International Journal of Engineering Science Invention (IJESI)
ISSN (Online): 2319 – 6734, ISSN (Print): 2319 – 6726
www.ijesi.org ||Volume 7 Issue 5 Ver. V|| May 2018 || PP 58-62

Zero-shot learning with Fast RCNN and Mask RCNN through semantic attribute Mapping

Jilsha P J[1], Chinchu Krishna[2]
[1]*(PG student, Rajagiri School of Engineering and Technology, India)*
[2]*(Assistant Professor, Rajagiri School of Engineering and Technology, India)*
Corresponding author;Jilsha P J

Abstract: *Machine learning applications increasingly address large and diverse data sets. While this leads to an abundance of some classes of data, it also uncovers objects that defy categorization in previously seen classes. Learning from sparse data set of images is challenging. The task of classifying samples taken from categories where no training examples exist is known as zero-shot learning. This problem has interest both from the practical standpoint of automatically labelling novel items, thereby saving the time needed to retrain classifiers and from the scientific standpoint of understanding how humans perform the task. Zero shot learning is a hierarchical approach to image classification problems that have no training examples. Zero shot learning is a combination of hierarchical and attribute based classification methods; the former gives a projected position in an established hierarchy, while the latter provides a ranked listing of potential classes and their estimated probabilities. Here we are using two hierarchical methods, Fast RCNN and Mask RCNN. Finally we compare the performance of both and will find the most efficient one.*

Keywords: *Fast RCNN ,Hierarchical classification ,Mask RCNN ,Semantic attribute mapping ,Zero-shot learning*

Date of Submission: 07-05-2018 Date of acceptance: 22-05-2018

I. Introduction

Zero-shot learning (ZSL) refers to a special object recognition paradigm where partial classes do not have any training image available but with some semantic descriptions of these classes, and then at the test phase try to recognize the images of these never seen classes. Besides, there usually provide some auxiliary classes with a number of labeled images and also their semantic descriptions. The success of zero-shot learning will promote large-scale object recognition for modern machine vision systems, which is very important for real-scenario applications such as automatic driving system, where there are tremendous unexpected objects and scenes need to be recognized for benefitting the right decision control. Despite great success have been made on general supervised learning paradigm with longtime improving techniques as feature representation learning, large capacity and flexible model training, and also availability of large image data and economical computation resource, some inherent limitations of supervised learning make it hard for implementing large-scale object recognition system, such as the involved great manual labeling cost. Alternatively, zero-shot learning will greatly improve modern machine vision system with better autonomic and adaptability. However, zero-shot learning is a nontrivial problem. There is some fundamentals assumptions and motivation for zero-shot learning to work reasonably. Generally, three conditions should be satisfied for a feasible zero-shot recognition system, i.e., [1] vector representation of objects classes should be provided either by human labeling or automatic mechanism, which is also called label embedding; [2] the embedding representation is competent for transferring knowledge from seen classes to unseen classes; [3] these embedding representation should be closely related to visual features in image domain, which ensures the convenience for implementing the mapping from image domain to the embedding space, i.e., image embedding. We can see that the embedding space bridges the semantic gap between low-level image features and high-level object classes. Attributes as an intermediate representation satisfy these conditions well and become a popular knowledge transferring media suitable for zero-shot learning.

Zero-shot learning is inherently a two stage process: training and inference. In the training stage, knowledge about the attributes is captured, and in the inference stage this knowledge is used to categorize instances among a new set of classes. Here our project is designed in such a way that, there are two phases, in the first phase we combine fast R-CNN and semantic attribute mapping; similarly in second phase we combine mask R-CNN and semantic attribute mapping for image classification. Finally we compare the results.

II. System Analysis And Design

2.1 Proposed System

The task of classifying samples taken from categories where no training examples exist is known as zero-shot learning. This problem has interest both from the practical standpoint of automatically labelling novel items (thereby saving the time needed to retrain classifiers) and from the scientific standpoint of understanding how humans perform the task. Zero shot learning is a combination of hierarchical classification and semantic attribute mapping. Hierarchical and attribute-based classification methods provide different types of information; the former gives a projected position in an established hierarchy while the latter provides a ranked listing of potential classes and their estimated probabilities.

