Detecting Objects in Less Response Time for Processing Multimedia Events in Smart Cities

Asra Aslam
Insight Centre for Data Analytics
NUI Galway, Ireland
asra.aslam@insight-centre.org

Abstract

Due to increase in multimedia traffic in smart cities, we are facing the problem of processing unseen classes in real-time. Existing neural-network based object detectors may support this growing demand of multimedia data but have the limitation of availability of trained classifiers for unseen concepts. This results in a long waiting time for users who want to detect unseen classes. In this paper, we proposed three approaches where we can utilize existing object detection models and can train unseen classes within short training time. Our approaches are based on similarity of unseen classes with seen classes, and availability (presence or absence) of bounding boxes. Our results indicate that the proposed framework can achieve accuracy between 95.14% to 98.53% within response time of ∼0.01 min to ∼30 min for seen and partially unseen classes. Moreover we achieve state of the art results (68.78 mAP within 10 min) for unseen classes that have only image-level labels for training and no bounding boxes. Our qualitative results indicate that our approaches can work well for any unseen class (not only for conventional object detection datasets).

1. Introduction

Event-based multimedia approaches exhibit high performance in the current scenario but are designed for specific domains (like traffic management, security, supervision activities, terrorist attacks, natural hazards [15, 21, 30, 39, 53]) and hence can handle only familiar classes (have bounded/limited vocabulary). The escalating growth of multimedia data with large numbers of user subscriptions poses multiple challenges for the processing of image-based events in smart cities [2]. On the other hand, a large number of unseen concepts are emerging and changing over time in various domains in smart cities. Furthermore, the essential requirement of multimedia applications is a real-time performance [1,36], which needs to be fulfilled for its usability.

This highlights the need for minimization of response time while maintaining accuracy from the user’s perspective for new/unseen classes. However, the existing object detection approaches including few shot models [26, 27, 35, 41, 51] takes long training time (days or large number of resources) to train on new classes.

Consider the scenario of object detection for analyzing image based events in smart cities (shown in Fig. 1). Suppose a user subscribes for the detection of “Bus” on “Bus Stand”. This type of query can be answered “Public Transport Management” using a camera observing the bus stand and producing images consisting of bus-status related information. Similarly, if a user subscribes for the detection of the empty parking spot (i.e. absence of car at parking spot), we require another application for processing “Car Parking Management” events. Moreover, if a user subscribes to the concepts like “taxi” or “pedestrian”, then existing public transport and car parking management systems will not be able to respond to any new class even if they already consist of similar classes like “car” and “person”. Thus, we need a generalizable image event processing system that can provide adaptation from seen to unseen concepts (like car to taxi) and able to answer any completely unseen concept (like a cat, dog, key, bicycle, etc.) of any domain. Existing deep neural network based model can be very useful if they can be trained in short time for unseen concepts.

In this paper we proposed three approaches that can be used in different scenarios of training models for unseen classes. We summarize the main contributions as follows:

- Hyperparameter tuning based approach for completely new classes which requires training from scratch.
- Domain adaptation based approach for partially unseen classes where similar seen classifiers are available.
- Classifier to Detector conversion based approach for partially unseen classes which have some similarity with seen classes but does not have bounding boxes.
2. Related Work

Adaptive classifiers and Self-Tuning  Existing adaptive classifier based machine learning techniques [11, 13, 49, 58] in this category are designed with the aim of evolution of classifiers with drift in concept of multimedia streams. The identification of concept drift in these dynamic approaches, is mainly focused on processing of text data streams and cannot accommodate multimedia data streams.

Auto-WEKA [46] is one of the most popular work towards analyzing machine learning algorithms automatically and setting appropriate hyperparameters in order to enhance performance. Similarly hyperopt-sklearn is another available software mainly includes random search and TPE for the automatic selection [5]. Inspite of the fact that these tools are automatic, most of them focuses only on accuracy and generalization ability of classifiers, or on the computation cost consisting of testing time [19, 48], while excluding the training time of neural-network based models.

