VOC2012 and COCO datasets. Table 2 lists the mAP of different approaches averaged over 20 objects. We can see that the Faster R-CNN performs best among the two-stage architectures. SSD and YOLO achieve similar performances. As the performance of object detection is affected by several factors, such as backbone networks and input size, suitable design of one-stage architecture would have comparable performance with the complicated two-stage networks. However, as reported in much of the literature, one-stage approaches exhibit a bit lower mAP on small objects.

4.3 Captioning

A number of datasets have been built specifically to support the research on image/video captioning. Each benchmark contains pairs of an image/video and its corresponding sentences annotated by humans. This section summarizes several widely adopted image/video captioning benchmarks and the corresponding evaluation metrics, followed by the quantitative results of some representative methods.

Datasets. COCO [71] is the most popular benchmark for image captioning, which contains 82,783 training images and 40,504 validation images. There are 5 human-annotated descriptions per image. As the annotations of the official testing set are not publicly available, most existing methods follow the widely used settings in [105, 141] and take 113,287 images for training, 5K for validation and 5K for testing. Moreover, all the descriptions in training set are commonly converted to lowercase and some rare words that occur less than 5 times are discarded, resulting in a final vocabulary with 10,201 unique words in the COCO dataset.

Microsoft Research Video Description Corpus (MSVD) [10] is a widely used video captioning benchmark that contains 1,970 YouTube snippets collected on Amazon Mechanical Turk (AMT) by requesting workers to pick short clips depicting a single activity. The video clips are then labeled with single-sentence descriptions by annotators. The original corpus has multilingual descriptions and only the English descriptions are commonly exploited on video captioning tasks. There are roughly 40 available English descriptions per video and the standard split of MSVD is 1,200 videos for training, 100 for validation and 670 for testing, as suggested in [32].

MSR Video to Text (MSR-VTT) [131] is a recent large-scale benchmark for video captioning tasks, which contains 10K Web video clips of 41.2 hours, covering the most comprehensive 20 categories obtained from a commercial video search engine, e.g., music, people, gaming, sports, and TV shows. Each clip is annotated with about 20 natural sentences by AMT workers. The training/validation/test split is provided by the authors with 6,513 clips for training, 2,990 for validation, and 497 for testing.

Evaluation Metrics. Five types of metrics are commonly used for quantitatively evaluating the results of image/video captioning: BLEU@N [90], METEOR [7], ROUGE-L [66], CIDEr-D [122], and SPICE [1]. BLEU@N is a popular machine translation metric that measures the fraction of N-gram (up to 4-gram) that are in common between a hypothesis and a reference or set of references. However, as pointed out in [16], the N-gram matches for a high N (e.g., 4) rarely occur at a sentence level, resulting in poor performance of BLEU@N, especially when comparing individual sentences. Hence, another more effective evaluation metric, METEOR, is used along with BLEU@N, which is also widely used in the natural-language processing (NLP) community. In contrast to BLEU@N, METEOR computes unigram precision and recall, extending exact word matches to include similar words based on WordNet synonyms and stemmed tokens. ROUGE-L computes an F-measure with a recall bias using the longest common subsequence between the result sentence and each reference sentence. Another important metric for image/video captioning is CIDEr, which measures consensus in image/video captioning by performing a Term Frequency Inverse Document Frequency (TF-IDF) weighting for each N-gram. The above four kinds of evaluation metrics (i.e., BLEU@N, METEOR, ROUGE-L and CIDEr-D) are primarily sensitive to N-gram overlap, which is neither necessary nor sufficient for two sentences to convey the same meaning. Therefore, a novel evaluation metric, SPICE, was recently devised to measure how effectively captions recover objects, attributes, and the relations between them over scene graphs, which better simulates human judgment. All metrics can be computed by leveraging the codes[3] released by the COCO Evaluation Server [16].

Quantitative Results of Representative Methods. Most popular methods of image/video captioning have been evaluated on COCO [71], MSVD [10], and MSR-VTT [131]. We summarize the results on these three datasets in Tables 3, 4, and 5. As can be seen, most of the works are very recent, indicating that image/video captioning is an emerging and fast-developing research topic.

4.4 Semantic Segmentation

This section reviews the standard benchmarks of semantic segmentation.

Evaluation Metric. To assess semantic segmentation, mean Intersection-over-Union (mIoU) [24] has been widely adopted in the literature: IoU=TP/(TP+FP+FN), where TP, FP, and FN are true positives, false positives and false negatives, respectively.

Dataset. The most important dataset for semantic segmentation is Pascal VOC2012 [24]. The performances of almost all of the methods are reported on Pascal VOC2012 for fair comparison.

[3] https://github.com/tylin/coco-caption.

