Automated Data Collection Using Spatial Relationship Detection in Videos

Martin Bartolo

Supervisor: Adrian Muscat

Faculty of ICT

University of Malta

30th June 2021

Submitted in partial fulfillment of the requirements for the degree of B.Sc. I.C.T. (Hons.)

Faculty of ICT

Declaration

I, the undersigned, declare that the dissertation entitled:

Automated Data Collection Using Spatial Relationship Detection in Videos

submitted is my work, except where acknowledged and referenced.

Martin Bartolo

30th June 2021

Acknowledgements

First and foremost, I would like to extend my sincere gratitude and appreciation to my supervisor Prof. Adrian Muscat for his guidance throughout the course of this project. I would also like to thank my mother, father and sister for providing encouragement and being patient with me over the past few months. A special thanks goes to my close friends, who have always been a source of support to get me through the most discouraging times. Finally, I would like to thank my cat Bino whose work ethic and professionalism is an endless source of inspiration.

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.

Contents

List of Figures

List of Tables

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 [9]. 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 from videos. This application is relevant since the creation of datasets in domains such as computer vision and human-machine interaction 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 project, we develop a system that will be able to go through reels and reels of videos 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. The objective of gathering these

instances will be to develop custom datasets or enrich existing ones by adding training instances that include under-represented relationships. In this way, larger datasets can be obtained, allowing for lower variance models to be trained on it.

When detecting spatial relationships in videos, we must first split the video into frames and consider each frame separately. The objects in the frame must be detected. For this task, the latest version of YOLO [18], YOLOv4 [3], will be installed and the pre-trained MS COCO [12] weights will be set up. The object detections will then be filtered to include only objects contained in the VOC dataset [4], removing irrelevant objects, detections with low confidence scores and double detections in the process. We will skip frames which are identical to the previous or contain less than 2 objects. Next, we will generate a list of geometrical features explained in [5] and language features using word embeddings [14, 17] for each pair of objects. Various spatial relationship models will be specified and trained on the SpatialVOC2K [2] dataset using different combinations of these geometric and language features in order to find the one which gives the most accurate results on test sets generated from both the SpatialVOC2K [2] and VRD datasets [13].

We will use the most accurate of the tested models on a dataset of videos collected from open-source stock video websites and take a few random samples of high confidence predictions of each of SpatialVOC2K's prepositions on these videos. These samples will then be subject to a human evaluation in which people will be asked whether or not they agree with the system's predictions on these samples. The results for each preposition, with particular attention being given to those of the under-represented prepositions, will be evaluated to determine their relevancy and to deduce a way to accurately filter out incorrect predictions for each preposition. The remaining correct predictions could then be used for dataset enrichment. This concept will be tested by adding predictions that include under-represented prepositions to the SpatialVOC2K [2] dataset and retraining a model to see if it would improve on our previous model's results on the same SpatialVOC2K [2] and VRD [13] test sets.

2. Literature Review

Object detection is one of the most widely researched computer vision tasks, meaning that there are plenty of available models to consider. One such model must be selected for this project, in order to detect objects from video frames, label them and record their bounding boxes. Two popular object detection frameworks are R-CNN [7] and YOLO [18]. R-CNN, or Region Based Convolutional Neural Networks, function by extracting around 2000 regions from an image, with each serving as a bounding box proposal from which the correct boxes can be selected and identified. More efficient versions which reduce the overhead time introduced by this region proposal method include Fast R-CNN [6] and Faster R-CNN, [19] the latter of which will be used in our comparison. YOLO [18], or You Only Look Once, splits the image into a grid and then takes cells in the grid as bounding boxes if the probability of them having a specific label is above a predefined threshold. At the time of writing, YOLOv4 [3] is the framework's latest version so it will be used in our comparison between various popular object detection frameworks in Table 2.1.

After detecting the objects, the most salient must be selected depending on which would be most likely to be mentioned by humans when describing the frame [24, 23]. The confidence score assigned to each object detection, as well as the bounding boxes' sizes and areas of overlap with other bounding boxes can be taken into consideration to eliminate any insignificant objects from the detection.