Transfer learning based methods  Many approaches [4, 40] with supervised/unsupervised transfer learning have been proposed for domain adaptation and mainly focused on the generalization ability for increasing accuracy not the overall response time. An event recognition in still images by transferring objects and scene representations has been proposed in [50], where the correlations of the concepts of object, scene, and events have been investigated. The authors proposed techniques to exploit the knowledge from other networks, and also evaluated the model on multiple event domains. Another domain adaptation approach based on the Faster R-CNN object detection model has been proposed [9] in order to reduce the domain discrepancy and enhance the effectiveness for cross-domain object detection. Such approaches are robust; however, in most cases, domain shift represents different changes in view-points, weather conditions, backgrounds, image quality, sketches, etc.

Weakly supervised learning  Recently, weakly supervised learning [25, 31, 47, 57] is emerging as a possible solution for large-scale unseen concepts. Such approaches are designed for limited classes and cannot incorporate new classes that have no pre-trained models. Weakly supervised learning [56] also formulated as a Multiple Instance Learning (MIL) problem. Most of the existing approaches [10, 16, 42, 43] are evaluated on classes of Pascal VOC [12], disregard the training time, and/or use Fast RCNN [14] as base network. Similarly, large scale domain adaptation based methods [17, 44] are also introduced particularly for the detection of objects and it is desirable to bring their abilities to core of multimedia event processing.

3. Approaches for Unseen Class Object Detection

The problem of multimedia event processing is divided into different scenarios shown in Fig. 2. Suppose a user subscribes for a concept, if we find a classifier that can detect subscribed concept, we call the concept “Seen” and recognize it with “Scenario 0”. In this case, we process the subscriptions directly using the existing classifier, without training any other model for the seen concept. However, if we don’t find any classifier to process the unseen concept, then we attempt to find “Any similar seen concept available?”. We use the names of seen classes and compute their individual similarities with the subscribed unseen class. In the worse case, if the concept is completely “unseen”, we introduce “Scenario 1” for the handling of subscriptions that are not related to any domain and resulted in low similarity scores. On the occurrence of complete “unseen” concept, we train the classifiers from scratch and optimize the training by hyperparameter tuning to reduce the overall response time of the multimedia event processing model. The most likely scenario is to receive the concept which is “unseen” and have similarity with one or more “seen” concepts. Thus, we can train classifiers for such unseen domains by...
knowledge transfer from seen domains (Scenario 2). Our final concern is the availability of bounding box annotations to train classifiers for subscribed concepts (Scenario 3). In this case, we use the classifier to detector conversion methods for the training of the detector for unseen classes.

3.1. Approach-I: Using Hyperparameters Tuning

Consider the scenario when the concept is completely “unseen”, i.e., there is no similar seen concept-based classifier available in the multimedia event processing model for the knowledge transfer. This specific problem associated with Scenario-1 is described in Fig. 3. For instance, a user subscribes for mirror detection and existing multimedia event processing model can detect only bicycle, bus, cat, traffic, person etc. In that case, we need to address the problem of classifier training for unseen concepts from scratch in low response time.

In this Approach-I, we incorporated object detection model with self hyperparameter tuning to train for unseen concepts in the required response time. As the choice of hyperparameter values greatly affects the performance of resulting classifiers, we leverage self-tuning of optimization hyperparameters, including the configuration of learning-rate, batch-size, and the number of epochs for minimizing the training time while maintaining the promising accuracy.

In this case, our model evaluates the training time and computes hyperparameters automatically based on prerequisite response time. Here, we train models from scratch as we assume no seen class-based classifiers are available, which is not true in most cases. Thus, we incorporate transfer learning to further improve performance in the next scenario.

