Batch construction and multitask learning in visual relationship recognition

Shane Josias
Applied Mathematics & CAIR
Stellenbosch University
Stellenbosch, South Africa
josias@sun.ac.za

Willie Brink
Applied Mathematics
Stellenbosch University
Stellenbosch, South Africa
wbrink@sun.ac.za

***Abstract*—An image can be described by the objects within it, as well as interactions between those objects. A pair of object labels together with an interaction label is known as a visual relationship, and is represented as a triplet of the form (subject, predicate, object). Recognising visual relationships in a given image is a challenging task, owing to the combinatorially large number of possible relationship triplets, which leads to an extreme classification problem, as well as a very long tail found typically in the distribution of those possible triplets. We investigate the effects of three strategies that could potentially address these issues. Firstly, instead of predicting the full triplet we opt to predict each element separately. Secondly, we investigate the use of shared network parameters to perform these separate predictions in a multitask setting. Thirdly, we consider a class-selective batch construction strategy to expose the network to more of the many rare classes during mini-batch training. Our experiments demonstrate that batch construction can improve performance on the long tail, possibly at the expense of accuracy on the small number of dominating classes. We also find that a multitask model neither improves nor impedes performance in any significant way, but that its smaller size may be beneficial.**

***Index Terms*—visual relationship recognition, batch construction, multitask learning**

I. Introduction

There exists a variety of effective methods for locating and labelling objects in an image [1], [2]. A subsequent task in the image understanding pipeline could be to label the interaction or relationship between different objects. A visual relationship is defined as a triplet of the form (subject, predicate, object) that describes some visible interaction between a pair of objects in an image. The image in Figure 1, for example, contains the visual relationship (`boy`, `on top of`, `surfboard`). Such visual relationships can be used to construct a scene graph representation of an image [3], for further visual reasoning in tasks such as image retrieval, visual question answering, and automated surveillance.

Visual relationship recognition is the problem of producing (subject, predicate, object) triplets for a given image. It is often coupled with object localisation, but the focus of this paper is on the labelling task and we will therefore assume knowledge of tight bounding boxes around pairs of objects. Bounding boxes around objects can be generated by an off-the-shelf object detector (e.g. [1]) and merged pairwise in a straightforward manner.

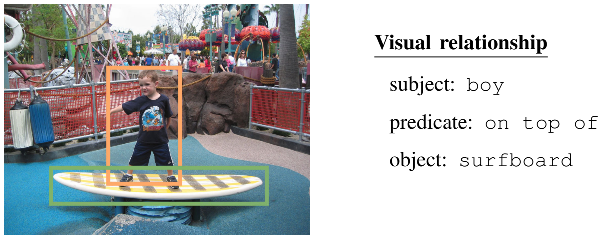


Fig. 1. An example of visual relationship recognition. The task is to label the subject, predicate (relationship) and object, given an image and bounding boxes around a pair of objects.

Visual relationship recognition is challenging for a number of reasons. Firstly, the number of possible relationships explodes combinatorially and leads to what is known as an extreme multiclass classification problem. For example, 100 possible subject and object labels, and 70 possible predicates, amount to 700,000 possible triplets. Secondly, the distribution of visual relationships in a dataset would typically exhibit a long tail: the vast majority of possible triplets might occur only a few times (or never) in the training set, while a small number might be frequent. Thirdly, predicates tend to be slightly more abstract than the subjects and objects, making their visual representations more difficult to model and recognise.

We investigate a number of strategies to deal with these issues. To address the combinatorially large set of possible classes, we design models that separately predict the elements of a triplet, instead of a single prediction of the complete triplet. This strategy allows for a multitask design where the different elements can be predicted with shared model parameters, potentially resulting in inductive transfer and statistical data amplification [4] for improved generalisation. For model training we also implement selective mini-batch construction through a type of training data distribution search, in an effort to better capture the long tail of the distribution over visual relationships. We compare the performance of our batch construction strategy against standard uniformly random batch

We thank the CSIR/SU Centre for Artificial Intelligence Research (CAIR) for financial support.

sampling, and also of our multitask model against multiple single-task models. The contributions of this work can be summarised as follows:

1) we show that batch construction is useful as a simple strategy for improved performance on underrepresented relationships (the long tail of the distribution);
2) we demonstrate that multitask learning is effective at reducing model complexity, without a significant positive or negative impact on performance.

