Open Images Challenge: Visual Relationship Detection Track

Abraham Jose
ID :5068109, CAP6614
abraham@knights.ucf.edu

Abstract

Visual Relationship detection is an important aspect of in Computer Vision and a challenging problem statement as we know it right now in the context of image understanding. The Visual Relationship track in Open Images Challenge from Google is a open image challenge from google to detect the visual relationships in an image including is,at,on,hold,wears,interacts with,under,hits,plays and holds. The data-sets is organized as triplets of relationships using the relationship phrases mentioned. The data-set is a collection of annotations of either an object and it's attribute or the relationship of two different objects. We will call these relationships as 'is'(attributes) and 'non-is'(non attributes) relationship. I have considered the 'is' data-set and used a Faster RCNN network to train on the 42 different unique classes in 'is' relationships. For the detection of 'non-is' triplets trained another model with 287 unique triplets for the detection along with the object detection for all the subjects and objects in non-is category. The performance of the 'is' data-set was significantly higher because the ground truth of the attribute and the object are in same bounding box. However, the results for the 'non-is' data-set using FRCNN was significantly poor.

1 Introduction

In the Computer Vision community, through CNN models and immense amount of data we were able to preform tasks such as object detection, image segmentation and image classification with good accuracy and performance. The deep learning techniques were able to improve the performance in these tasks over the period of time. However, many other tasks in computer vision are still not doing as good as detection or classification. One such tasks is understanding an image. It requires understanding of the image objects and the relationships between multiple entities in an image. The Visual Relationship data-set from google's Open Image data-set is geared towards this objective to understanding of image. In the data-set, we have multiple objects, relationships and attributes corresponding to objects in the image. This data-set is manually annotated and ground truth has no noise in the label. There are 329 triplets of object to object relationship and object to attribute relationship triplets in the image. Here I propose a object detection based architecture to train the triplet models and the for getting the predictions of each triplet relationship. The models for 'is' and 'non-is' is trained separately and ensembled both the models to create the prediction. In the relationships that are 'is' based, the object detection task was able to perform really well and in the 'non-is' task the proposed method performed poorly, which is why the model requires an external reinforcement using XG-Boost or Light-GBM to create a prediction. These gradient decent boosting methods helps in using the certain parameters like overlap, overlap %, euclidean distance between the center of the objects and normalized distance between the centers.

CAP 6614: Current Topics in Machine Learning (Fall 2019), UCF.

2 Dataset

2.1 Open Images Dataset V5

The Open Images challenge from google has one of the large database for the object detection, visual relationship detection and segmentation challenges. They host all the challenges each year like ImageNet data-set. The version 5 of the Open Image data-set is the 2019 challenge data-set of the Open Image data-set. The Open Image data-set consists almost 600 categories a15B bounding boxes or annotations. In Visual Relationship challenge, we have around 329 distinct relationships and attributes of objects. Distinct relationship triplets in the 'is' or attribute category is around 42 and 287 in the 9 distinct 'non-is' relationship.There are 23 and 57 classes respectively.
The training data-set has around 374,768 relationship and attribute triplets in the whole datset in 3.29 B bounding boxes.In total the whole data-set has almost 57 object classes, 10 different relationships and 5 different visual accounting towards the 329 distinct combinations of triplets including the relationship. To go around the model, I separated these distinct triplets and found the data-set which will correspond to each of the distinct relationship pairs and derived the bounding boxes corresponding to them.

2.2 IS Dataset Processing

The data pre-processing was mostly to remove any error data by checking the bounding boxes dimension with respect to the image and their height and width to see if the bounding boxes are strictly bounding in the case of 'is' data-set. In the 'is' data-set the relationship itself is attribute like 'chair is wooden' or 'fork is plastic'. So the bounding box of the object and the attributes like plastic, transparent, textile, leather and wooden will be the same. There are 42 object-attribute distinct pairs and hence we considered the data-set to be 42 classes in the 'is' relationship and trained a faster RCNN detection model for detection of these 42 distinct objects. Hence the only data-processing involved with 'is' relationship category is cross-validating the bounding boxes of each objects(checking if they have same bounding boxes in 'is' relationship) and actual image sizes to check for correctness. Also, I randomly included first 10,000 instances(relationship) in the category of 'non-is' relationship. By introducing this new negative class we will reduce the over-fitting on all 'is' relationship. Thus a Faster-RCNN model has been used with 42 different classes including positive and negative data-set to learn just the 'is' relationship.