Table 3. Reported Results on COCO Dataset, where B@*N*, M, R, C and S are Short for BLEU@*N*, METEOR, ROUGE-L, CIDEr-D and SPICE Scores

	Cross-Entropy Loss						CIDEr Optimization					
	B@1	**B@4**	**M**	**R**	**C**	**S**	**B@1**	**B@4**	**M**	**R**	**C**	**S**
NIC (G) [126]	66.6	20.3	-	-	-	-	-	-	-	-	-	-
LRCN (G) [23]	69.7	27.8	22.9	50.8	83.7	15.8	-	-	-	-	-	-
HA (V) [132]	71.8	25.0	23.0	-	-	-	-	-	-	-	-	-
SA (V) [132]	70.7	24.3	23.9	-	-	-	-	-	-	-	-	-
ReviewNet (V) [135]	-	29.0	23.7	-	88.6	-	-	-	-	-	-	-
ATT (G) [144]	70.9	30.4	24.3	-	-	-	-	-	-	-	-	-
SC (V) [153]	71.6	30.1	24.7	-	97.0	-	-	-	-	-	-	-
LSTM-A$_3$ (G) [141]	73.5	32.4	25.5	53.9	99.8	18.5	-	-	-	-	-	-
LSTM-A$_5$ (G) [141]	73.4	32.6	25.4	54.0	100.2	18.6	-	-	-	-	-	-
Adaptive (R)[75]	74.2	33.2	26.6	-	108.5	-	-	-	-	-	-	-
FC-2K (R) [105]	-	29.6	25.2	52.6	94.0	-	-	31.9	25.5	54.3	106.3	-
Att2all (R) [105]	-	30.0	25.9	53.4	99.4	-	-	34.2	26.7	55.7	114.0	-
Up-Down (R) [2]	77.2	36.2	27.0	56.4	113.5	20.3	79.8	36.3	27.7	56.9	120.1	21.4
GCN-LSTM (R) [140]	77.4	37.1	28.1	57.2	117.1	21.1	80.9	38.3	28.6	58.5	128.7	22.1

The short name in the brackets indicates the image features, where V, G and R denotes VGG, GoogleNet and ResNet feature, respectively. All values are reported as percentage (%).

Table 4. Reported Results on MSVD Dataset, where B@*N*, M and C are Short for BLEU@*N*, METEOR and CIDEr-D Scores

Model	**B@1**	**B@2**	**B@3**	**B@4**	**M**	**C**
LSTM (A) [124]	-	-	-	31.2	26.9	-
TA (G+M) [136]	80.0	64.7	52.6	41.9	29.6	51.7
S2VT (V+O) [123]	-	-	-	-	29.8	-
LSTM-E (V+C) [85]	78.8	66.0	55.4	45.3	31.0	-
GRU-RCN (G) [6]	-	-	-	43.3	31.6	68.0
h-RNN (V+C) [147]	81.5	70.4	60.4	49.9	32.6	65.8
BAE (R+C) [8]	-	-	-	42.5	32.4	63.5
AF (V+C) [39]	-	-	-	52.4	32.0	68.8
LSTM-TSA (V+C) [88]	82.8	72.0	62.8	52.8	33.5	74.0

The short name in the brackets indicates the video features, where A, V, G, C, O, R and M denotes AlexNet, VGG, GoogleNet, C3D, optical flow, ResNet and motion feature learnt by 3D CNN on hand-crafted descriptors, respectively. All values are reported as percentage (%).

Other popular datasets include MSCOCO [71] and Cityscapes [19]. In this survey, we focus on Pascal VOC2012 to evaluate a wide range of methods.

Pascal VOC2012 [24] contains 21 classes in total: 20 foreground object classes and one background class. The dataset contains 1,464 (train), 1,449 (val), and 1,456 (test) pixel-level labeled images for training, validation, and testing, respectively. Cityscapes [19] mainly focuses on the street scene in 50 cities in different conditions, which includes 30 classes of annotation. Classes that are too rare are excluded, leaving 19 classes for evaluation. COCO [71] contains photos of 91 objects types frequently appeared in daily life. A total number of 2.5 million labeled instances in 328k images are included.

Table 5. Reported Results on MSR-VTT Dataset, Where B@*N*, M, C and R are Short for BLEU@*N*, METEOR, CIDEr-D and ROUGE-L Scores

Model	B@4	M	C	R
LSTM (G) [124]	33.5	24.2	34.1	54.1
LSTM (C) [124]	33.7	24.4	34.6	54.7
LSTM (G+C) [124]	34.1	24.8	35.5	55.8
LSTM (G+C+A) [124]	35.7	25.6	38.1	58.2
TA (G+C+A) [136]	34.8	25.1	36.7	57.1
LSTM-E (G+C+A) [85]	36.1	25.8	38.5	58.6
S2VT (G+C+A) [123]	36.0	26.0	39.1	58.4
AF (V+C) [39]	39.4	25.7	40.4	-
AF (V+C+A) [39]	39.7	25.5	40.0	-

The short name in the brackets indicates the video features, where V, G, C and A denotes VGG, GoogleNet, C3D and Audio feature, respectively. All values are reported as percentage (%).

Table 6. Reported Results on Pascal VOC2012 Dataset

Model	mIoU	Year
FCN [74]	62.2	2014
SegNet [4]	59.9	2015
Deeplabv1 [12]	71.6	2015
Dilated Conv [146]	75.3	2015
Deeplabv2 [13]	79.7	2016
RefineNet [67]	84.2	2016
PSPNet [151]	85.4	2016
GCN [93]	83.6	2017
Deeplabv3 [14]	86.9	2017
Deeplabv3+ [15]	89	2018

All values are reported in percentage (%).

