Automated Data Collection Using Spatial Relationship Detection in Videos

Martin Bartolo
Supervisor: Adrian Muscat
University of Malta
Msida, MSD 2080, Malta

ABSTRACT

While spatial relationship detection has been well researched in recent years, literature relating to its applications is scarce. We showcase a system that uses a multilabel spatial relationship classifier to automatically generate spatial relationship instances between objects-subject pairs detected in frames of videos in the wild. The system is built from a pre-trained open source [3] object detector and localizer (YOLOv4) and a multi-label spatial relationship model is defined and trained over the SpatialVOC2K [2] dataset. This system can be used as a tool to create new custom spatial relationship datasets or enrich existing ones using a collection of videos as an input. As a proof of concept, we use the system to generate additional instances for under-represented classes from the SpatialVOC2K dataset and retrain a classifier to study its efficacy in detecting these spatial relations. The system was successful in generating additional instances and its inaccuracies were identified and explained using a human evaluation to see how it could be improved in the future.

1. INTRODUCTION

A spatial relationship is a 3-tuple of the form $\langle subject - \textbf{relationship} - object \rangle$ that describes the way that a pair of objects are related in terms of their relative size, position and spatial distribution. An example would be $\langle person - \textbf{next to} - dog \rangle$ where person is the subject, *next to* is the relationship and dog is the object. While the ability to detect spatial relationships is a key step in a machine's ability to understand the world around it, much of its applications, though acknowledged [1], have remained unexplored [6]. Examples of such applications, particularly of the use of spatial relationship detection in videos, include real-time uses such as robotics in the healthcare and maintenance industries, autonomous vehicles and aids for the visually impaired. Another application, and one which we will be exploring in this project, is automated spatial relationship data collection.

Attempts to streamline the data collection procedure for object detection datasets in domains such as computer vision and human-machine interaction have already been made, an example of which being [13] where Sapp et al. used a green screen to collect images of objects in front of various simulated backgrounds to create large amounts of training instances from a small set of manually captured object images. Such research is relevant since the creation of these datasets is very costly and it is very difficult to obtain annotations which are consistent and of a high quality. This problem results in noisy datasets, many of which are limited in size due to the complexity of finding suitable instances to include. In our case, we will attempt to collect meaningful spatial relationship data from videos using a neural-network model trained on the SpatialVOC2k dataset [2].

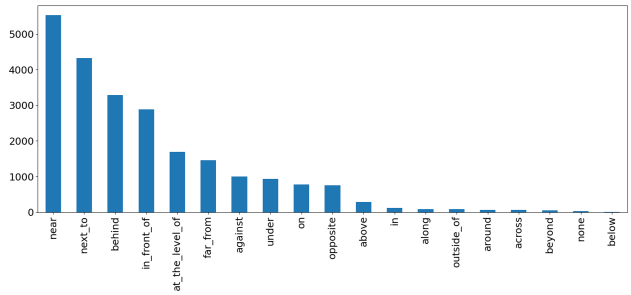

Figure 1: Number of instances of each SpatialVOC2k class

Automated data collection from videos is not a new concept. Interesting examples include the collection of football match analysis data [14] and the collection of traffic data [7] from video recordings. We develop a system that is able to go through reels and reels of videos, obtained from open-source stock video websites, and pick up predetermined relationships between specified object pairs. This is carried out in two steps. First, the system selects candidate solutions and gives a confidence score to each. Next, the system considers involving a human in the loop to accept or reject proposals with low confidence scores. When finding the candidate solutions, we must first detect the objects in each frame using a YOLOv4 object detector, chosen due to its speed and accuracy compared to alternative models [3]. Next, a neural-network model, which we train and test beforehand on the SpatialVOC2k dataset [2], is used to identify the spatial relationships between these objects, along with their confidence scores.

The SpatialVOC2k dataset [2] is itself a noisy dataset, limited in size due to its many under-represented classes (Figure 1). We hypothesise that if we add our system's high-confidence spatial relationship detections, particularly those which include the dataset's under-represented prepositions, we will obtain an enriched dataset which can be used to train a model with an improved ability to correctly identify these prepositions. In turn, the model will obtain more accurate overall results on future unseen instances.