Once we have a list of significant objects in the frame, the spatial relation-

Framework	**FPS**	**AP**	**AP**$_{50}$	**AP**$_{75}$
YOLOv4	54	41.2%	62.8%	44.3%
CenterMask-Lite	50	30.2%	-	-
EFGRNet	47.6	33.2%	53.4%	35.4%
HSD	40	33.5%	53.2%	36.1%
Faster R-CNN	9.4	39.8%	59.2%	43.5%

Table 2.1: Comparison of speed and accuracy of various object detection frameworks on the COCO [12] (test-dev 2017) dataset using a Pascal GPU. Values taken from [3]

ships between each pair of objects may be detected. Deducing spatial relationships between objects has long been a topic of interest in the field of Computer Vision and is a key step in image understanding. The most conventional example of this is to detect these relationships without the use of bounding boxes, using only the objects and space around them to make detections [10]. However, using bounding boxes opens the door for more controlled detections by using them to generate predetermined geometrical features between the object pairs such as their relative positions, sizes and spatial distributions [1, 27]. The addition of language features introduces a semantic representation for each object, which gives context and insight to detections. Along with improving the accuracy of regular spatial detections [25] (For example: $\langle person - \textbf{on} - bicycle \rangle$ makes more semantic sense than $\langle person - \textbf{under} - bicycle \rangle$ since it is common for a person to be on a bicycle but rare for a person to be under a bicycle), they can be used to give a deeper understanding of the image through contextual detections such as $\langle person - \textbf{riding} - bicycle \rangle$ or $\langle person - \textbf{pushing} - bicycle \rangle$ [13].

Detecting these spatial relationships from videos means that large amounts of data must be processed, a lot of which is redundant. A proposed idea in this project is to reduce the amount of meaningless frames that are processed is to extract key frames and compare the sets of frames that follow them, removing each frame that is too similar to the key frame. When a different enough frame is found, this frame will be set as the key frame and the process is repeated [16]. In this way, the amount of frames that will be processed will be reduced, leading to faster

processing times and more meaningful results.

The detection of visual relationships from videos has been showcased using custom video datasets [21], live video streams [8] and video frames in the wild [8]. While more challenging than detecting these relationships from images, detecting visual relationships from videos gives rise to new dynamic relationships between objects such as $\langle dog - \textbf{follow} - person \rangle$ throughout short and long sets of frames [21]. Despite these advancements, its applications have not yet been explored. In fact, literature related to the applications of visual relationship detection results in general is scarce.

This project aims to propose and explore one such application of video spatial relationship detection which is its use in customisable automated data collection. Automated data collection from videos is not a new concept. Interesting examples include the collection of football match analysis data from video recordings by detecting the relative positions of football players and the ball, [22] and the collection of traffic data including vehicle types and vehicle speeds from video recordings [11]. Attempts to streamline the data collection procedure for object detection datasets have also been made, an example of which being to use a green screen and take pictures of various objects which can then be automatically placed on different backgrounds to create a large amount of training instances from a small set of manually captured object images [20]. In our case, we will attempt to extract meaningful spatial relationship detections from videos which can be used to build new custom datasets or enrich existing ones with additional training instances.

3. Methodology

3.1 Datasets

Throughout the course of the project, a number of different datasets are used. We make use of two object detection datasets in order to identify objects from images, namely the Microsoft Common Objects in Context, shortened to COCO [12] and the PASCAL Visual Object Classes 2008 dataset [4], shortened to VOC. To train our spatial relationship model the SpatialVOC2K dataset [2] is used. We also use this dataset, along with the Visual Relationship Detection (VRD), dataset [13] to test model's performance on labelled images. Lastly, we create a dataset of unlabelled videos which we will use to evaluate our model's performance in its intended use. Let us now explain each dataset, together with its purpose in the project, in more detail.

3.1.1 COCO

The COCO dataset consists of over 200,000 images including over 1.5 million objects whose classes and bounding boxes have been labelled. The objects are depicted in realistic scenes making it suitable for our project since we must detect objects in frames taken from wild videos. These objects are split into 80 classes (a full list is given in Table A.2), meaning that our object detection model, which uses weights that have been pre-trained on COCO, can detect occurrences of these 80 classes of objects in images. Even though we will only keep object detections

from 20/80 of these classes, the dataset's ability to detect a large number of objects that are commonly found in the wild means that the project has the potential to be scaled up to work on all 80 objects in the future.

3.1.2 VOC

The 2008 iteration of the PASCAL VOC dataset consists of 4340 images containing 10,363 objects whose classes and bounding boxes have been labelled. These objects are split into 20 classes (a full list is given in Table A.1). Each of these object classes is also found in the COCO object class list (see Table A.2), meaning that our object detection model can detect all VOC object classes despite it using COCO weights. We do not use any models trained on VOC but it is important to have a good understanding of this dataset due to its use in SpatialVOC2K.