Quantitative Results. Since almost all popular methods report their results on Pascal VOC2012 [24], we summarize the performances based on Pascal VOC2012. Please note that the official website also hosts a leaderboard[4] for comparison. Here, we briefly summarize several representative methods in Table 6 in chronological order. As shown, semantic segmentation has been prospective since 2014, and state-of-the-art performance has also been significantly improved to 89%. Notably, FCN serves as the milestone for subsequent methods, despite the primitive performance in 2014. The DeepLab family is active and performs strong in the timeline.

Several insights can be observed for decent performance by reviewing existing works. First, dilated convolution for large FoV is required, while keeping spatial information. Second, pyramid pooling for multi-level feature representations is essential. Third, skip connection can be adopted for feature fusion. Fourth, encoder-decoder networks can be explored for better incorporating context.

[4] http://host.robots.ox.ac.uk:8080/leaderboard/displaylb.php?challengeid=11&compid=6.

5 DISCUSSION AND OPEN ISSUE

In this survey, we have reviewed deep-learning techniques on four key topics related to multimedia analytics, including classification, detection, captioning, and segmentation. We present recent advances, showcase innovative methodologies and ideas, evaluate the state-of-the-art, and introduce popular benchmarks and challenges for these tasks. Though extensive efforts have been made on multimedia analytics with deep learning, we believe that we are still in the early stage of unleashing the power of deep learning in the era of big data. Given the substantial amounts of multimedia data generated every day, how to devise effective deep-learning models to facilitate multimedia content understanding remains an open problem. We hope that this survey will shed light on the nuts and bolts of multimedia analytics for both current and new researchers.

In addition, we pose several possible future research directions for each task. First, learning a class of deep nets in the generative model view will be a promising way to further boost recognition. Previous works focus more on general image classification. Leveraging efficient and compact networks in a more applicable way where candidate categories are much less than 1,000 is still not well studied. Furthermore, how to automatically search an effective network structure given a learning task is also an interesting topic. Second, the extensions of region-based methods or the exploration of recurrent neural networks in detection should be helpful for detecting and tracking objects simultaneously. Third, to reduce the high cost of collecting expert labeled data with pixel-level annotations for segmentation, an alternative way is to use synthetic data, which is largely available from computer games, and the ground truth could be freely generated automatically. One major obstacle in object detection and segmentation is the lack of accurate labeled training data. As we have observed large amounts of image-level annotations, using these weakly labeled training data to facilitate the object-level vision tasks will be a promising method. Finally, how to generate free-form sentences and support open vocabulary is vital to captioning in practical scenarios.

ACKNOWLEDGEMENT

This work was partially funded by National Natural Science Foundation of China No. 61602463, and the Open Project Program of the National Laboratory of Pattern Recognition (NLPR).

REFERENCES

[1] Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. 2016. SPICE: Semantic propositional image caption evaluation. In *Proceedings of the European Conference on Computer Vision*. Springer, 382–398.

[2] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. 2018. Bottom-up and top-down attention for image captioning and visual question answering. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*. IEEE Computer Society, 6077–6086.

[3] M. Arjovsky, S. Chintala, and L. Bottou. 2017. Wasserstein GAN. In *arXiv:1701.07875*.

[4] Vijay Badrinarayanan, Ankur Handa, and Roberto Cipolla. 2015. SegNet: A deep convolutional encoder-decoder architecture for robust semantic pixel-wise labelling. In *arXiv:1505.07293*.

[5] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. In *arXiv:1409.0473*.

[6] Nicolas Ballas, Li Yao, Chris Pal, and Aaron Courville. 2015. Delving deeper into convolutional networks for learning video representations. In *arXiv:1511.06432*.

[7] Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*. Association for Computational Linguistics, 65–72.

[8] Lorenzo Baraldi, Costantino Grana, and Rita Cucchiara. 2017. Hierarchical boundary-aware neural encoder for video captioning. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*. IEEE Computer Society, 3185–3194.

[9] Larry S. Davis and Bharat Singh. 2018. An analysis of scale invariance in object detection. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*. IEEE Computer Society, 3578–3587.

[10] David L. Chen and William B. Dolan. 2011. Collecting highly parallel data for paraphrase evaluation. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 190–200.

[11] Jingwen Chen, Ting Yao, and Hongyang Chao. 2018. See and chat: Automatically generating viewer-level comments on images. *Multimedia Tools and Applications*. (In press).

[12] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. 2014. Semantic image segmentation with deep convolutional nets and fully connected CRFs. In *arXiv:1412.7062*.

[13] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. 2018. DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40, 4 (2018), 834–848.

[14] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. 2017. Rethinking atrous convolution for semantic image segmentation. In *arXiv: 1706.05587*.

[15] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. 2018. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *arXiv:1802.02611*.

[16] Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C. Lawrence Zitnick. 2015. Microsoft COCO captions: Data collection and evaluation server. In *arXiv:1504.00325*.

[17] François Chollet. 2016. Xception: Deep learning with depthwise separable convolutions. In *arXiv:1610.02357*.

[18] Djork-ArnÃl' Clevert, Thomas Unterthiner, and Sepp Hochreiter. 2015. Fast and accurate deep network learning by exponential linear units (ELUs). In *arXiv:1511.07289*.

[19] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. 2016. The Cityscapes dataset for semantic urban scene understanding. In *arXiv:1604.01685*.