2. SET-UP

2.1 SpatialVOC2k

This dataset contains 2,026 images, annotated with spatial relationships between objects in the 20 VOC [4] object classes. These relationships are split into 23 classes, whose names we translate from French into 19 English classes (the number reduces since multiple French classes share the same English meaning). The distribution of instances in each spatial relationship class, which we will call prepositions, is skewed with some classes having far more instances than others (Figure 1). Each SpatialVOC2K image instance, aside from its height and width, contains a set of spatial relationships between object pairs which we will be represent by 3-tuples of the form $\langle subject - \textbf{preposition} - object \rangle$ with the object and subject referred to as the object pair. Each such relationship contains information about the object/subject including its depth in the image, its class, its pose (i.e. direction that it is facing) and the coordinates of each corner of its bounding box. It also notes each relevant preposition (in French) that can be used to describe the spatial relationship between the object and the subject with the most suitable of these prepositions being highlighted. Object/subject depth and pose is not used in this project since we have no way of extracting this information from unlabelled frames in videos without the use of additional models. We use each object pair's classes and bounding box data, along with the list of their acceptable prepositions to generate a list of features which we use to train a model to identify spatial relationships between unseen object pairs.

2.2 Unlabelled Videos

This dataset is made up of 15 unlabelled videos collected from various open source stock footage websites. The videos depict objects from the 20 VOC [4] classes performing a variety of activities in real world environments and was created with the aim of capturing a wide variety of spatial relationships between objects from as many VOC classes as possible. When creating the dataset, tests were ran using the object detection model to make sure that a sufficient amount of objects were being detected in each video so that each video would be relevant and provided value to the dataset. We use this dataset to evaluate the performance of our spatial relationship model in the wild. This is done by using the model to label the spatial relationships between each pair of objects in each frame from the set of videos and then randomly selecting a number of predictions whose accuracy will be evaluated by a group of people who will state whether they agree or disagree with the predictions.

2.3 Object Detector

Before detecting the spatial relationships between object pairs, the objects themselves must be identified. Since we will be working with unseen videos, we require an object detector with a high degree of efficiency (in terms of frames that it can predict per second), while still maintaining high levels of accuracy. After evaluating a number of frameworks, the latest version of YOLO at the time of writing, YOLOv4 [3] was selected since it is the fastest open source object detection framework available and also has a higher average precision value than its counterparts.

The object detection model assigned a confidence score to each of its predictions ranging from 0 to 1 with 0 meaning that it had no confidence in its prediction and 1 meaning that it had full confidence. After carrying out an error analysis on a number of wild images, it was concluded that objects that received a confidence score of less than 0.5 should be ignored. These objects were often blurred and unclear, leading to bounding boxes that did not reflect the objects' true size or position. Many of these predictions also included incorrect object classes. We therefore only kept object detections that received a confidence score of 0.5 or above. Aside from this, objects were sometimes detected as being in two different classes, both with a confidence above the 0.5 threshold. To solve this problem, we caught cases where pairs of objects' bounding boxes overlapped by 95% and removed the object with the lower confidence score from the prediction. After this process, features could be extracted from the newly obtained object class and bounding box data to use in the training of the spatial relationship model.

2.4 Features

From the object detections, we generated 39 geometrical features and a number of language features. Each of the geometrical features was calculated using the object pair's bounding boxes and stored information describing the relative sizes and positions of the objects. A description of each geometrical feature is given by [5]. A subset of geometrical features was made by using a selection of some of the features from the full geometric feature set with the intention of maximising the model's performance [10]. The features were initially chosen using a greedy backward feature elimination procedure as discussed in [10] further streamlining of the list using the correlation between features described in [1]. In the evaluation stage, the feature importance ranking in [5] was used to test out whether different geometrical feature subsets would improve the model's performance on specific prepositions.