3.2. Approach-II: Using Domain Adaptation

The problem associated with Scenario-2 is described in Fig. 4, where we need to reduce the response time for cases where the intended classifier is not available but similar classifiers are available. Suppose a user subscribes to the detection of class “car”, unseen to the multimedia event processing model. If a model already consists of related classifiers (like bus in the present case), we need to train classifiers for such partial unseen concepts in a reduced response time.

In Approach-II, on arrival of new concept, the proposed model first identifies if any similar classifier is available, or if there is any possibility for domain adaptation. Second, it performs the training of classifiers on need for the intended new subscription. More specifically for domain adaptation, we are using fine-tuning and freezing of classifier layer based methods. In this approach, we use the previously trained classifier to instantiate the network of
Figure 3. Scenario-1: Completely Unseen Concept Arrived (Approach-I: Hyper-Parameters Tuning for short Training Time)

Figure 4. Scenario-2: Partially Unseen Concept Arrived (Approach-II: Domain Adaptation for short Training Time)

Figure 5. Classifier to detector conversion for Unseen classes (like dog) that has only image-level labels and no bounding boxes.

Table 1. Hyperparameter values for Adaptive Training

<table>
<thead>
<tr>
<th>Hyperparameter for different Training Time</th>
<th>Model</th>
<th>Batch Size</th>
<th>Learning Rate</th>
<th>#Epochs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Defaults</td>
<td>YOLO</td>
<td>64</td>
<td>0.001</td>
<td>300</td>
</tr>
<tr>
<td>Hyperparameters for long training time</td>
<td>SSD</td>
<td>32</td>
<td>0.001</td>
<td>120</td>
</tr>
<tr>
<td></td>
<td>RetinaNet</td>
<td>1</td>
<td>1e-5</td>
<td>50</td>
</tr>
<tr>
<td>Our Derived Hyper-parameters for short</td>
<td>YOLO</td>
<td>64</td>
<td>0.005315</td>
<td>2</td>
</tr>
<tr>
<td>response time (≤ 15 min)</td>
<td>SSD</td>
<td>8</td>
<td>0.002612</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>RetinaNet</td>
<td>1</td>
<td>0.000195</td>
<td>5</td>
</tr>
<tr>
<td>Our Derived Hyper-parameters for average</td>
<td>SSD</td>
<td>4</td>
<td>0.003600</td>
<td>12</td>
</tr>
<tr>
<td>response time (≤ 60 min)</td>
<td>RetinaNet</td>
<td>2</td>
<td>0.000224</td>
<td>9</td>
</tr>
</tbody>
</table>

user to construct detectors for unseen classes without the need for detection data (no bounding boxes) within the short training time. Our model is based on making use of existing object detection datasets of bounded vocabulary (consists of seen concepts) to construct detectors for unseen concepts (i.e., unbounded vocabulary) by using the differences between a weak detector (trained on image classification dataset) and a strong detector (trained on object detection datasets).

In Approach-III, we train two separate detectors, one on existing object level labels (of MCOCO, and OID dataset) and another on image-level labels (using ImageNet dataset), respectively. Then Approach-III follows the below steps:

1. Download images using only image-level labels on request of any unseen concept (like dog).
2. The object-level detector is then fine-tuned on collected images of unseen concepts by labeling the most semantically similar class (like cat) with the unseen class name (like dog).
3. At this stage, we compute the visual similarity of the constructed unseen class detector (trained on classification data) with seen classes of image-level detector, combine it with semantic similarities, and select top-k classes ranked on comprehensive similarities. Visual similarity is presently the difference between the weights of the last layers of seen and unseen classes.
4. We transfer the knowledge of classifier-detector differences of top classes to the constructed unseen class detector and adapt it into the stronger detector without further training.
5. Finally, we perform the detection using our trained network communicate results.

In this approach all networks use YOLOv3 with MobileNetv3 backbone as our aim is to reduce response time and these are fast models for detection and classification.