[20] Jifeng Dai, Kaiming He, and Jian Sun. 2015. BoxSup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation. In *IEEE International Conference on Computer Vision*. IEEE Computer Society, 1635–1643.

[21] Emily L. Denton, Soumith Chintala, Arthur Szlam, and Rob Fergus. 2015. Deep generative image models using a Laplacian pyramid of adversarial networks. In *Advances in Neural Information Processing Systems*. The MIT Press, 1486–1494.

[22] Jacob Devlin, Hao Cheng, Hao Fang, Saurabh Gupta, Li Deng, Xiaodong He, Geoffrey Zweig, and Margaret Mitchell. 2015. Language models for image captioning: The quirks and what works. In *Proceedings of Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 100–105.

[23] Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. 2015. Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*. IEEE Computer Society, 2625–2634.

[24] Mark Everingham, S. M. Eslami, Luc Gool, Christopher K. Williams, John Winn, and Andrew Zisserman. 2015. The Pascal visual object classes challenge: A retrospective. *Int. J. Comput. Vision* 111, 1 (2015), 98–136.

[25] Ali Farhadi, Mohsen Hejrati, Mohammad Amin Sadeghi, Peter Young, Cyrus Rashtchian, Julia Hockenmaier, and David Forsyth. 2010. Every picture tells a story: Generating sentences from images. In *Proceedings of the European Conference on Computer Vision*. Springer, 15–29.

[26] Mostafa Gamal, Mennatullah Siam, and Moemen Abdel-Razek. 2018. ShuffleSeg: Real-time semantic segmentation network. In *arXiv:1803.03816*.

[27] Ross Girshick. 2015. Fast R-CNN. In *Proceedings of the 2015 IEEE International Conference on Computer Vision*. IEEE Computer Society, 1440–1448.

[28] Ross B. Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. 2013. Rich feature hierarchies for accurate object detection and semantic segmentation. In *arXiv:1311.2524*.

[29] Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. *JMLR W&CP: Proceedings of the 13th International Conference on Artificial Intelligence and Statistics*, Vol. 9. 249–256.

[30] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, Vol. 2. 2672–2680.

[31] Alex Graves. 2013. Generating sequences with recurrent neural networks. In *arXiv:1308.0850*.

[32] Sergio Guadarrama, Niveda Krishnamoorthy, Girish Malkarnenkar, Subhashini Venugopalan, Raymond Mooney, Trevor Darrell, and Kate Saenko. 2013. Youtube2text: Recognizing and describing arbitrary activities using semantic hierarchies and zero-shot recognition. In *IEEE International Conference on Computer Vision*. IEEE Computer Society, 2712–2719.

[33] Richard H. R. Hahnloser, Rahul Sarpeshkar, Misha Mahowald, Rodney J. Douglas, and H. Sebastian Seung. 2000. Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit. *Nature* 405, 6789 (2000), 947–51.

[34] Jun Han and Claudio Moraga. 1995. *The Influence of the Sigmoid Function Parameters on the Speed of Backpropagation Learning*, Vol. 930, Lecture Notes in Computer Science. Springer, 195–201.

[35] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. 2017. Mask R-CNN. *In IEEE International Conference on Computer Vision.* IEEE Computer Society, 2980–2988.

[36] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2014. Spatial pyramid pooling in deep convolutional networks for visual recognition. In *arXiv:1406.4729.*

[37] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Deep residual learning for image recognition. In *arXiv:1512.03385.*

[38] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification. In *Proceedings of the IEEE International Conference on Computer Vision.* IEEE Computer Society, 1026–1034.

[39] Chiori Hori, Takaaki Hori, Teng-Yok Lee, Ziming Zhang, Bret Harsham, John R. Hershey, Tim K. Marks, and Kazuhiko Sumi. 2017. Attention-based multimodal fusion for video description. In *IEEE International Conference on Computer Vision.* IEEE Computer Society, 4203–4212.

[40] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. 2017. MobileNets: Efficient convolutional neural networks for mobile vision applications. In *arXiv:1704.04861.*

[41] Han Hu, Jiayuan Gu, Zheng Zhang, Jifeng Dai, and Yichen Wei. 2018. Relation networks for object detection. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition.* IEEE Computer Society, 3588–3597.

[42] Gao Huang, Shichen Liu, Laurens van der Maaten, and Kilian Q. Weinberger. 2017. CondenseNet: An efficient DenseNet using learned group convolutions. In *arXiv:1711.09224.*

[43] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. 2017. Densely connected convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition.* IEEE Computer Society, 2261–2269.

[44] Forrest N. Iandola, Song Han, Matthew W. Moskewicz, Khalid Ashraf, William J. Dally, and Kurt Keutzer. 2016. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size. In *arXiv:1602.07360.*

[45] Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on Machine Learning*, Vol. 37. Omnipress, 448–456.

[46] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. 2016. Image-to-image translation with conditional adversarial networks. In *arXiv:1611.07004.*

[47] Yu-Gang Jiang, Zuxuan Wu, Jinhui Tang, Zechao Li, Xiangyang Xue, and Shih-Fu Chang. 2018. Modeling multimodal clues in a hybrid deep learning framework for video classification. *IEEE Transactions on Multimedia* 20, 11 (2018), 3137–3147.

[48] Li Shen Jie Hu and Gang Sun. 2018. Squeeze-and-excitation networks. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition.* IEEE Computer Society, 7132–7141.