Three types of language features were collected from the object pair's class labels. The purpose of these features was to add context to the geometrical features by giving information about the objects themselves. This would allow the model to learn about the way that certain objects interact with others in specific ways (for example: $\langle car - \textbf{on} - person \rangle$ is unlikely but $\langle person - \textbf{on} - bicycle \rangle$ is likely). The first type was a one-hot encoded vector. Each object was encoded by placing a 1 in the index representing the object's class and 0s in all other indices. Each object's vector was 20-dimensional due to the 20 VOC classes that were marked with 1 or 0. In the second type, the object classes were converted into pre-trained word embeddings using GloVe [12] which represents each class as a 50-dimensional vector of real numbers. In this way, the model would be able to achieve a better understanding of the objects since any similarities between classes would be represented between these word embeddings (for example: the Euclidean distance between the sheep and cow word embedding vectors would be closer than the Euclidean distance between the person and car word embedding vectors). The third type of language features used pre-trained Word2Vec [9] word embeddings which represent each object class as a 300-dimensional vector of real numbers.

Different feature lists were made and tested using different combinations of the full/partial geometrical features with the word embeddings/one-hot encoded language features so that the best one could be used to train the final model.

2.5 Training the Spatial Relationship Model

The next step was to train a multi-label classifier which would assign a confidence score to each preposition according to how well it described the spatial relationship between a pair of objects. In order to train the best possible model for this task, different combinations of features were used to train eight different models. Each of these models was tested on the SpatialVOC2K dataset so that the best one would be kept as the final model.

Model	Full Geo	Partial Geo	One-hot	GloVe	Word2Vec
1	✓	-	-	-	-
2	✓	-	✓	-	-
3	✓	-	-	✓	-
4	✓	-	-	-	✓
5	-	✓	-	-	-
6	-	✓	✓	-	-
7	-	✓	-	✓	-
8	-	✓	-	-	✓

Table 1: Features used to train each model

When training each model, the corresponding list of features was composed and split into a training set, a validation set and a test set. These splits were stratified to ensure that there were instances of each preposition present in each set. Multiple splits were tested out until it was concluded that the most effective split would be 80% training and 20% test. The training/validation split was also 80%/20%. The full list of features was then normalized.

Different activations, solvers and hidden layer sizes were tried by using them as hyper-parameters to train the different models and test their performance on the validation set. In this way we could select the hyper-parameters which would be the most effective for each model. The ReLU activation function, along with the Adam solver and three hidden layers with 40 or 50 nodes per layer gave the best results on all of the models. The remaining hyper-parameters, namely the random state, beta1, beta2 were selected by running each model through a random search whereby random combinations of hyper-parameters are tested and scored until the best combination with regards to the validation set is found. Once the hyper-parameters were selected, the models could be trained on the combined training and validation set.

Each model's overall accuracy, precision, recall and f1 scores were calculated along with the precision, recall and f1 score for each individual preposition using the formulaes in [15]. Model 7 was selected since it gave the highest scores, with a 0.527 overall accuracy, a 0.707 overall precision, a 0.611 overall recall and a 0.656 overall f1 score. The model's overall accuracy score of 0.527 was a slight improvement over the score of 0.525 obtained in [5] when using geometric and language features.

3. EXPERIMENTS

3.1 Testing the Model on Labelled Images from VRD

The best model from the training process discussed above, model 7, was tested on the VRD [8] dataset. This was done with the objective of checking each model for data drift on one of the most widely used datasets in the field in order to obtain a better understanding of how it would perform on frames taken from wild videos. After downloading the VRD dataset, it had to be filtered to include only objects and relationships found in SpatialVOC2K, since this is what our model was trained to identify. After doing this the model was tested on the filtered dataset.

3.2 Testing the Model on Unlabelled Videos

When developing the system, the aim was to run it on all of the videos in our unlabelled dataset and save the classifier's prediction for each object pair detected in each frame of every video in the dataset. Videos were sampled at 10 frames per second with the intention of minimising the system's running time while still accurately capturing each object's movement and the resulting changes to the spatial relationships between object pairs. This value could also be changed manually to match the rate of change and movement in a specific video. After detecting the objects in each frame and filtering out the bad detections as described in the Object Detector section, we ignored frames where no object pairs, and therefore no spatial relationships, were present. In the accepted frames, the bounding boxes of the detected objects were used to generate a list of features. The list consisted of the selected geometrical features and the language features generated by the GloVe word embeddings, since model 7 was used. This list of multi-modal features was normalized and used to predict a confidence score for each preposition between each object pair using the previously trained multi-label classifier. These predictions were then saved, along with the corresponding frames, marked with the detected objects' class and bounding box labels. An option was also given for the full prediction to be optimised whereby successive frames with identical predictions were removed using a method inspired by [11].