2.3 NON-IS Dataset Processing

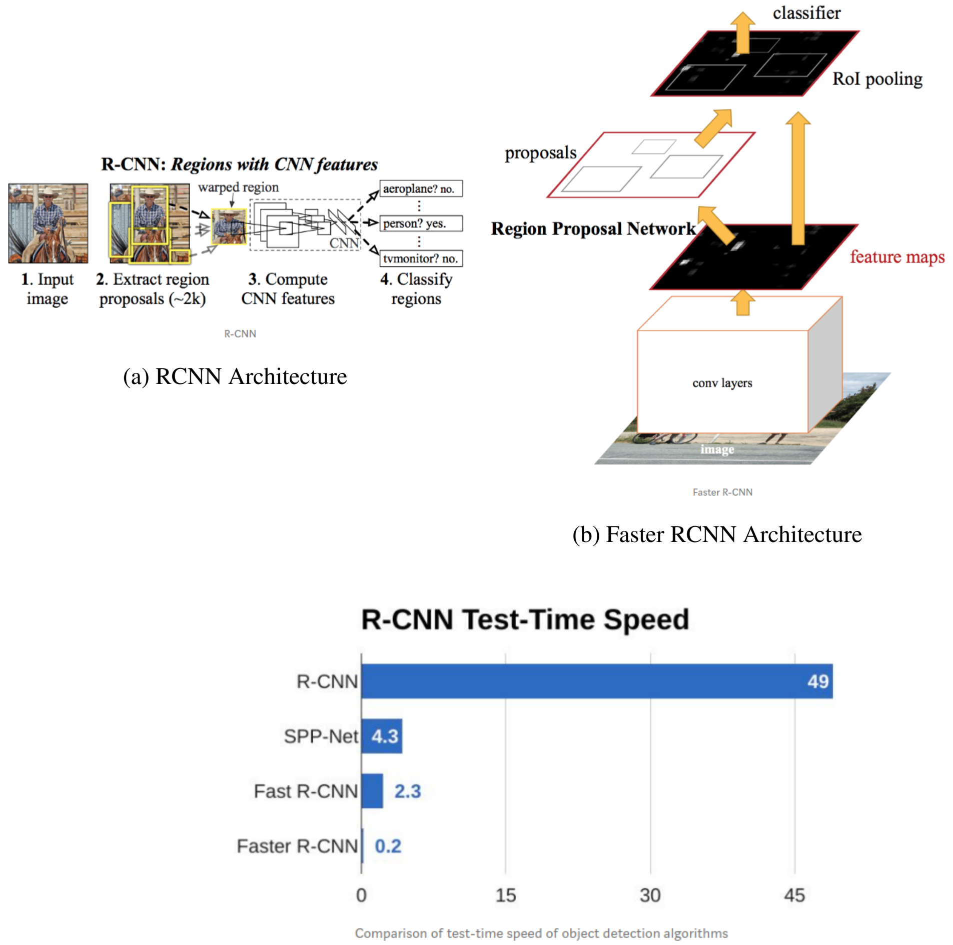
Towards the 'non-is' relationship detection, we have trained all the triplets of the relationship and the distinct objects in the class. There are 287 such triplets and 90 distinct objects in the class which accounts to 377 unique detection. Further, while creating the data-set for training, I find the minimum and maximum of the object bounding box of both the objects in a triplet to get the x and y co-ordinates of the triplet's bounding box. They will be x-min and y-min respectively. This new data-set can be used with the actual bounding boxes of the 90 objects to get the ground truth bounding boxes of all of the objects in the data-set and the triplets. Also, these

3 Model Architecture

3.1 Faster RCNN

Faster RCNN is a CNN model that is built on top of the RCNN network that is used to extract the region proposals from the image and after extracting the features using a backbone network which essentially work as a relevant feature extractor for making prediction for the region proposed. Faster RCNN is a faster version of the region proposal CNN architechture for object detection built on top of the RCNN network. Essentially, in faster RCNN we will use an alternate algorithm to selective search to find out the region proposals, which is in effect faster and optimal. This is by using another network to make the region proposals which was introduced by Shaoqing Ren et al.