[49] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. 2017. Progressive growing of GANs for improved quality, stability, and variation. In *arXiv:1710.10196v2.*

[50] Alex Kendall, Vijay Badrinarayanan, and Roberto Cipolla. 2015. Bayesian SegNet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding. In *arXiv:1511.02680.*

[51] Taeksoo Kim, Moonsu Cha, Hyunsoo Kim, Jung Kwon Lee, and Jiwon Kim. 2017. Learning to discover cross-domain relations with generative adversarial networks. In *Proceedings of the International Conference on Machine Learning*, Vol. 70. 1857–1865.

[52] Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. In *arXiv:1412.6980.*

[53] Atsuhiro Kojima, Takeshi Tamura, and Kunio Fukunaga. 2002. Natural language description of human activities from video images based on concept hierarchy of actions. *International Journal of Computer Vision* 50, 2 (2002), 171–184.

[54] Tao Kong, Fuchun Sun, Anbang Yao, Huaping Liu, Ming Lu, and Yurong Chen. 2017. RON: Reverse connection with objectness prior networks for object detection. In *arXiv:1707.01691.*

[55] Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. 2017. Dense-captioning events in videos. In *Proceedings of the IEEE International Conference on Computer Vision.* IEEE Computer Society, 706–715.

[56] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. 2012. ImageNet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25.* 1097–1105.

[57] Girish Kulkarni, Visruth Premraj, Vicente Ordonez, et al. 2013. Babytalk: Understanding and generating simple image descriptions. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35, 12 (2013), 2891–2903.

[58] Wei-Sheng Lai, Jia-Bin Huang, Narendra Ahuja, and Ming-Hsuan Yang. 2017. Deep Laplacian pyramid networks for fast and accurate super-resolution. In *arXiv:1704.03915*.

[59] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. 1989. Backpropagation applied to handwritten zip code recognition. *Neural Comput.* 1, 4 (1989), 541–551.

[60] Dong Li, Zhaofan Qiu, Qi Dai, Ting Yao, and Tao Mei. 2018. Recurrent tubelet proposal and recognition networks for action detection. In *Proceedings of the IEEE International Conference on Computer Vision.* IEEE Computer Society, 306–322.

[61] Dong Li, Ting Yao, Lingyu Duan, Tao Mei, and Yong Rui. 2018. Unified spatio-temporal attention networks for action recognition in videos. *IEEE Transactions on Multimedia.* (In Press).

[62] Qing Li, Zhaofan Qiu, Ting Yao, Tao Mei, Yong Rui, and Jiebo Luo. 2016. Action recognition by learning deep multi-granular spatio-temporal video representation. In *Proceedings of the ACM on International Conference on Multimedia Retrieval.* ACM, 159–166.

[63] Qing Li, Zhaofan Qiu, Ting Yao, Tao Mei, Yong Rui, and Jiebo Luo. 2017. Learning hierarchical video representation for action recognition. *International Journal of Multimedia Information Retrieval* 6, 1 (2017), 85–98.

[64] Yehao Li, Ting Yao, Tao Mei, Hongyang Chao, and Yong Rui. 2016. Share-and-chat: Achieving human-level video commenting by search and multi-view embedding. In *ACM Multimedia.* ACM, 928–937.

[65] Yehao Li, Ting Yao, Yingwei Pan, Hongyang Chao, and Tao Mei. 2018. Jointly localizing and describing events for dense video captioning. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition.* IEEE Computer Society, 7492–7500.

[66] Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Proceedings of the ACL Workshop on Text Summarization Branches Out.* 10 pages.

[67] Guosheng Lin, Anton Milan, Chunhua Shen, and Ian D. Reid. 2016. RefineNet: Multi-path refinement networks for high-resolution semantic segmentation. In *arXiv:1611.06612*.

[68] Min Lin, Qiang Chen, and Shuicheng Yan. 2013. Network in network. In *arXiv:1312.4400*.

[69] Tsung-Yi Lin, Piotr Dollár, Ross B. Girshick, Kaiming He, Bharath Hariharan, and Serge J. Belongie. 2017. Feature pyramid networks for object detection. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition.* IEEE Computer Society, 936–944.

[70] Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, Kaiming He, and Piotr Dollár. 2017. Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision.* IEEE Computer Society, 2999–3007.

[71] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. 2014. Microsoft COCO: Common objects in context. In *Proceedings of the IEEE International Conference on Computer Vision.* IEEE Computer Society, 740–755.

[72] Siqi Liu, Zhenhai Zhu, Ning Ye, Sergio Guadarrama, and Kevin Murphy. 2017. Optimization of image description metrics using policy gradient methods. In *arXiv:1612.00370*.

[73] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, and Scott E. Reed. 2015. SSD: Single shot MultiBox detector. In *arXiv:1512.02325*.

[74] Jonathan Long, Evan Shelhamer, and Trevor Darrell. 2014. Fully convolutional networks for semantic segmentation. In *arXiv:1411.4038*.

[75] Jiasen Lu, Caiming Xiong, Devi Parikh, and Richard Socher. 2017. Knowing when to look: Adaptive attention via a visual sentinel for image captioning. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition.* IEEE Computer Society, 3242–3250.