II. Related work

The literature on visual relationship recognition can be grouped broadly into three common approaches. The first involves the learning of a visual-semantic embedding space. This can be achieved by imposing criteria such as small distances between similar relationships [5], or by modelling a relationship as vector translation between embedded objects [6], or by minimising a triplet-softmax loss [7]. Visual-semantic embedding allows for few- and zero-shot learning, and could therefore be suited for modelling a long-tailed distribution, but a separate classifier would still need to be trained on top of the embedding.

The second common approach attempts to generate the scene graph, or collection of interconnected relationships, directly. Xu et al. [8] perform graph inference with a structural recurrent neural network and an iterative message passing scheme to refine its predictions. Zellers et al. [9] observe that natural images usually have certain kinds of structural regularities, which they dub motifs, and propose stacked neural networks ("MotifNets") to predict graph elements and an LSTM to encode global context. Further examples of this approach include the use of associative embeddings [10], graph parsing neural networks [11], and graph R-CNN [12]. Woo et al. [13] improve on graph generation strategies by designing an explicit relational reasoning module. Generating a scene graph is more direct than the visual-semantic embedding approach, and end-to-end training to accomplish the intended task directly can lead to superior performance.

The third approach, and the one most relevant to our work, treats the prediction of each element of the relationship triplet as its own classification task. Some works use multi-stream architectures for each task [14]–[17], while others employ a single multitask scheme [18], [19] similar to what we will investigate.

There seems to be a central theme of transferring knowledge for improved performance, through message passing, global context cues, or inductive transfer in multitask learning. The multi-stream and multitask settings can deal with the huge number of classes in visual relationship recognition, by making use of multiple outputs of smaller dimensions. It does remain unclear whether multitask learning could necessarily provide better performance. Existing approaches also tend to build very large systems, with many parameters, and it is usually not clear exactly how the long tail of typical datasets are dealt with. We have not yet come across approaches dealing with data distribution searches during training.

III. Our approach

We want to train a model that takes an image as input, cropped around a pair of objects, and outputs a (subject, predicate, object) triplet. Training labels are used to define fixed vocabularies for each element. We may therefore treat visual relationship recognition as classification, and models will be set up to output normalised class scores over triplets. It may be noted that subjects and objects usually share a vocabulary, but it is not a strict requirement.

Instead of attempting to train a convolutional neural network to output one massive vector of scores over all possible triplets, we consider three separate tasks: predicting the subject label, predicting the predicate label, and predicting the object label. Each of these tasks has far fewer possible classes, and by making the simplifying assumption that the tasks are conditionally independent given an image, we may combine their normalised output scores through multiplication. The top scoring triplet can then be obtained by combining the top scoring elements from each of the three tasks.

We note that typical datasets for visual relationship recognition exhibit a long tail not only in the distribution over all triplets, but also in each of the marginal distributions over subjects, predicates and objects. An example of this behaviour follows in section IV-A.

A. Single-task learning with standard batching

We create three separate neural network models to predict the subject, the predicate and the object from the same image. Each network consists of the convolutional block of a pretrained network (we chose ResNet-18 [20] for its good balance between size and performance), followed by three trainable, 2,048-dimensional fully-connected layers and a softmax output layer. Refer to Figure 2.

In order to train each model we minimise a cross-entropy loss function with mini-batch gradient descent [21]. For each training iteration a mini-batch of some prespecified size is sampled without replacement, uniformly across all samples in the training set. For visual relationship recognition where the data is often heavily skewed, this "standard" approach to batch selection is likely to pick samples mostly from a small number of frequently occurring classes. The network may thus learn these dominant classes very well, but would be unable to recognise the vast majority of classes in the long tail of the data distribution.