In the RCNN network as in the figure above, they will compute the region proposals and use them to classify based on the features in the region proposal which will be used to compute the detection in



(a) RCNN Architecture

(b) Faster RCNN Architecture

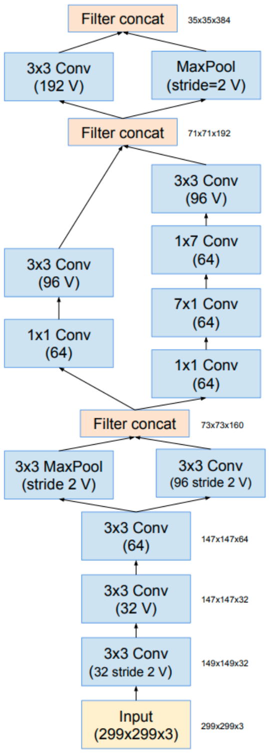
Figure 2: Comparison of Faster RCNN, Fast RCNN and RCNN network

those region proposals. This region proposal technique is slow and that's why in Faster RCNN we will use a network itself to create the region proposals from the extracted features. These features extracted from the region proposal network will be used on a classifier to detect which kind of the object it is. After the region proposal network, we reshape it using a RoI pooling layer which is then used to distinguish the proposed region and then to make prediction for the offset for the bounding box. In theory, Faster RCNN performs way faster than the RCNN CNN detector. The inference time is as follows. That was the motivation to pick the Faster RCNN network for detection task.

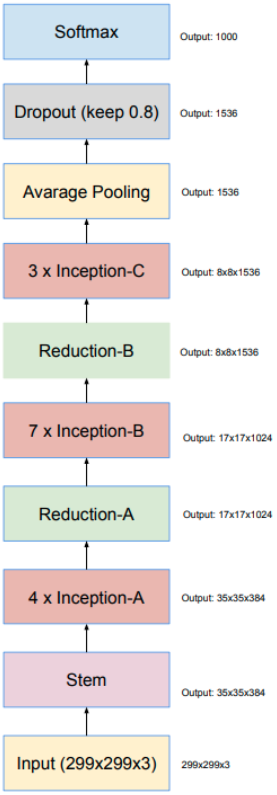
3.2 Inception V4 or Inception-Resnet Network

Inception V4 is a method that outperforms the in the classification of the objects as a CNN network. It utilizes the residual connection in the inception network which will accelerate the training process of the network by smoothing the gradient landscape significantly. This ensemble model will have ensemble of three residual and one Inception V4 network. The model also uses the Atrous convolution from the DeepLab CNNs which will facilitate less computation overhead while training the model by striding the convolution blocks. For the residual version of the inception network, the model uses a cheaper inception network with residual connection and each of these inception network is followed by filter expansion layer which is a 1x1 convolution without activation which is used for scaling up the dimensionality of the filter bank before the addition to match the depth of the input.

This modified version of the inception was able to create inference and learn the model much faster and get to the state of the art result.



(a) Pure Inception-v4 and Inception-ResNet-v2



(b) Inception V4 schema

3.3 Open Images Pre-trained Model

To get the best of the results, the pretrained model is used by utilizing the model zoo in tensorflow. The pretrained model with faster RCNN Inception Resnet network is trained on the open image data-set for object detection. The pretrained model in open images(version 4 and version 2 of the data-set) ensure that the detection will use the pretrained model weights and training time will be significantly reduced as the features are already trained for the same image. The checkpoints used for the training belongs to the Inception V4 based Faster RCNN model. The performance accuracy for detection task in the Open Image is as follows for the version 2 data-set and version 4 data-set of open images.