[76] Ningning Ma, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun. 2018. ShuffleNet V2: Practical guidelines for efficient CNN architecture design. In *Proceedings of the IEEE International Conference on Computer Vision.* IEEE Computer Society, 122–138.

[77] Andrew L. Maas, Awni Y. Hannun, and Andrew Y. Ng. 2013. Rectifier nonlinearities improve neural network acoustic models. In *ICML Workshop on Deep Learning for Audio, Speech and Language Processing*, Vol. 28.

[78] Junhua Mao, Wei Xu, Yi Yang, Jiang Wang, and Alan L. Yuille. 2014. Explain images with multimodal recurrent neural networks. In *arXiv:arXiv:1410.1090.*

[79] Xiao-Jiao Mao, Chunhua Shen, and Yu-Bin Yang. 2016. Image restoration using very deep fully convolutional encoder-decoder networks with symmetric skip connections. In *Advances in Neural Information Processing Systems.* Curran Associates, 2810–2818.

[80] M. Mirza and S. Osindero. 2014. Conditional generative adversarial nets. In *arXiv:1411.1784*.

[81] Margaret Mitchell, Xufeng Han, Amit Goyal, et al. 2012. Midge: Generating image descriptions from computer vision detections. In *Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics.* The Association for Computer Linguistics, 747–756.

[82] Vinod Nair and Geoffrey E. Hinton. 2010. Rectified linear units improve restricted Boltzmann machines. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*. Omnipress, 807–814.

[83] Augustus Odena, Christopher Olah, and Jonathon Shlens. 2016. Conditional image synthesis with auxiliary classifier GANs. In *arXiv:1610.09585*.

[84] Yingwei Pan, Yehao Li, Ting Yao, Tao Mei, Houqiang Li, and Yong Rui. 2016. Learning deep intrinsic video representation by exploring temporal coherence and graph structure. In *Proceedings of the International Joint Conference on Artificial Intelligence*. AAAI Press, 3832–3838.

[85] Yingwei Pan, Tao Mei, Ting Yao, Houqiang Li, and Yong Rui. 2016. Jointly modeling embedding and translation to bridge video and language. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*. IEEE Computer Society, 4594–4602.

[86] Yingwei Pan, Zhaofan Qiu, Ting Yao, Houqiang Li, and Tao Mei. 2017. Seeing bot. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 1341–1344.

[87] Yingwei Pan, Zhaofan Qiu, Ting Yao, Houqiang Li, and Tao Mei. 2017. To create what you tell: Generating videos from captions. In *ACM Multimedia*. ACM, 1789–1798.

[88] Yingwei Pan, Ting Yao, Houqiang Li, and Tao Mei. 2017. Video captioning with transferred semantic attributes. In *arXiv:1611.07675*.

[89] Yingwei Pan, Ting Yao, Tao Mei, Houqiang Li, Chong-Wah Ngo, and Yong Rui. 2014. Click-through-based cross-view learning for image search. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*. 717–726.

[90] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: A method for automatic evaluation of machine translation. In *Proceedings of the Annual Meeting on Association for Computational Linguistics*. 311–318

[91] Adam Paszke, Abhishek Chaurasia, Sangpil Kim, and Eugenio Culurciello. 2016. ENet: A deep neural network architecture for real-time semantic segmentation. In *arXiv:1606.02147*.

[92] Pauline Luc, Camille Couprie, Soumith Chintala, and Jakob Verbeek. 2016. Semantic segmentation using adversarial networks. In *arXiv:1611.08408*.

[93] Chao Peng, Xiangyu Zhang, Gang Yu, Guiming Luo, and Jian Sun. 2017. Large kernel matters - improve semantic segmentation by global convolutional network. In *arXiv:1703.02719*.

[94] Pedro H. O. Pinheiro and Ronan Collobert. 2015. From image-level to pixel-level labeling with Convolutional Networks. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*. IEEE Computer Society, 1713–1721.

[95] B. T. Polyak and A. B. Juditsky. 1992. Acceleration of stochastic approximation by averaging. *SIAM J. Control Optim.* 30, 4 (July 1992), 838–855.

[96] Zhaofan Qiu, Qing Li, Ting Yao, Tao Mei, and Yong Rui. 2015. MSR Asia MSM at THUMOS Challenge 2015. In *THUMOS Challenge Workshop of the IEEE International Conference on Computer Vision and Pattern Recognition*. IEEE Computer Society.

[97] Zhaofan Qiu, Ting Yao, and Tao Mei. 2017. Deep quantization: Encoding convolutional activations with deep generative model. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*. IEEE Computer Society, 4085–4094.

[98] Zhaofan Qiu, Ting Yao, and Tao Mei. 2017. Learning spatio-temporal representation with pseudo-3D residual networks. In *Proceedings of the IEEE International Conference on Computer Vision*. IEEE Computer Society, 4489–4497.

[99] Zhaofan Qiu, Ting Yao, and Tao Mei. 2018. Learning deep spatio-temporal dependence for semantic video segmentation. *IEEE Transactions on Multimedia* 20, 4 (2018), 939–949.

[100] Alec Radford, Luke Metz, and Soumith Chintala. 2015. Unsupervised representation learning with deep convolutional generative adversarial networks. In *arXiv:1511.06434*.