B. Class-selective batch construction

In an effort to mitigate the potential problem with standard batching mentioned above, and expose the network to more classes in the tail of the dataset, we implement the following batch construction strategy. For a particular task (which can be to predict the subject, or to predict the predicate, or to predict the object), we sample at every training iteration n classes from the vocabulary of that task, uniformly at random. We then randomly select m samples from each of those n classes, for a mini-batch of size mn.

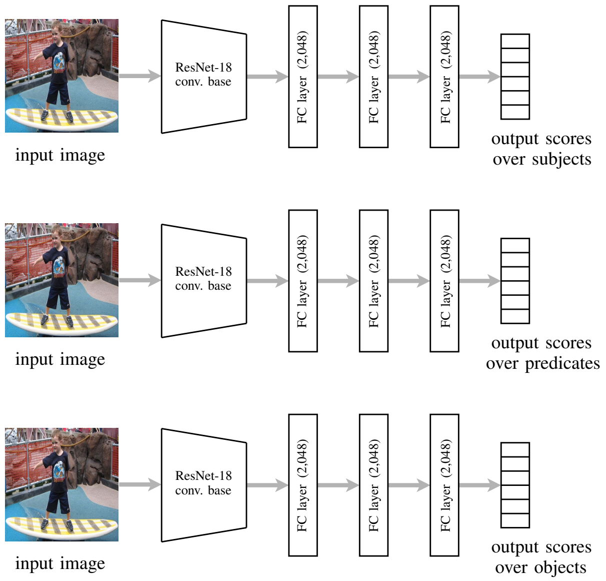


Fig. 2. For single-task learning we construct three separate models to predict respectively the subject, predicate and object from a given image crop. The trainable fully-connected layers all have 2,048 neurons and the output is a softmax over the classes of each task.

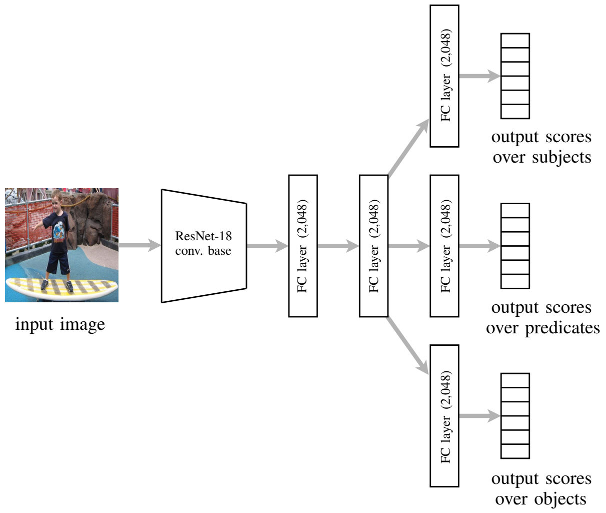


Fig. 3. For multitask learning we construct a single model to output three score vectors over the subject labels, predicate labels and object labels. The shared and separate fully-connected layers all have 2,048 neurons and the outputs all use softmax.

Constructing batches in this manner would allow the network to learn from all the classes in a particular task, in equal measure. We hypothesise that it can lead to better performance on the many rare classes in the long tail of the data, potentially at the expense of reduced performance on the small number of dominant classes. Of course, there is now a risk of biasing the network against the true distribution of the data and impede its ability to generalise properly. We investigate these issues experimentally in section IV.

C. Multitask learning

In addition to batch construction we also explore multitask learning, which can be thought of as an inductive form of transfer learning where knowledge is transferred across tasks. The premise is that it might lead to a more robust model, capable of generalising better [4].

In our case we may use a single network with multiple output vectors, instead of multiple networks each performing a single task. Specifically, we use the convolutional base of ResNet-18, add two trainable, 2,048-dimensional fully-connected layers, and then split the network to three parts. Each part has its own 2,048-dimensional layer and a softmax output over the subjects, predicates and objects, respectively. Figure 3 clarifies. The first two fully-connected layers are thus shared and might learn effectively from the three different tasks. The network is trained to minimise the sum of cross-entropy losses over the three output vectors, using mini-batch gradient descent.