Model name	Speed (ms)	mAP@0.5[2]	Outputs
faster_rcnn_inception_resnet_v2_atrous_oidv2	727	37	Bounding boxes
faster_rcnn_inception_resnet_v2_atrous_oidv4	435	54	Bounding boxes

3.4 Ensemble Model

An ensemble model was used for the visual relationship track by combining both the 'is' and 'non-is' model. Both the models are faster RCNN models with Inception Resnet model that are trained in relationship triplets of 'is' and 'non-is' models. While ensembling the model, we will use the same image, predict the output for 'is' Faster RCNN model and then pass the same image variable to 'non-is' model. The first model will deliver results that are relevant to the 'is' relationships in the image and second model will deliver 'non-is' pairs in the image and we will append both the results to create the prediction for the particular image. The schematic model of the model is as follows.

4 Training

4.1 [1]Metrics

IoU and Mean IoU IoU, Intersection of the Union is the metrics used for training and testing the model. Mean IoU is the amount of intersection or overlap in the prediction and the actual data-set. The metric is used to calculate the precision at each instance if the IoU is greater than 0.50. When ever the IoU exceeds the threshold, it will be considered as a True Positive prediction or else as False Negative prediction. While training the FRCNN model we used Mean IoU as a metric to learn the bounding box regions.

mAP$_{rel}$- Average Precision on Relationship Detection For each relationship type (e.g. 'at', 'on') Average Precision (AP) is computed by extending the PASCAL VOC 2010 definition to relationship triplets. The main modification is that a matching criteria must apply on the two object boxes and three class labels (two object labels and a relationship label). We consider a detected triplet to be a True Positive (TP) if and only if both object boxes have IoU > threshold, (here threshold=0.50) with a previously undetected ground-truth annotation, and all three labels match their corresponding ground-truth labels. Any other detection is considered a False Positive (FP) in the two cases (1) both class labels of the objects are annotated in that image (regardless of positive or negative); or (2) one or both labels are annotated as negative. Finally, if either of the labels is unannotated, the detection is not evaluated (ignored). mAPrel is computed as the average of per-relationship APs.

Recall@N$_{rel}$ - Recall @ N=50 in the Relationship Detection The triplet detections are sorted by score and then the top N predictions are evaluated as TP, FP or ignored (see above). A recall point is scored if there is at least one True Positive is found among these top N detections.

mAP$_{phrase}$ - Mean Average Precision on Phrase Detection Each relationship detection triplet is transformed so that a single enclosing bounding box is formed from the two object detections. This bounding box has three labels attached (two object labels and one relationship label). The enclosing box is considered to be a TP if IoU > threshold with a previously undetected ground-truth annotation and all three labels match their corresponding ground-truth labels. The AP for each relationship type is computed according to the PASCAL VOC 2010 definition. mAPphrase is computed as the average of per-relationship APs.

Final Score The final score of the model is calculated using a weighted average of the all 3 above mentioned metrics. ie, we will use the following weights to each of the metrics in-order to calculate the score. Weights are $[0.4, 0.2, 0.4]$ respectively for mAP_rel, Recall@50_rel and mAP_phrase respectively.

5 Results

We are trying to create an ensemble model for 'is' and 'non-is'. As mentioned, the 'is' model and 'non-is' has been trained using a Faster RCNN with inception Resnet architecture on the triplet relationship as mentioned. In the is, as anticipated, the model 'is' was able to learn better and in the 'non-is', the performance was poor. It is because that the model for 'is' has less number of classes(42 opposed to 377) and the model has a clear objective as the subject and the attribute which is related by 'is' are in the same bounding box.