[101] Joseph Redmon, Santosh Kumar Divvala, Ross B. Girshick, and Ali Farhadi. 2015. You only look once: Unified, real-time object detection. In *arXiv:1506.02640*.

[102] Joseph Redmon and Ali Farhadi. 2016. YOLO9000: Better, faster, stronger. In *arXiv:1612.08242*.

[103] Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. 2016. Generative adversarial text-to-image synthesis. In *Proceedings of the International Conference on International Conference on Machine Learning*, Vol. 48. JMLR, 1060–1069.

[104] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. 2017. Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell.* 39, 6 (2017), 1137–1149.

[105] Steven J Rennie, Etienne Marcheret, Youssef Mroueh, Jerret Ross, and Vaibhava Goel. 2017. Self-critical sequence training for image captioning. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*. IEEE Computer Society, 1179–1195.

[106] Marcus Rohrbach, Wei Qiu, Ivan Titov, Stefan Thater, Manfred Pinkal, and Bernt Schiele. 2013. Translating video content to natural language descriptions. In *Proceedings of the IEEE International Conference on Computer Vision*. IEEE Computer Society, 433–440.

[107] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-Net: Convolutional networks for biomedical image segmentation. In *arXiv:1505.04597*.

[108] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. 1988. Learning Representations by Back-propagating Errors. In *Neurocomputing: Foundations of Research*. MIT Press, 696–699.

[109] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, et al. 2015. ImageNet large scale visual recognition challenge. *International Journal of Computer Vision (IJCV)* 115, 3 (2015), 211–252.

[110] David Saad. 1998. *On-line Learning in Neural Networks*. Cambridge University Press, New York, NY.

[111] Dominik Scherer, Andreas Müller, and Sven Behnke. 2010. Evaluation of pooling operations in convolutional architectures for object recognition. In *Proceedings of the 20th International Conference on Artificial Neural Networks: Part III*. 92–101.

[112] Zhiqiang Shen, Zhuang Liu, Jianguo Li, Yu-Gang Jiang, Yurong Chen, and Xiangyang Xue. 2017. DSOD: Learning deeply supervised object detectors from scratch. In *Proceedings of the IEEE International Conference on Computer Vision*. IEEE Computer Society, 1937–1945.

[113] Karen Simonyan and Andrew Zisserman. 2014. Two-stream convolutional networks for action recognition in videos. In *Proceedings of the International Conference on Neural Information Processing Systems*, Vol 1. MIT Press, Cambridge, 568–576.

[114] K. Simonyan and A. Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. In *arXiv: 1409.1556*.

[115] Shuochen Su, Mauricio Delbracio, Jue Wang, Guillermo Sapiro, Wolfgang Heidrich, and Oliver Wang. 2016. Deep video deblurring. In *arXiv:1611.08387*.

[116] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *NIPS*.

[117] Christian Szegedy, Sergey Ioffe, and Vincent Vanhoucke. 2016. Inception-v4, inception-ResNet and the impact of residual connections on learning. In *arXiv:1602.07261*.

[118] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott E. Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. 2014. Going deeper with convolutions. In *arXiv:1409.4842*.

[119] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. 2015. Rethinking the inception architecture for computer vision. In *arXiv:1512.00567*.

[120] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. 2015. Learning spatiotemporal features with 3D convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*. IEEE Computer Society, 4489–4497.

[121] Dmitry Ulyanov, Andrea Vedaldi, and Victor S. Lempitsky. 2016. Instance normalization: The missing ingredient for fast stylization. In *arXiv:1607.08022*.

[122] Ramakrishna Vedantam, C. Lawrence Zitnick, and Devi Parikh. 2015. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*. IEEE Computer Society, 4566–4575.

[123] S. Venugopalan, M. Rohrbach, J. Donahue, R. Mooney, T. Darrell, and K. Saenko. 2015. Sequence to sequence - video to text. In *Proceedings of the IEEE International Conference on Computer Vision*. IEEE Computer Society, 4534–4542.

[124] Subhashini Venugopalan, Huijuan Xu, Jeff Donahue, Marcus Rohrbach, Raymond Mooney, and Kate Saenko. 2015. Translating videos to natural language using deep recurrent neural networks. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, 1494–1504.

[125] Dumoulin Vincent and Visin Francesco. 2016. A guide to convolution arithmetic for deep learning. In *arXiv:1603.07285*.

[126] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. 2015. Show and tell: A neural image caption generator. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*. IEEE Computer Society, 3156–3164.

[127] Qi Wu, Chunhua Shen, Lingqiao Liu, Anthony Dick, and Anton van den Hengel. 2016. What value do explicit high level concepts have in vision to language problems?. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*. IEEE Computer Society, 203–212.

[128] Zuxuan Wu, Yu-Gang Jiang, Xi Wang, Hao Ye, and Xiangyang Xue. 2016. Multi-stream multi-class fusion of deep networks for video classification. *ACM Multimedia* (2016). ACM, 791–800.

[129] Zuxuan Wu, Yu-Gang Jiang, Xi Wang, Hao Ye, and Xiangyang Xue. 2015. Modeling spatial-temporal clues in a hybrid deep learning framework for video classification. *Proceedings of ACM Multimedia* (2015). ACM, 461–470.