The resulting predictions were analysed to see the amount of times that the system gave them confidence scores of between 0.5-0.6, 0.6-0.7, 0.7-0.8, 0.8-0.9 and 0.9-1 for each preposition. This level of control was important when it came to evaluating the model's performance on each individual preposition. In order to properly judge the model's performance on the Unlabelled Videos dataset, the predictions underwent a human evaluation. The predictions were in the form of an $\langle object - \textbf{preposition} - subject \rangle$ 3-tuple together with a confidence score higher than 0.5. Two questionnaires, each containing a sample of the model's predictions, were prepared and handed out to the participants. To prepare this sample, a random selection of a maximum of two different predictions per range of confidence scores for every preposition were made. In the questionnaire, the participants were asked to state whether they agreed or disagreed with each of the predictions. The responses for each prediction were then averaged out and compared to the model's confidence score. An evaluation of these scores was then carried out in order to study the model's performance on the dataset. In this way, any problematic patterns that the model displayed when predicting each preposition could be identified and possible solutions could be tested.

4. RESULTS

4.1 Evaluating the Model on VRD

The model displayed an accuracy of 0.31 on the VRD dataset. This value was favourable for our model since it was an expected decrease of around 0.2 from its accuracy on SpatialVOC2K. Such a decrease was expected due to the different properties of the two datasets. VRD is a single-label dataset compared to SpatialVOC2K which is multi-label and the distribution of instances among the prepositions of each dataset also differ.

4.2 Evaluating the Model on Unlabelled Videos

The human scores for each object pair example were calculated by asking a group of people whether they agree or disagree with each of the models' predictions on this set of frames. An average score was then calculated from the answers of each person and the mean value of the human evaluation scores on each set images of each preposition were calculated.

Preposition	Precision	Recall	F1	Human Score
Above	0.667	0.151	0.246	0.75
Against	0.475	0.197	0.278	0.857
Around	0.667	0.333	0.444	0
At The Level Of	0.517	0.437	0.473	0.75
Behind	0.713	0.587	0.644	0.9
Far From	0.7	0.587	0.638	0.8
In	0.478	0.524	0.5	0.8
In Front Of	0.703	0.472	0.565	0.889
Near	0.747	0.852	0.796	0.9
Next To	0.679	0.659	0.669	0.5
On	0.857	0.703	0.773	0.778
Under	0.796	0.715	0.753	0.4

Table 2: Overall Performance of the Model on the Unlabelled Videos Dataset

We will analyse the model's performance on each of the prepositions that appeared in our set of frames collected from videos. We will then attempt to explain the model's behaviour on specific correct and incorrect examples from each preposition in order to explain why they have occurred and provide possible solutions to improve future predictions. The correctness of these examples was determined by comparing the model's score to the score obtained by the human evaluation. If the human score was lower than the model's confidence score then the example was deemed to be incorrect.

For the other prepositions, namely *across*, *along*, *beyond*, *below*, *none*, *opposite* and *outside of*, the model was not able to find any instances with a confidence higher than 0.5 so we will not discuss them.

We must also pay special attention to the incorrect examples from the under-represented prepositions in our training set (i.e. prepositions with less than 300 instances in the training set) and compare them to the correct examples in an attempt to find a way to accurately identify and keep examples which are correct and discard examples which are incorrect. In this way we will see if this procedure can be automated or if a human is necessary for the final acceptance decision for each example.

Note: We will refer to the geometric features by their F1,F2,... abbreviations as listed in [5].

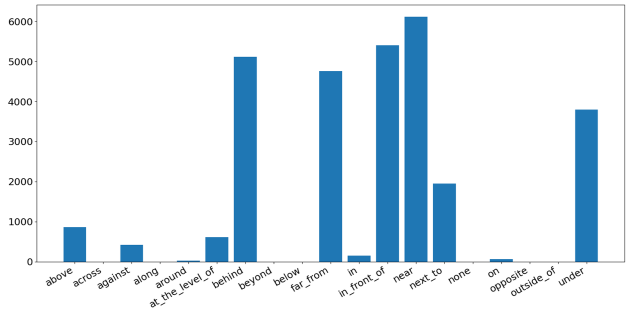

Figure 2: Number of Detected Instances from Each SpatialVOC2k Class in the Unlabelled Videos Dataset