5.1 IS Model

The performance in 'is' model was better when compared to the 'non-is' detection task. The result image samples are as follows. These are the samples of correct predictions in difficult tasks like where there is a clutter of objects that has to be identified. The prediction w.r.t object attributes that causes a texture in the image are predicted accurately when compared to attributes like transparent and plastic. This is mostly because of the fact that the 'chair is wooden' class has the texture of wood

[1] From the Open Image Dataset V4 website. The metrics calculation script is no longer working. The prediction of is in CSV format is available with submission

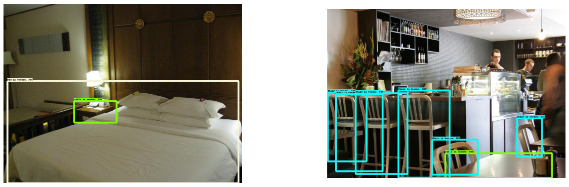


Figure 4: Correct Predictions in is

in brown while plastic can be blue, yellow, white or can have any texture. So the number samples to generalize the model for a plastic bottle will be really hard.

These results are mostly the confused output samples for the detection samples. In this, we have the model with with wrong detection bounding boxes or multiple bounding boxes for the same object. In figure 5.a we can see that the redbull can is identified as a transparent bottle. Similarly, in figure 5.b and 5.c we can see that the bottle(transparent) and piano(wooden) is being depicted with multiple bounding boxes.

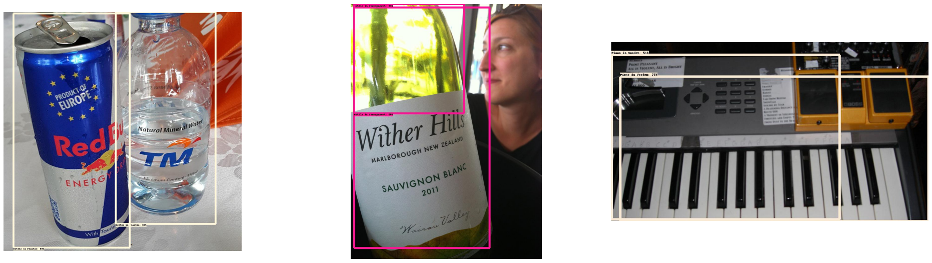


Figure 5: Confused prediction samples

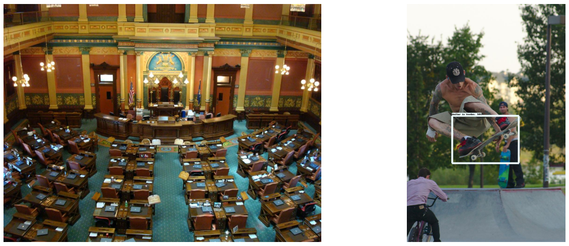


Figure 6: Wrong Prediction

These samples are the image samples in which the model failed. Figure 6.a has many tables and chairs, however from a different angle when compared to all the training image sets. The model is biased towards an angled image set of tables and chairs from sideways which is because it is giving no predictions at the given threshold value. In figure 6.b we have the skateboard wrongly predicted as a guitar which can be attributed to the wooden finish of both skateboard and guitar with presence of a person near to it. The metric results are as follows for 'is' model.

Mean IoU	Mean Average Precision(mAP@0.5)	Mean Average Phrase	Recall in top 50
0.26	0.40	0.40	0.25

5.2 NON-IS Model

There are 287 triplets of relationship pairs in the 'non-is' relationships in the training data-set. The data in the 'non-is' triplets are unbalanced and there are

The following triplets are samples where the model did not performed really well. The triplets pairs which had less number of samples in the training data-set as given in figure 7.a and 7.b. The number of samples for the given training set for the 'man near table' and 'man interacts with horse/dog' were quiet frequent in the training samples. Hence the model has good detection response towards these images.