4. Experiments

4.1. Implementation Details

Here, we utilized Pascal VOC, Microsoft COCO, and Open Images dataset (OID) for the construction of detectors. Specifically for Approach I, number of training images for the subscriptions cat, dog, laptop, car, bus, bicycle, and football classes are 1804, 2204, 5528, 2820, 847, 1108, and 4339. If bounding box annotations of image consist of any of the classes (cat, dog, laptop and so on), then added it to testing events set. The number of testing events for the same classes are 384, 538, 355, 1588, 256, 396, and 413 respectively. In Approach II, we added classes cricket ball, laptop bus, and mango classes consist of 95, 5528, and 126 training images; and 15, 355, and 23 testing images respectively.

In Approach III, object-level detector consist of 80 classes of Microsoft COCO [28] and 20 classes of OID [23]; while image-level detector consist of same 100 seen classes of ISLVRC [37] dataset. We chose unseen classes in such a way that the same classes should be present in OID (consist of 600 classes). So that testing dataset of OID can be served as groundtruth. For the training of 100 unseen classes, constant learning rate of $10^{-4}$ is used and evaluated on 0.5 IoU.

4.2. Comparison of performance before and after adaptation for short response time

We performed our investigation on different object detection models (YOLOv3 [34], SSD-300 [29], and RetinaNet [27]). Table 1 represents default hyperparameter values and our derived hyperparameter values for short training time. We derived the values for low and average training time by performing experiments on large number of trials and using TPE search method [6], which needed to be minimized based on mean average precision (mAP).

It can be observed in Table 2 that in most of the case our approach based hyperparameters gives us better mAP than default hyperparameters of object detection models. Here we took 15 min and 60 min as an example to represent less and average training time respectively. Using Approach I, we found that RetinaNet performs best on low and average training time.

4.3. Comparison of performance for domain adaptation for short response-time

We evaluate transfer learning techniques on same models, to analyze which classifiers can perform well on applying what type of training (scratch, fine-tuning, and freezing layers). The results of performance with response time trade-off are shown in Fig. 6. Firstly, it represents the trade-off on arrival of a completely new subscription, when there is no possibility of domain adaptation, for the training time of 120 min. In this case, all models are trained from scratch without the use of any pre-trained model. We can observe the performance of RetinaNet (Fig. 6c) is higher than other object detection models and the SSD model (Fig. 6b) is very difficult to converge with training from scratch, thus resulting in the worse performance, whereas the performance of YOLOv3 (Fig. 6a) is also low. However, by choosing training time (~30 min) using RetinaNet, we can reach accuracy $\sim 77.10\%$ with precision $\sim 0.21$.

The performance of RetinaNet and SSD are better than YOLOv3 in initial 30 min of training for both cases of fine-tuning (Fig. 6d, 6e, and 6f) and freezing (Fig. 6g, 6h, and 6i) layers. However there is a sudden rise in performance of YOLOv3 in the first few minutes signifies its higher slope in terms of short time training as compared to other object detection models. Trend lines are also shown with comparison, just to give the clear demonstration of initial precision that a particular object detection model can achieve at zero response time, as well as the highest precision of a model in a given response time. We can easily observe that all object detection models with fine-tuning technique are performing better as compared to the adaptation technique of freezing layers for long training time (> 120 min). However for short training time (< 30 min) YOLOv3 with freezing technique and RetinaNet with fine-tuning are performing best.

It is worth to mention that the testing time on our resources for YOLOv3, SSD300, and RetinaNet are 0.009 sec, 0.05 sec, and 0.08 sec for one image.