In multitask learning it is generally common to define a main task together with auxiliary tasks which could be less important. For our visual relationship recognition model we may want to regard each of the three tasks equally important. However, when coupled with batch construction (as described in section III-B), we have to sample the m classes from a single task, at every training iteration, and then use the triplets from the complete labels of the training samples in the batch. In section IV we explore how performance of the multitask model changes depending on which task is used for batch construction.

D. Implementation

All models are implemented in the PyTorch framework [22]. For standard batching we use a batch size of 300. For class-selective batch construction we choose $m = 6$ and $n = 50$. There could a trade-off in performance between the number of classes and sampled instances per class, but informal experimentation showed no significant difference for our models (which might be somewhat surprising, although it could be that effects average out over multiple batches). Mini-batch gradient descent is performed using Adam with a learning rate scheduler. We make use of mixed precision training via NVIDIA Apex, to enable more optimal use of GPU memory. All training is performed on a single NVIDIA GeForce RTX 2070.

IV. Experimental results

A. Data

We use the VRD dataset of Lu et al. [5]. It contains 5,000 images, and a total of 37,987 relationship instances (triplets) that we split into a training set and a test set (80:20). More specifically, in an effort to ensure representativeness in both sets, we consider each class i of predicates and split the subset of triplets that has i as a predicate into 80% training data and 20% test data. We chose to base the split on the predicate, since it has fewer samples per class in the tail.

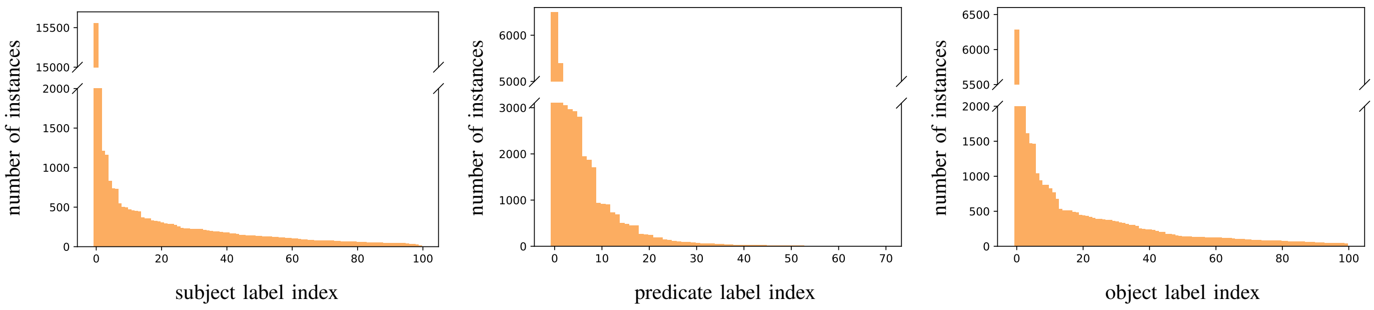


Fig. 4. Plots of the number of relationship instances containing each subject label, predicate label and object label, across the entire VRD dataset [5].

Each predicate is an action verb (e.g. `kick`), a non-action verb (e.g. `wear`), a spatial relationship (e.g. `on top of`), a preposition (e.g. `with`), or comparative (e.g. `taller than`).

There are 100 labels shared between subjects and objects, and 70 labels for predicates, for a total of 700,000 possible (subject, predicate, object) triplet labels. We note that our training set contains only 15,448 unique triplets. However, the manner in which the models are set up to output subject, predicate and object separately, potentially enables the recognition of triplets never seen during training.

The long-tailed nature alluded to throughout this paper exists in this dataset not only at the relationship triplet level, but also at the level of subjects, predicates and objects, as shown in Figure 4.

B. Evaluation metrics

We evaluate the performance of the various models first in terms of predicting each of the three elements of a visual relationship, then in terms of predicting the triplet as a whole.