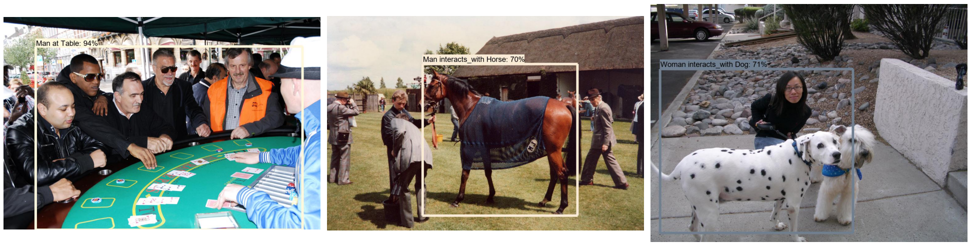


Figure 7: Confused prediction samples

However, in many image samples in the validation, the model was not even able to perform any close. There are many flawed and undetected images in the validation set. In the following figure 8.b, we can see that the model is over-fitted to the 'man plays guitar' while the actual description should not be that. Figure 8.a also demonstrates same over-fitting on the unbalanced data-set. There has been only around 500 valid predictions that the model could make in the 1537.

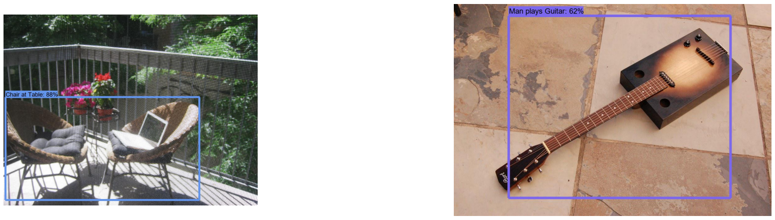


Figure 8: Wrong Prediction

The performance of the 'non-is' model has been very poor than expected and the model over-fitted to the highly frequent images in the data-set. This reduced the confidence in the prediction in the validation data-set.

6 Further Improvements

There are a lot of improvements or possible reserach direction for this particular relationship data-set that can be implemented on top of the given ensemble model as described in the model architecture. The model can be improved further especially in the poorly performing 'non-is' model. The 'non-is' triplet data-set has a unequal data distribution, ie, out of 287 triplets in the training sample, 88 triplets has less than 10 samples per triplet and 28 triplets has more than 1000 samples per triplet. This is the reason for the poor performance in the 'non-is' category. In order to achieve better performance we will take the first 100 triplets with most frequent occurrence and these data triplets can be used train a XGBoost or LightGBM model. To train the gradient boost model, we can use the following features to train the triplet pair.

1. Intersection of union of object1 and object2
2. % of object1 in object2 and % of object2 in object1.
3. Horizontal and vertical offset of centers of object1 and object2
4. Euclidean distance between the centers

7 Conclusion

The visual relationship nevertheless a hard problem in computer vision even if we are using deep learning techniques and huge computations. This is to an extent attributed to the fact that the when the data-set is comparatively small while we consider the number of data per triplet relationship for around 329 triplets of relationship in the Open Image data-set, we have a huge gap in the frequency of triplets with only 89 classes having 100 or more image samples per class and more than half of the triplets(around 160) do not have at least 50 image samples per triplet. Also, there are 88 triplets having less than 10 images per triplet relationship. This huge gap in the data-set distribution affects the model's training significantly. We are able to prove that the model was able to learn the aspects of 'is' by training the triplet relationship. However for 'non-is' the Faster RCNN model failed because there were many number of distinct triplets and the model was not able to learn well as the data samples per triplet was way too less for the learning when compared. The possible research direction will be to improvise the model for 'non-is' using the XGBoost technique described in the paper. The XGBoost on the first 100 high frequent relationship triplet can improve the model's performance. For the rest of the triplet pairs, another XG-boost algorithm can be used. Thus we can improve the current performance of the model.

References

[1] Kaggle discussion : https://www.kaggle.com/c/google-ai-open-images-visual-relationship-track/discussion/64642

[2] Kaggle discussion : https://www.kaggle.com/c/google-ai-open-images-visual-relationship-track/discussion/64630latest-584276

[3] Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun; "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks" The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 10364-10374

[4] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, Alex Alemi "Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning", The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016

[5] Ross Girshick, Jeff Donahue, Trevor Darrell, Jitendra Malik. "Rich feature hierarchies for accurate object detection and semantic segmentation". Computer Vision and Pattern Recognition (cs.CV),2014

[6] https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/detection_model_zoo.md

Appendix

All relevant codes used for this experiment to reproduce the result is attached as zip file.