Table 3 shows four examples of classes on domain transfers with different similarity scores. It also presents Transfer Loss, Accuracy, and Distribution Discrepancy metrics computed for analyzing domain transfers (laptop to mango, mango to bus, cat to mango, and dog to mango).
(a) Training from Scratch (YOLOv3)
(b) Training from Scratch (SSD300)
(c) Training from Scratch (RetinaNet)
(d) Training by Fine-Tuning (YOLOv3)
(e) Training by Fine-Tuning (SSD300)
(f) Training by Fine-Tuning (RetinaNet)
(g) Training by Freezing (YOLOv3)
(h) Training by Freezing (SSD300)
(i) Training by Freezing (RetinaNet)

Figure 6. Performance vs Response Time with and without Domain Adaptation

Table 3. Analysis of Domain Adaptation using Transfer Loss, Accuracy, and A-Distance

<table>
<thead>
<tr>
<th>Classes with Domain Adaptation</th>
<th>Semantic Similarity</th>
<th>Transfer Loss</th>
<th>Accuracy</th>
<th>A-Distance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mango Detector from Laptop Detector</td>
<td>0.08</td>
<td>0.00%</td>
<td>14.21%</td>
<td>13.73%</td>
</tr>
<tr>
<td>Dog Detector from Cat Detector</td>
<td>0.20</td>
<td>0.24%</td>
<td>-0.97%</td>
<td>-7.25%</td>
</tr>
<tr>
<td>Cricket Ball Detector from Foot Ball Detector</td>
<td>0.33</td>
<td>0.25%</td>
<td>13.87%</td>
<td>23.89%</td>
</tr>
<tr>
<td>Bus Detector from Car Detector</td>
<td>0.50</td>
<td>2.96%</td>
<td>-3.51%</td>
<td>1.31%</td>
</tr>
</tbody>
</table>

Football to cricket ball, car to bus, and cat to dog). Transfer loss indicates how well the transfer works on multiple domains, and its lower values are recommended. In this case the best transfer is achieved by RetinaNet on the transfer of cat to dog class. However YOLOv3 achieve the best accuracy on all domain transfers. In-order to realize the variation of approximate distance (i.e. Distribution Discrepancy) among different domains, we have trained few binary classifiers that can classify source-target pair of classes. Here also YOLOv3 neural network closes the cross-domain gap more effectively as distribution discrepancy (lower is better) is lowest with it as compared to other object detection models. The qualitative detection performance after applying Approach-II on unseen class “bus” on an event of bus arrival and leaving of bus on bus stand is shown in Fig. 7.

Table 4 provides a comparison of average accuracy and response time of proposed with existing models by considering their best performance. It can be observed that existing domain specific models are designed only for the detection of specific objects and answer such seen (known) subscriptions in low response time, however they fail to process any unseen (unknown) subscription of different domain.

2049
Table 4. Comparison of Proposed with Existing Model(s)

<table>
<thead>
<tr>
<th>Approach</th>
<th>Example of Seen/Unseen Classes</th>
<th>Performance</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Response Time</td>
</tr>
<tr>
<td>Existing Domain</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Specific Models</td>
<td>Vehicle Detection [45] (Seen)</td>
<td>0.001 min</td>
</tr>
<tr>
<td></td>
<td>Firearm Detection [25] (Seen)</td>
<td>0.0007 min</td>
</tr>
<tr>
<td></td>
<td>Stolen Objects [38] (Seen)</td>
<td>0.17 min</td>
</tr>
<tr>
<td></td>
<td>Traffic Light, Key, Pedestrian, Ball etc. (Unseen)</td>
<td>∞</td>
</tr>
<tr>
<td>Proposed Approaches</td>
<td>Car, Football, Cat, Laptop etc. (Seen)</td>
<td>0.01 min</td>
</tr>
<tr>
<td></td>
<td>Bus, Dog, Mango, Cricket ball, Mirror etc. (Unseen)</td>
<td>15 min (low)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>60 min (average)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>30 min</td>
</tr>
<tr>
<td></td>
<td></td>
<td>10 min (no BBox)</td>
</tr>
</tbody>
</table>

The proposed model can achieve accuracy of 95.14% even when all concepts are unseen by taking an average response time of ~30 min. Also, the accuracy with Approach-I on low and average response time are 79.00% and 84.28% respectively. Moreover, with Approach-III we get 42.86% accuracy but without bounding boxes, and with training time of only 10 min (discussed in next section).