A standard metric for visual relationship recognition is the recall at k, abbreviated as R@k and sometimes called the top-k accuracy, which measures the percentage of times the correct label occurs in the top k predictions (if ordered by output scores). For the tasks of predicting individual elements, i.e. looking only at the output over subjects, or over predicates or over objects, we measure R@1 and R@3 on the test set. For the task of predicting the full (subject, predicate, object) triplet, we measure R@50 and R@100 which seem to be standard practice for a label set of this size [5], [6], [8], [10]. Keep in mind that there are 700,000 possible triplets that can be predicted. We found that a random classifier yields an R@100 score of approximately 0.026% on the skewed test set.

In order to evaluate how effectively each model deals with the many rare classes in the tail of the data distribution, we also measure the mean per-class accuracy, abbreviated as MPCA, over the test set. This metric effectively ignores class imbalance. Note that we use this metric only to evaluate the prediction of single elements (subjects, predicates or objects), and not the prediction of full triplets. The large number of possible triplets, and the fact that relatively few of them appear in the test set, make MPCA less informative in this setting.

For an indication of how the models fare on only the rare classes, we construct a subset of the test set by keeping only those triplets for which the subject, predicate and object each have fewer than 1,000 instances across the full dataset (refer to Figure 4). We use counts over the full dataset merely as a proxy for rarity, and remind the reader that elements in the training set are distributed similarly to those in the full set.

C. Quantitative evaluation

Results from the various models are presented in Table I, for the three tasks of predicting subject, predicate and object over all the samples in the test set.

Based on the MPCA metric we may note that class-selective batch construction improves performance on the long tail-end of each individual task, but only if batches are constructed through the sampling of classes from the same task. Lower accuracies from all models for the prediction of predicates verify our suspicion that predicates are harder to recognise visually. Furthermore, we note that multitask learning does not seem to significantly improve or worsen mean per-class accuracy on the prediction of individual elements. Generally speaking, it is not yet clear under which circumstances a multitask model will improve performance, but there are arguments suggesting that more uniform label distributions in the auxiliary tasks might be preferred for multitask learning to be effective [23]. In our case, the multitask models do provide similar performance to the multiple single-task models, which is useful if limitations on model size and complexity are important. Reduced model capacity can also act as a form of regularisation.

The results in Table I also indicate higher R@1 and R@3 scores for models trained with standard batching compared to those that implement batch construction. There seems to be a trade-off: batch construction contributes to better generalisation on the many rare classes, at a cost of lower accuracy on the small number of dominant classes.

Table II shows evaluation results from the different models predicting full (subject, predicate, object) triplets. We report on R@50 and R@100, as is standard in the literature, and remind the reader that there are 700,000 possible classes in this case. Our obtained results are quite similar to related work, but since we focus only on the labelling of visual relationships, and not also the localisation of individual objects, a direct comparison would not mean much.

TABLE I
QUANTITATIVE TEST RESULTS FROM VARIOUS VERSIONS OF OUR MODELS, ON PREDICTING SINGLE ELEMENTS OF VISUAL RELATIONSHIPS.

Model	Description	Predicting the subject			Predicting the predicate			Predicting the object		
		MPCA	R@1	R@3	MPCA	R@1	R@3	MPCA	R@1	R@3
ST-SB	single-task, standard batching	19.09	53.29	73.63	4.51	31.96	56.77	28.92	40.23	68.17
ST-BC-S	single-task, batch construction from subject labels	33.13	16.14	39.39	4.13	19.26	41.36	22.44	38.20	63.74
ST-BC-P	single-task, batch construction from predicate labels	16.70	49.55	68.86	17.01	10.78	31.72	25.20	34.99	61.50
ST-BC-O	single-task, batch construction from object labels	16.67	52.66	71.43	5.24	27.93	51.39	40.72	26.62	50.58
MT-SB	multitask, standard batching	19.96	53.44	74.62	4.74	32.24	57.12	28.34	40.03	68.94
MT-BC-S	multitask, batch construction from subject labels	32.83	17.37	43.41	4.09	19.35	40.70	22.46	38.46	64.92
MT-BC-P	multitask, batch construction from predicate labels	17.18	50.26	70.95	17.54	12.71	32.39	26.24	35.59	62.65
MT-BC-O	multitask, batch construction from object labels	17.52	53.05	72.08	6.27	28.34	52.06	40.60	27.33	51.91

TABLE II
QUANTITATIVE TEST RESULTS FROM THE MODELS, ON PREDICTING FULL VISUAL RELATIONSHIP TRIPLETS.