There are two phases in the project as follows:

2.1.1 Phase 1: Implementing Fast R-CNN and apply semantic attribute mapping.

Fast R-CNN is a fast framework for object detection with deep ConvNets. Fast R-CNN,

- Trains state-of-the-art models, like VGG16, 9x faster than traditional R-CNN and 3x faster than SPPnet.
- Runs 200x faster than R-CNN and 10x faster than SPPnet at test-time.
- Has a significantly higher mAP on PASCAL VOC than both R-CNN and SPPnet.

The Fast RCNN method has several advantages

1. Higher detection quality (mAP) than R-CNN, SPPnet.
2. Training is single-stage, using a multi-task loss.
3. Training can update all network layers.
4. No disk storage is required for feature caching.

1. The ROI Pooling Layer

The RoI pooling layer uses max pooling to convert the features inside any valid region of interest into a small feature map with a fixed spatial extent of H x W (e.g., 7 x 7), where H and W are layer hyper-parameters that are independent of any particular RoI. AnRoI is a rectangular window into a conv feature map. Each RoI is defined by a four-tuple (r, c, h,w) that specifies its top-left corner (r, c) and its height and width (h,w). RoI max pooling works by dividing the h × w RoI window into an H × W grid of sub-windows of approximate size h/H × w/W and then max-pooling the values in each sub-window into the corresponding output grid cell. Pooling is applied independently to each feature map channel, as in standard max pooling.

2. Initializing from Pre-trained Networks

When a pre-trained network initializes a Fast R-CNN network, it undergoes three transformations: first, the last max pooling layer is replaced by a RoI pooling; second, the network's last fully connected layer and softmax are replaced with the two sibling layers described earlier; third, the network is modified to take two data inputs: a list of images and a list of RoIs in those images.

1. Fine-tuning for Detection

Training all network weights with back-propagation is an important capability of Fast R-CNN. First, let's elucidate why SPPnet is unable to update weights below the spatial pyramid pooling layer. The root cause is that back-propagation through the SPP layer is highly inefficient when each training sample (*i.e.* RoI) comes from a different image, which is exactly how R-CNN and SPPnet networks are trained. The inefficiency stems from the fact that each RoI may have a very largereceptive field, often spanning the entire input image. Sincethe forward pass must process the entire receptive field, thetraining inputs are large (often the entire image).Here more efficient training method that takesadvantage of feature sharing during training. In Fast RCNNtraining, stochastic gradient descent (SGD) minibatchesare sampled hierarchically, first by sampling N imagesand then by sampling R/N RoIs from each image.Critically, RoIs from the same image share computationand memory in the forward and backward passes. MakingN small decreases mini-batch computation. For example,when using N = 2 and R = 128, the proposed trainingscheme is roughly 64× faster than sampling one RoI from128 different images (*i.e.*, the R-CNN and SPPnet strategy).One concern over this strategy is it may cause slow trainingconvergence because RoIs from the same image are correlated.This concern does not appear to be a practical issueand we achieve good results with N = 2 and R = 128using fewer SGD iterations than R-CNN.In addition to hierarchical sampling, Fast R-CNN uses astreamlined training process with one fine-tuning stage thatjointly optimizes a softmax classifier and bounding-box regressors,rather than training a softmax classifier, SVMs,andregressors in three separate stages. The componentsof this procedure (the loss, mini-batch sampling strategy,back-propagation through RoI pooling layers, and SGD hyper-parameters) are described below.

4. Scale Invariance

We explore two ways of achieving scale invariant object detection: (1) via “brute force” learning and (2) by using image pyramids. In the brute-force approach, each imageis processed at a pre-defined pixel size during both training and testing. The network must directly learn scale-invariant object detection from the training data. The multi-scale approach, in contrast, provides approximate scale-invariance to the network through an imagepyramid. At test-time, the image pyramid is used to approximately scale-normalize each object proposal. During multi-scale training, we randomly sample a pyramid scale each time an image is sampled, as a form of data augmentation. We experiment with multi-scale training for smaller networks only, due to GPU memory limits.

Semantic attribute mapping

We use 3-class classifiers for each attribute, allowing for both yes and no states (denoted 1 and -1), as well as an in-between or \not-applicable" state (0). We build Na attribute classifier models from these data using Linear Support Vector Machines (LSVM) and use them to infer the feature to attribute mapping. In the indirect method, we first learn the posterior distribution on class given features. This is achieved through one-vs-all LSVM classification followed by Platt sigmoid scaling. To generate class estimates and probability distributions (both non-novel and novel), we employ the maximum likelihood (ML) method. This method uses the measured error rates of each attribute classifier in a validation data set as well as assumed independence of errors to compute posterior likelihoods of class given the inferred attribute vector. Critical to our purposes here, the ML method provides class posterior distribution estimates.