Above: The most important features for determining *above* are the F18 (ratio between subject top edge - object top edge to object bottom edge - object top edge), F19 (ratio of subject bottom edge - object top edge to object bottom edge - object top edge), F12 (distance between bounding box centroids normalized by the union diagonal) and the language features [5]. Higher F18 values caused the model to give higher *above* confidence scores as expected since this would indicate a large vertical distance between the objects. Our model did not make use of the F12 feature, however, an alternative model which included this feature seemed to correct most of the incorrect predictions. The F19 values and the language features did not seem to have a notable effect on the model's *above* predictions.

Against: Many incorrect *against* predictions also contained high confidence scores for *on* showing that the system required a way to prevent the two prepositions from being confused, either through additional training examples or by simply disregarding such predictions. Aside from this, the most important geometrical features for predicting *against* are F5 (area of object and subject bounding box overlap normalised by minimum bounding box area), F11 (distance between bounding box centroids normalized by the union of the bounding boxes) and F12 (distance between bounding box centroids normalized by the union diagonal) [5]. Higher F5 scores seemed to be given priority over higher F11 and F12 values, leading to situations where predictions with high F5 values being incorrectly marked as *against* despite their unfavourable F11 and F12 values. This bias could either be fixed manually or through additional training examples.

Around: The fact that only 12 around training instances were available in SpatialVOC2k (Figure 1), as well as the fact that the F0 (area of object normalized by image area) and F1 (area of subject normalized by image area) features were not used in the model could explain the lack of *around* predictions (Figure 2). The language features seemed to be responsible for the *around* predictions made by the model since when an alternative model without language features was used instead, no object pairs received an *around* confidence score of over 0.5. An alternative model with F0 and F1 features added successfully eliminated the incorrect *around* predictions but did not make any correct predictions.

At The Level Of: Removing the language features seemed to drastically lower the confidence score for every *at the level of* prediction including birds, suggesting that the model had developed a bias. Removing the language features seemed to give more accurate results with F18 (ratio between subject

top edge - object top edge to object bottom edge - object top edge) values close to 0 contributing to high confidence scores as expected.

Behind: The system displayed largely positive results in its *behind* predictions. Despite this, the depth features are among the most important features for detecting *behind* relationships. This means that the system would surely benefit from a model that can collect object depth data from unseen video frames since this would contribute to more accurate *behind* predictions.

Far From: As expected, larger F18 (ratio between subject top edge - object top edge to object bottom edge - object top edge), F30 (object centroid y-axis position relative to subject centroid y-axis position as a unit vector normalized by the union of the bounding boxes), F28(object centroid y-axis position relative to subject centroid y-axis position as a vector normalized by the union of their bounding boxes) values seemed to lead to higher *far from* confidence scores. Replacing the F14 feature (euclidean distance between bounding boxes normalized by the union of their bounding boxes) with the F15 feature (euclidean distance between bounding boxes normalized by the size of the image) drastically improved the quality of *far from* predictions.

In: From the results, it was clear that the language features played a big part in the predictions. This is evident since most incorrect predictions did not have overlapping areas and had very large values of F19 (ratio of subject bottom edge - object top edge to object bottom edge - object top edge) and F30 (object centroid y-axis position relative to subject centroid y-axis position as a unit vector normalized by the union of the bounding boxes), going against the correct *in* predictions.

In Front Of: Like **behind**, the model's accuracy when making *in front of* predictions would greatly benefit from having access to object depth data. Despite this, results were largely positive with more correct predictions displaying similar F5 (area of object and subject bounding box overlap normalised by minimum bounding box area) and F30 (object centroid y-axis position relative to subject centroid y-axis position as a unit vector normalized by the union of the bounding boxes) values to each other as expected.

Near: The absence of the F22 (object centroid position relative to subject centroid position as a vector) and F23 (object centroid x-axis position relative to subject centroid x-axis position as a euclidean unit vector) geometrical features (both of which are marked as important in *near* predictions by [5]) seemed to account for most of the incorrect *near* predictions made by the model. Indeed, adding them seemed to lower the incorrect predictions' confidence scores while leaving the correct predictions' confidence scores largely unchanged.