Model	Predicting the full triplet			
	R@50	R@100	Tail R@50	Tail R@100
ST-SB	49.18	58.18	13.10	17.74
ST-BC-S	23.87	30.84	20.96	27.82
ST-BC-P	31.79	42.10	16.93	23.58
ST-BC-O	40.66	48.58	18.95	24.59
MT-SB	50.27	59.69	12.50	18.95
MT-BC-S	24.95	32.37	19.35	27.21
MT-BC-P	33.56	44.08	17.94	26.41
MT-BC-O	41.83	49.47	20.76	26.20

As before, standard batching produces better recall at 50 and 100 compared to batch construction. However, when focusing only on the long tail of the distribution (as explained at the end of section IV-B), we find that batch construction does offer an improvement.

One may postulate that the predicate is most representative of the visual relationship, but it appears from our experiments that batch construction with the object labels is a better strategy. The ResNet-18 layers might have an influence here, since they have been pretrained for object classification and thus potentially less suited for the more abstract concept of a predicate.

A fundamental difference between the single- and multitask setting is that the latter receives a training signal from every element of the visual relationship triplet simultaneously. In some sense it sees the full visual relationship, yet does not yield significantly better scores compared to the single-task setting. Again, this shows that it is not immediately obvious that multitask learning will give improved metrics, and corroborates previous findings [23].

A significant drawback of these quantitative evaluations is that they compare model predictions to a particular ground truth label, despite the fact that visual relationships are often ambiguous and a prediction different from the ground truth might not be completely wrong. We explore this briefly in the next section.

D. Qualitative evaluation

Table III shows a number of test image samples and the top five predicted triplets from four of the models. Here we choose to highlight batch construction based on the object labels, since it performed best overall in the quantitative evaluations.

For the second example shown in the table, (`giraffe`, `taller than`, `giraffe`), ST-BC-O correctly predicts the subject and object but predicts `in front of` as the predicate; perhaps a forgivable error. Similarly sensible errors can be seen throughout the examples, and demonstrate a level of ambiguity often present in visual relationship labels. The predicates `behind` and `in front of` do appear with very different confidence scores. This is undesirable behaviour that a human would not display, and might motivate the inclusion of multi-modal semantics in the modelling process.

We note that `person` is the dominating subject class, and is predicted correctly in almost all cases shown. In the third example of Table III models favour predicates other than `on`, despite it being the dominating predicate class. This may be due to the strong visual cues in favour of interactions between the person and their items of clothing, rather than the slightly more obscure skateboard.

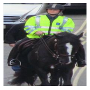
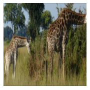
Models trained with batch construction appear to make predictions with relatively high confidence scores. There are a total of 700,000 normalised confidence scores, so high scores in the top five predictions mean exceptionally low scores for the remaining 699,995 relationships. It is interesting that under an arguably more uniform training data distribution, the confidence scores are this heavily skewed.

The ground truth predicate of the last example in Table III, namely `feed`, is a rare tail-end predicate and is misclassified even under our batch construction strategy. The predicted predicates do seem sensible with regards to the visual cues in this example, and further motivate an investigation into semantic modelling.

V. CONCLUSION

We investigated the potential of class-selective batch construction and multitask learning for the task of visual relationship recognition. It is a challenging task in computer vision, given the large number of possible relationships as well as a typical long-tail distribution over those relationships. We saw that our batch construction strategy does improve performance on the tail of the distribution, but at the cost of performance on the small number of dominating classes at the head of the distribution. Multitask learning does not seem to

TABLE III
QUALITATIVE RESULTS FOR A FEW TEST IMAGES.