4.4. Analysis of Performance for domain adaptation without bounding boxes for short response time

To retrieve the effective range of response-time in our model, we train each category until the point testing accuracy starts to decrease (to avoid overfitting). We show a few examples of unseen concepts in Figure 8. Please note here we compute the total number of epochs for varying the training time. We first train our model on weak level labels (i.e., without bounding boxes) and then test on strong labels (i.e., with bounding boxes). We observe that the maximum mAP of each class could be achieved within 10 min of training. After that, the mAP decreases and remains constant. We recommend 10 min of training to attain maximum mAP 43.07. It is worth noting that mAP at 0 min training time on unseen classes is not zero due to the use of detectors trained on object-level labels for initialization.

We show few examples of our model detections for qualitative analysis in Fig. 9. Here unseen classes of Fig. 9 (a) – (d) consist of OID dataset images where groundtruths are shown in green color while our detections are shown in red color. Fig. 9 (e) – (h) consists of additional unseen classes which are present neither in any object detection nor in image classification datasets to date. Due to this reason we have only our detections (in red color) for these images.

Presently, existing few-shot object detection models are showing great promise by providing competitive performance with only few shots of annotated. Thus, we compared our model performance with recent zero-shot, one-shot, and few-shot detection approaches in Table 5. Due to lack of space, in case of few shot object detection we show here the performance of 10 shots of existing approaches [8, 22, 26, 41, 51, 52, 54, 55] which is best among all. It is important to note that we use only image-level labels; thus, our approach does not need any shot.

5. Conclusion

The problem of processing multimedia events that can include a large number of seen/unseen concepts belonging to the same or multiple domains are analyzed in this paper. We proposed approaches for completely unseen concepts, partially unseen concepts, and unseen concepts without bounding boxes. Our approaches utilized hyperparameter tuning, domain adaptation, and classifier to detector conversion method to training unseen classes detectors in short time. The proposed approach can achieve the accuracy ranges from 95.14% to 98.53% within ~ 0.01 min to ~ 30 min. Moreover, we achieve mAP 68.78 in training time of 10 min on unseen classes of without bounding boxes. Quali-
Figure 8. Examples of mAP with Response-Time. For each “Unseen” category, we use the top-10 weighted average nearest neighbor “Seen” categories for adaptation. This shows that maximum mAP could be achieved within 10 min.

Figure 9. Examples of detections of our model on “Unseen” categories shown in red color and groundtruth (taken from OID) in green. Last four unseen classes images are downloaded online, so no groundtruth available to date.

Figure 10. mAP of our model on 100 “Unseen” Categories within 10 min of training. We use the top-10 nearest neighbor “Seen” categories for each unseen category, with average similarity scores shown on top.

Table 5. Comparison with N-Shot detection models.

<table>
<thead>
<tr>
<th>Methods</th>
<th>mAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Zero-Shot [3,33]</td>
<td>15.32</td>
</tr>
<tr>
<td>One-Shot [7,18]</td>
<td>72.2</td>
</tr>
<tr>
<td>Few-Shot [8,22,26,41,51,52,54,55]</td>
<td>57.37</td>
</tr>
<tr>
<td>Ours*</td>
<td>68.78</td>
</tr>
</tbody>
</table>

*Like existing methods, mAP is computed for random 5-class splits.

Acknowledgments This publication has emanated from research conducted with the financial support of Science Foundation Ireland (SFI) under Grant Number SFI/12/RC/2289 P2, co-funded by the European Regional Development Fund.

References


[38] Juan Carlos San Miguel and José M Martínez. Robust unattended and stolen object detection by fusing simple algorithms. In 2008 IEEE Fifth International Conference on Advanced Video and Signal Based Surveillance, pages 18–25. IEEE, 2008. 7


[40] Baochen Sun, Jiashi Feng, and Kate Saenko. Return of frustratingly easy domain adaptation. In AAAI, volume 6, page 8, 2016. 2