[130] Saining Xie, Ross B. Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. 2016. Aggregated residual transformations for deep neural networks. In *arXiv:1611.05431*.

[131] Jun Xu, Tao Mei, Ting Yao, and Yong Rui. 2016. MSR-VTT: A large video description dataset for bridging video and language. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*. IEEE Computer Society, 1063–6919.

[132] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, et al. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *Proceedings of the International Conference on Machine Learning*. PMLR, 2048–2057.

[133] Ning Xu, Brian Price, Scott Cohen, and Thomas Huang. 2017. Deep image matting. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*. IEEE Computer Society, 311–320.

[134] Yezhou Yang, Ching Lik Teo, Hal Daumé III, and Yiannis Aloimonos. 2011. Corpus-guided sentence generation of natural images. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 444–454.

[135] Zhilin Yang, Ye Yuan, Yuexin Wu, William W. Cohen, and Ruslan R. Salakhutdinov. 2016. Review networks for caption generation. In *Proceedings of the Advances in Neural Information Processing Systems*. 2361–2369.

[136] L. Yao, A. Torabi, K. Cho, N. Ballas, C. Pal, H. Larochelle, and A. Courville. 2015. Describing videos by exploiting temporal structure. In *Proceedings of the IEEE International Conference on Computer Vision*. IEEE Computer Society, 4507–4515.

[137] Ting Yao, Yehao Li, Zhaofan Qiu, Fuchen Long, Yingwei Pan, Dong Li, and Tao Mei. 2017. MSR Asia MSM at ActivityNet challenge 2017: Trimmed action recognition, temporal action proposals and dense-captioning events in videos. In *CVPR ActivityNet Challenge Workshop*.

[138] Ting Yao, Tao Mei, and Chong-Wah Ngo. 2015. Learning query and image similarities with ranking canonical correlation analysis. In *Proceedings of the IEEE International Conference on Computer Vision*. IEEE Computer Society, 28–36.

[139] Ting Yao, Yingwei Pan, Yehao Li, and Tao Mei. 2017. Incorporating copying mechanism in image captioning for learning novel objects. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*. IEEE Computer Society, 5263–5271.

[140] Ting Yao, Yingwei Pan, Yehao Li, and Tao Mei. 2018. Exploring visual relationship for image captioning. In *Proceedings of the European Conference on Computer Vision*. Springer, 711–727.

[141] Ting Yao, Yingwei Pan, Yehao Li, Zhaofan Qiu, and Tao Mei. 2017. Boosting image captioning with attributes. In *Proceedings of the IEEE International Conference on Computer Vision*. IEEE Computer Society, 4904–4912.

[142] Zili Yi, Hao Zhang, Ping Tan, and Minglun Gong. 2017. DualGAN: Unsupervised dual learning for image-to-image translation. In *Proceedings of the IEEE International Conference on Computer Vision*. IEEE Computer Society, 2868–2876.

[143] Yibo Yang, Zhisheng Zhong, Tiancheng Shen, and Zhouchen Lin. 2018. Convolutional neural networks with alternately updated clique. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*. IEEE Computer Society, 2413–2422.

[144] Quanzeng You, Hailin Jin, Zhaowen Wang, Chen Fang, and Jiebo Luo. 2016. Image captioning with semantic attention. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*. IEEE Computer Society, 4651–4659.

[145] Changqian Yu, Jingbo Wang, Chao Peng, Changxin Gao, Gang Yu, and Nong Sang. 2018. BiSeNet: Bilateral segmentation network for real-time semantic segmentation. In *Proceedings of the European Conference on Computer Vision*. Springer, 334–349.

[146] Fisher Yu and Vladlen Koltun. 2015. Multi-scale context aggregation by dilated convolutions. In *arXiv:1511.07122*.

[147] Haonan Yu, Jiang Wang, Zhiheng Huang, Yi Yang, and Wei Xu. 2016. Video paragraph captioning using hierarchical recurrent neural networks. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*. IEEE Computer Society, 4584–4593.

[148] H. Zhang, T. Xu, H. Li, S. Zhang, X. Huang, X. Wang, and D. Metaxas. 2017. StackGAN: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *arXiv:1612.03242*.

[149] Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. 2017. ShuffleNet: An extremely efficient convolutional neural network for mobile devices. *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*. IEEE Computer Society, 6848–6856.

[150] Yiheng Zhang, Zhaofan Qiu, Ting Yao, Dong Liu, and Tao Mei. 2018. Fully convolutional adaptation networks for semantic segmentation. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*. IEEE Computer Society, 6810–6818.

[151] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. 2016. Pyramid scene parsing network. In *arXiv:1612.01105*.

[152] Nuno Vasconcelos Zhaowei Cai. 2018. Cascade R-CNN: Delving into high quality object detection. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*. IEEE Computer Society, 6154–6162.

[153] Luowei Zhou, Chenliang Xu, Parker Koch, and Jason J Corso. 2016. Image caption generation with text-conditional semantic attention. In *arXiv:1606.04621*.

[154] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. 2017. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision*. IEEE Computer Society, 2242–2251

[155] Wentao Zhu, Xiang Xiang, Trac D. Tran, and Xiaohui Xie. 2016. Adversarial deep structural networks for mammographic mass segmentation. In *arXiv:1612.05970*.

Received June 2018; revised September 2018; accepted September 2018