Test image	Top 5 triplet predictions and confidence scores							
	ST-SB		ST-BC-O		MT-SB		MT-BC-O	
(a) 	person, on, horse	12.0	person, on, horse	18.7	person, wear, horse	9.3	person, on, horse	13.2
	person, ride, horse	7.0	person, has, horse	11.8	person, on, horse	6.8	person, above, horse	12.0
	person, wear, horse	5.3	person, wear, horse	7.7	person, wear, person	3.4	person, behind, horse	6.3
	person, has, horse	5.2	person, in front of, horse	4.3	person, behind, horse	3.1	person, ride, horse	5.3
	person, on, person	3.1	person, next to, person	3.7	person, has, horse	2.6	person, has, horse	4.8
(b) 	giraffe, taller than, giraffe	25.1	giraffe, in front of, giraffe	98.6	giraffe, taller than, giraffe	45.4	giraffe, in front of, giraffe	92.5
	giraffe, in front of, giraffe	20.8	giraffe, taller than, giraffe	0.4	giraffe, in front of, giraffe	18.9	giraffe, taller than, giraffe	6.0
	giraffe, next to, giraffe	9.5	giraffe, behind, giraffe	0.4	giraffe, next to, giraffe	8.6	giraffe, behind, giraffe	0.9
	giraffe, above, giraffe	7.6	giraffe, next to, giraffe	0.1	giraffe, behind, giraffe	7.3	giraffe, next to, giraffe	0.3
	giraffe, behind, giraffe	7.2	giraffe, beside, giraffe	0.1	giraffe, under, giraffe	2.6	giraffe, beside, giraffe	0.07
(c) 	person, wear, person	11.8	person, wear, skateboard	25.6	person, wear, shirt	15.5	person, wear, skateboard	20.0
	person, wear, shirt	10.5	person, on, skateboard	10.0	person, wear, person	9.6	person, wear, shoes	14.0
	person, wear, skateboard	10.0	person, has, skateboard	9.6	person, wear, skateboard	6.9	person, wear, helmet	12.0
	person, wear, shoes	5.4	person, ride, skateboard	5.2	person, wear, shoes	6.1	person, has, skateboard	3.8
	person, wear, pants	4.4	person, wear, shoes	3.5	person, wear, pants	4.1	person, wear, pants	3.7
(d) 	person, above, street	4.3	person, under, elephant	16.4	person, on, street	4.7	person, in front of, elephant	7.4
	person, on, street	4.1	person, in front of, elephant	16.0	person, under, street	3.9	person, near, elephant	6.9
	person, under, street	3.0	person, above, elephant	10.0	person, above, street	3.4	person, under, elephant	5.1
	sky, above, street	1.7	person, near, elephant	4.7	person, on, person	2.4	person, on, elephant	3.4
	sky, on, street	1.6	person, behind, elephant	4.1	person, under, person	1.9	person, above, elephant	2.4

The ground truth relationships for these test images are (a) person, on, horse; (b) giraffe, taller than, giraffe; (c) person, on, skateboard; (d) person, feed, elephant.

improve or impede performance when compared to the single-task learning, but provides other benefits such as a reduced model capacity. We also saw that it is more difficult to model and recognise the predicate of a relationship, and suggest that current pretrained models might not be suitable for that task. Finally we demonstrated through a few examples that some misclassifications are semantically similar to the ground truth labels, suggesting that the incorporation of a language model may be useful.

REFERENCES

[1] Redmon J, Divvala S, Girshick R, Farhadi A. You only look once: unified, real-time object detection. In: IEEE Conference on Computer Vision and Pattern Recognition; 2016. p. 779–788.

[2] Zhang S, Wen L, Bian X, Lei Z, Li S. Single-shot refinement neural network for object detection. In: IEEE Conference on Computer Vision and Pattern Recognition; 2018. p. 4203–4212.