Next To: The model seemed to give too much weight to the F14 (euclidean distance between bounding boxes normalized by the union of bounding boxes) feature since any instance with a high F14 value received a high *next to* confidence score. Replacing F14 with F15 (euclidean distance between bounding boxes normalized by image) gave similar results, this time with F18 (ratio between subject top edge - object top edge to object bottom edge - object top edge) given too much weight. The obvious solution would be to introduce more correct training instances in an effort to balance out the incorrect weighting. Another possible solution would be to remove any predictions with a high F18 values since this would mean that their vertical positions are too different for them to be considered as *next to* each other.

On: The language features seemed to be the most important features for the model when making *on* predictions since removing them drove almost every *on* confidence score down to zero. More varied training examples are required to bring down the language features to a more reliable level. Incorrect examples could be filtered out by removing positive examples with unsuitable F11 (distance between bounding box centroids normalized by the union of bounding boxes) and F30 values (object centroid y-axis position relative to subject centroid y-axis position as a unit vector normalized by the union of bounding boxes).

Under: Again, language features seemed to be the most important for making *under* predictions since an alternative model without language features reduced the confidence scores of all of the above predictions to almost 0. Aside from this, lower F30 (object centroid y-axis position relative to subject centroid y-axis position as a unit vector normalized by the union of bounding boxes) values led to higher *under* confidence scores as expected. A possible way to filter out incorrect predictions would be to check the F29 (object centroid x-axis position relative to subject centroid x-axis position as a unit vector normalized by the union of bounding boxes) values to make sure that the object is not too horizontally offset from the subject. This would work since most of the model's incorrect predictions featured objects whose x-axis positions were too far away from the subjects' x-axis positions for them to be verified as *under* by the human evaluation.

4.3 Adding Additional Training Examples to the Dataset

The hypothesis that adding additional training examples (obtained by our system from the unlabelled video dataset for under-represented prepositions) would improve the model's performance on the original test set was tested. More specifically, we test the hypothesis on the *above*, *against* and *under* prepositions since they are under-represented in the SpatialVOC2k dataset (Figure 1). Three models were trained for each preposition and the precision, recall and f1 scores of each were compared to those seen in the original model. The three models use a different amount of additional training examples. The first uses each example given a confidence score of above 0.5 by the system, the second uses each example given a confidence score of above 0.8 by the system and the third uses each example accepted by the human evaluation.

The *Above* preposition did not seem to yield positive results, with the added examples causing the model to perform worse on the test set, particularly when the examples accepted by the human evaluation were added. This could be because there were too little *above* examples to begin with (53) resulting in our system having an understanding of *above* that is different from a human's. On the other hand, *against* yields more positive results, displaying varying degrees of improvement when adding the different sets of training examples, the most notable of which was from the set of additional examples obtained from the human evaluation. Lastly, *under* displays the most positive results, perhaps because of the large amount of additional examples that the system obtained from the unlabelled video dataset. Some noise was clearly present in the >0.5 set of examples,

resulting in the precision recall and f1 scores reducing but the majority of noise was filtered out in the other two sets of examples resulting in improvements to the scores.

Above	Instances	Precision	Recall	F1
Original (53 examples)	-	0.667	0.151	0.246
Confidence Score >0.5	415	0.56	0.603	0.788
Confidence Score >0.8	276	0.571	0.075	0.133
From Human Evaluation	49	0.25	0.075	0.116
Against	**Instances**	**Precision**	**Recall**	**F1**
Original (193 examples)	-	0.475	0.197	0.278
Confidence Score >0.5	418	0.603	0.197	0.297
Confidence Score >0.8	217	0.478	0.228	0.308
From Human Evaluation	53	0.538	0.202	0.287
Under	**Instances**	**Precision**	**Recall**	**F1**
Original (158 examples)	-	0.796	0.715	0.753
Confidence Score >0.5	3794	0.788	0.684	0.732
Confidence Score >0.8	2576	0.832	0.702	0.755
From Human Evaluation	442	0.844	0.652	0.736

Table 3: Results from Adding Additional Training Instances for the Under-Represented Prepositions