2.2.2 Phase 2: Implementing Mask R-CNN and apply semantic attribute mapping

Mask R-CNN is conceptually simple: Faster R-CNN has two outputs for each candidate object, a class label and a bounding-box offset; to this we add a third branch that outputs the object mask. Mask R-CNN is thus a natural and intuitive idea. But the additional mask output is distinct from the class and box outputs, requiring extraction of much finer spatial layout of an object. Next, we introduce the key elements of Mask R-CNN, including pixel-to-pixel alignment, which is the main missing piece of Fast/Faster R-CNN.

Faster R-CNN consists of two stages. The first stage, called a Region Proposal Network (RPN), proposes candidate object bounding boxes. The second stage, which is in essence Fast R-CNN , extracts features using RoIPool from each candidate box and performs classification and bounding-box regression. The features used by both stages can be shared for faster inference.

Mask R-CNN adopts the same two-stage procedure, with an identical first stage (which is RPN). In the second stage, in parallel to predicting the class and box offset, Mask R-CNN also outputs a binary mask for each RoI. This is in contrast to most recent systems, where classification depends on mask predictions. Formally, during training, we define a multi-task loss on each sampled RoI as L = Lcls + Lbox + Lmask. The classification loss Lcls and bounding-box loss Lbox. The mask branch has a Km2-dimensional output for each RoI, which encodes K binary masks of resolution m X m, one for each of the K classes. To this we apply a per-pixel sigmoid, and define Lmask as the average binary cross-entropy loss. For anRoI associated with ground-truth class k, Lmask is only defined on the k-th mask (other mask outputs do not contribute to the loss). Our definition of Lmask allows the network to generatemasks for every class without competition among classes; we rely on the dedicated classification branch to predict the class label used to select the output mask. This decouplesmask and class prediction. This is different from common practice when applying FCNs to semantic segmentation, which typically uses a per-pixel softmax and a multinomial cross-entropy loss. In that case, masks across classes compete; in our case, with a per-pixel sigmoid and a binary loss, they do not. We show by experiments that this formulation is key for good instance segmentation results.

Here the semantic attribute mapping is same as mentioned in the phase1. Finally compare the results of phase1 and phase 2 to find the better method.

III. Architecture Of The Proposed System

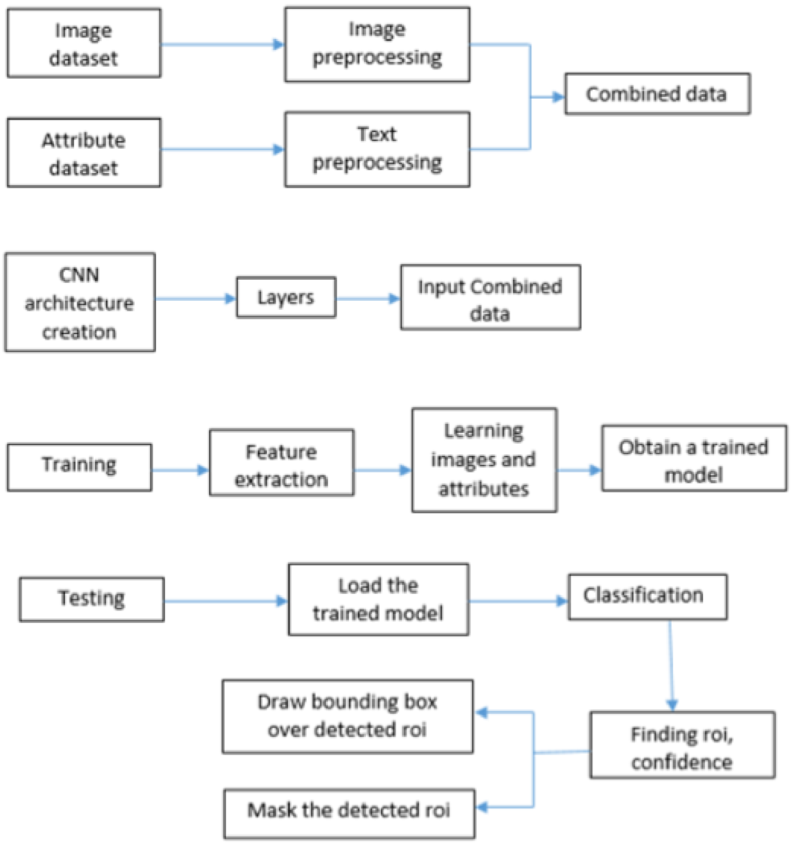

Fig 3.1: Architecture diagram of proposed system

Figure 3.1 shows the architecture diagram of the proposed system, initial step is to preprocess the data,openCv is utilized to preprocess the image dataset. Currently OpenCV bolsters a wide assortment of programming dialects like C++, Python, Java and so forth and is accessible on various stages including Windows, Linux, OS X, Android, iOS and so on. Likewise, interfaces in view of CUDA and OpenCL are additionally under dynamic improvement for fast GPU tasks.