[3] Johnson J, Krishna R, Stark M, Li LJ, Shamma D, Bernstein M, et al. Image retrieval using scene graphs. In: IEEE Conference on Computer Vision and Pattern Recognition; 2015. p. 3668–3678.

[4] Caruana R. Multitask learning. Machine Language. 1997;28(1):41–75.

[5] Lu C, Krishna R, Bernstein M, Fei-Fei L. Visual relationship detection with language priors. In: European Conference on Computer Vision; 2016. p. 852–869.

[6] Zhang H, Kyaw Z, Chang SF, Chua TS. Visual translation embedding network for visual relation detection. In: IEEE Conference on Computer Vision and Pattern Recognition; 2017. p. 5532–5540.

[7] Zhang J, Kalantidis Y, Rohrbach M, Paluri M, Elgammal A, Elhoseiny M. Large-scale visual relationship understanding. Computing Research Repository. 2018;arXiv:1804.10660.

[8] Xu D, Zhu Y, Choy CB, Fei-Fei L. Scene graph generation by iterative message passing. In: IEEE Conference on Computer Vision and Pattern Recognition; 2017. p. 5410–5419.

[9] Zellers R, Yatskar M, Thomson S, Choi Y. Neural motifs: scene graph parsing with global context. In: IEEE Conference on Computer Vision and Pattern Recognition; 2018. p. 5831–5840.

[10] Newell A, Deng J. Pixels to graphs by associative embedding. In: Advances in Neural Information Processing Systems; 2017. p. 2171–2180.

[11] Qi S, Wang W, Jia B, Shen J, Zhu SC. Learning human-object interactions by graph parsing neural networks. In: European Conference on Computer Vision; 2018. p. 401–417.

[12] Yang J, Lu J, Lee S, Batra D, Parikh D. Graph R-CNN for scene graph generation. In: European Conference on Computer Vision; 2018. p. 670–685.

[13] Woo S, Kim D, Cho D, Kweon IS. LinkNet: relational embedding for scene graph. In: Advances in Neural Information Processing Systems; 2018. p. 560–570.

[14] Dai B, Zhang Y, Lin D. Detecting visual relationships with deep relational networks. In: IEEE Conference on Computer Vision and Pattern Recognition; 2017. p. 3076–3086.

[15] Wang G, Luo P, Lin L, Wang X. Learning object interactions and descriptions for semantic image segmentation. In: IEEE Conference on Computer Vision and Pattern Recognition; 2017. p. 5859–5867.

[16] Chao YW, Liu Y, Liu X, Zeng H, Deng J. Learning to detect human-object interactions. In: IEEE Winter Conference on Applications of Computer Vision; 2018. p. 381–389.

[17] Yin G, Sheng L, Liu B, Yu N, Wang X, Shao J, et al. Zoom-Net: mining deep feature interactions for visual relationship recognition. In: European Conference on Computer Vision; 2018. p. 322–338.

[18] Gkioxari G, Girshick R, Dollár P, He K. Detecting and recognizing human-object interactions. In: IEEE Conference on Computer Vision and Pattern Recognition; 2018. p. 8359–8367.

[19] Li Y, Ouyang W, Wang X, Tang X. ViP-CNN: visual phrase guided convolutional neural network. In: IEEE Conference on Computer Vision and Pattern Recognition; 2017. p. 1347–1356.

[20] He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In: IEEE Conference on Computer Vision and Pattern Recognition; 2016. p. 770–778.

[21] Ruder S. An overview of gradient descent optimization algorithms. Computing Research Repository. 2016;arXiv:1609.04747.

[22] Paszke A, Gross S, Massa F, Lerer A, Bradbury J, Chanan G, et al. PyTorch: an imperative style, high-performance deep learning library. In: Advances in Neural Information Processing Systems; 2019. p. 8024–8035.

[23] Alonso HM, Plank B. When is multitask learning effective? Semantic sequence prediction under varying data conditions. Computing Research Repository. 2016;arXiv:1612.02251.