5. CONCLUSIONS

In this project, we developed a system which uses a model trained on the SpatialVOC2k dataset to identify spatial relationships between objects found in unlabelled frames of videos in the wild. The system was subjected to a human evaluation and its performance on each SpatialVOC2k preposition was analysed. The set of spatial relationship examples outputted by the system contained a large amount of noise and the human beings involved in the evaluation often disagreed with both the model and each other when marking whether the preposition used to describe the spatial relationship between a pair of objects was correct or incorrect. Two possible ways to clean the outputted set of examples for each preposition were therefore found in an effort to improve the Spatial Relationship model. The first would be to use a separate model for each preposition, using a different list of features for each as discussed in the evaluation. The second would be to have a human being manually accept or reject each outputted example. While the latter solution is more time consuming, it is the most effective way of ensuring near-perfect results and is still more efficient than obtaining quality annotations from scratch.

Improving the model in these ways would improve the quality of the system's output and, in turn, make it more effective at its applications, in particular in dataset enrichment. This experiment proved to be successful since the addition of examples to the original training set enabled retrained models to give better precision, recall and f1 scores for the under-represented *above*, *against* and *under* prepositions. Despite this, a more effective original model would lead to less noise in the system's output, allowing for better spatial relationship annotations to use in applications such as this one. Such a model could be obtained in the future by making use of the depth and pose features offered by SpatialVOC2k. Specialised models would need to be trained to recognise object depth and pose from unseen video frames for this to be possible.

Since the applications of spatial relationship detection in videos remain largely unexplored, the system developed in this project serves as a baseline for further research and experiments in the area.

6. REFERENCES

[1] A. G. A. Muscat. Predicting visual spatial relations in the maltese language. 2018.

[2] A. Belz, A. Muscat, P. Anguill, M. Sow, G. Vincent, and Y. Zinessabah. SpatialVOC2K: A multilingual dataset of images with annotations and features for spatial relations between objects. In *Proceedings of the 11th International Conference on Natural Language Generation*, pages 140–145, Tilburg University, The Netherlands, Nov. 2018. Association for Computational Linguistics.

[3] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao. Yolov4: Optimal speed and accuracy of object detection, 2020.

[4] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2008 (VOC2008) Results. http://www.pascal-network.org/challenges/VOC/voc2008/workshop/index.html.

[5] G. Farrugia and A. Muscat. Explaining spatial relation detection using layerwise relevance propagation. pages 378–385, 01 2020.

[6] N. Gkanatsios, V. Pitsikalis, P. Koutras, A. Zlatintsi, and P. Maragos. Showcasing deeply supervised multimodal attentional translation embeddings: a demo for visual relationship detection. In *2019 IEEE International Conference on Image Processing (ICIP)*, pages 2456–2456, 2019.

[7] S. Li, H. Yu, J. Zhang, K. Yang, and B. Ran. A video-based traffic data collection system for multiple vehicle types. *Intelligent Transport Systems, IET*, 8:164–174, 03 2014.

[8] C. Lu, R. Krishna, M. S. Bernstein, and F. Li. Visual relationship detection with language priors. *CoRR*, abs/1608.00187, 2016.

[9] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space, 2013.

[10] A. Muscat and A. Belz. Learning to generate descriptions of visual data anchored in spatial relations. *IEEE Computational Intelligence Magazine*, 12(3):29–42, 2017.

[11] A. Nasreen and G. Shobha. Reducing redundancy in videos using reference frame and clustering technique of key frame extraction. In *International Conference on Circuits, Communication, Control and Computing*, pages 348–440, 2014.

[12] J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014.

[13] B. Sapp, A. Saxena, and A. Ng. A fast data collection and augmentation procedure for object recognition. pages 1402–1408, 01 2008.

[14] M. Stein, H. Janetzko, T. Breitkreutz, D. Seebacher, T. Schreck, M. Grossniklaus, I. D. Couzin, and D. A. Keim. Director's cut: Analysis and annotation of soccer matches. *IEEE Computer Graphics and Applications*, 36(5):50–60, 2016.

[15] M.-L. Zhang and Z.-H. Zhou. A review on multi-label learning algorithms. *IEEE Transactions on Knowledge and Data Engineering*, 26(8):1819–1837, 2014.