OpenCV-Python is the Python API of OpenCV. It joins the best characteristics of OpenCV C++ API and Python language.NLP is utilized for content information preprocessing, at that point consolidate these preprocessed information. Second step is the formation of CNN architecture,for that we need to characterize layers first at that point give consolidate information as information. To manufacture the model, we utilize both tensorflow and keras.Flexible design of tensorflow permits simple sending of calculation over an assortment of stages (CPUs, GPUs, TPUs).Keras is an abnormal state neural systems API, written in Python and equipped for running over TensorFlow, CNTK, or Theano.Third step is preparing, in preparing stage separate the highlights and learn image and attributes to get a prepared model. Fourth step is testing,in this stage at first we stack the prepared model at that point arrange input information and discover RoI, certainty, etc.In instance of Fast-RCNN, draw bounding box over identified RoI. If there should arise an occurrence of Mask-RCNN, mask the distinguished RoI.

IV. Conclusion

Learning from sparse data set of images is challenging. Zero shot learning is a solution for this. It is a hierarchical approach to image classification problems that have no training examples, we evaluate and combine two different approaches to zero shot learning ie; hierarchical classification and mapping to semantic attributes. Thus learning from sparse data sets are made easier with zero-shot classification. Here we are using two hierarchical classification methods, Fast RCNN and Mask RCNN. Fast R-CNN is a fast framework for object detection with deep ConvNets. Mask RCNN introduces the pixel level segmentation and it is also the most recent object recognition method. Fast R-CNN and Mask R-CNN improve their performance by adding semantic attribute mapping. We can compare the efficiency of both to find the better method.

References

[1]. J, ared Markowitz, Aurora C. Schmidt, Philippe M. Burlina, I-JengWang,*"Hierarchical Zero-Shot Classification with Convolutional Neural Network Features and Semantic Attribute Learning"*, 2017 Fifteenth IAPR International Conference on Machine Vision Applications (MVA)

[2]. Yan Yan,"*Hierarchical Classification with Convolutional Neural Networks for Biomedical Literature",* International Journal of Computer Science and Software Engineering (IJCSSE), Volume 5, Issue 4, April 2016

[3]. Ross Girshick, "*Fast R-CNN*" , Microsoft Research , 2015 IEEE

[4]. Kaiming He, Ross Girshick,Shaoqing Ren and Jian Sun, "*Faster R-CNN: Towards real-time object detection with region proposal networks"* , arXiv:1506.01497v3 [cs.CV] 6 Jan 2016

[5]. Kaiming He, Georgia Gkioxari ,Piotr Doll´ar, Ross Girshick, "*Mask R-CNN*" , Facebook AI Research (FAIR), arXiv:1703.06870v2 [cs.CV] 5 Apr 2017

[6]. Philippe M. Burlina, Aurora C. Schmidt, I-Jeng Wang, *"Zero Shot Deep Learning from Semantic Attributes"* , 2015 IEEE 14th International Conference on Machine Learning and Applications

[7]. Ross Girshick,*"Rich feature hierarchies for accurate object detection and semantic segmentation*" , Microsoft Research , 2014 IEEE

[8]. Koen E. A. van de Sande, Jasper R. R. Uijlingst , Theo Gevers and Arnold W. M. Smeulders *,"Segmentation as selective search for object recognition"*, Computer Vision (ICCV), 2011 IEEE International Conference

[9]. FeiZang and Jiang-she Zhang *,"Softmax Discriminant Classifier",* 2011 Third International Conference on Multimedia Information Networking and Security

[10]. Yoshiyuki Yabuuchi, *" The difference between the formulations of possibilisticrobustregression model*", 2017 Joint 17th World Congress of International Fuzzy Systems Association and 9th International Conference on Soft Computing and Intelligent Systems (IFSA-SCIS)