3.1.3 SpatialVOC2K

This dataset contains 2,026 images, annotated with spatial relationships between objects in the 20 VOC 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) as shown in A.3. 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 (see Figure A.1). This motivates our aim to generate correct examples of under-represented preposition instances in an attempt to improve a model's performance when predicting these instances. 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. Adding such models to our pipeline is a potential future improvement. 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.

3.1.4 VRD

The VRD dataset is made up of 5000 images with 37,993 relationships between object pairs. These relationships are split into 70 classes and depict the interactions between pairs of objects which, themselves, are split into 100 classes. Most, but not all, of the prepositions in the SpatialVOC2K dataset are contained in the list of VRD relationship classes and all 20 of the VOC object classes are contained in the list of VRD object classes. We use the VRD dataset as a benchmarking tool to test our trained spatial relationship detection model. This will be accomplished by taking each VRD instance that only contains objects and relationships whose classes are also in the SpatialVOC2K lists of classes, and seeing how well the model's relationship predictions match with each instance's list of correct relationships.

3.1.5 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 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.

3.2 Object Detector

Before detecting the spatial relationships between object pairs, the objects themselves must be identified. An object detector is a model that has been trained to recognize a number of object classes in unseen images. It takes these images as an input and returns a list of objects which are labelled with their classes, bounding boxes and the model's confidence in their prediction. 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 (see Table 2.1), 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.

A YOLOv4 implementation in Tensorflow 2.0 was installed and pre-trained weights were downloaded and converted to work on Tensorflow. The weights were pre-trained on the COCO dataset [12], meaning that the object detector could identify all COCO object classes straight out of the box. Since we only required objects from the VOC classes (see Table A.1) to be identified, objects which were in COCO classes that were not part of the 20 VOC classes (see cells in A.2 that are not highlighted) were ignored. The remaining identified objects' class numbers

were then changed to match the VOC class numbers. The COCO weights were preferred to pre-trained VOC weights since the VOC weights available at the time of writing were not yet updated to work with YOLOv4. Additionally, using COCO weights would allow us to expand the project in the future to include all COCO weight classes.

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. An example is shown in Figure A.2. 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. An example of such a case is shown in Figure A.3. 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.

3.3 The Features

From the object detections, we generated 39 geometrical features [5] (listed in full in Table A.4) and a number of language features. A subset of the geometrical features, which we will refer to as the partial set (highlighted in blue in Table A.4), was selected from the full set to see if it would give better results [1, 15]. Language features were made using word embeddings and one-hot encoding and

were added to the geometrical features to make the full feature list that we used to train our spatial relationship models. 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. Let us explore both types of features in more detail.

3.3.1 Geometrical Features

Each of these 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 (as shown in Table A.4) with the intention of maximising the model's performance [15]. The features were initially chosen using a greedy backward feature elimination procedure as discussed in [15] 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.

3.3.2 Language Features

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: $\langle car - \mathbf{on} - person \rangle$ is unlikely but $\langle person - \mathbf{on} - bicycle \rangle$ is likely). The first type was a one-hot encoded vector, encoded using sklearn's *MultiLabelBinarizer()* function. 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 [17] 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 [14] word embeddings which represent each object class as a 300-dimensional vector of real numbers.

3.4 Training the Spatial Relationship Model

The aim of the next task was to train a multi-label classifier using sklearn's *MLP-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 the features explained in the previous section 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. Table 3.1 shows the features used in each of the models.

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

Table 3.1: Features used to train each model

When training each model, the full list of features was composed as shown in Algorithm 1. This list of features is 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 for would be 80% training and 20% test. The training/validation split was also 80%/20%. The full list of features was then normalized using sklearn's *StandardScaler()* function which was trained on the feature training set and then saved to be used later when normalizing both the training and test sets.

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 using Algorithm 2.

The performance of each model on the test set was measured using a number of different metrics. 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 [26]. 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. Its overall precision, recall and f1 scores also showed a slight improvement to the scores in [5]. A full list of each model's scores, including the selected model 7 is shown in Table A.5

3.5 Experiment 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 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 using Algorithm 3.

3.6 Experiment 2: Testing the Model on Unlabelled Videos

In order to test the system on unlabelled videos, a system was developed that would iterate through frames of the video, detect the objects in each frame and then predict the spatial relationships between them. These predictions were then evaluated by a group of people to see how well the average human score for each would compare to the model's confidence score.

3.6.1 Running the System

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. This value was chosen 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. Values higher than 10

resulted in many redundant frames with little to no changes between one frame and the next. Values lower than 10 frames per second, on the other hand, resulted in many objects' movements being missed. The 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 the detection using the process described in Section 3.2 (Object Detector), we ignored each frame that contained less than 2 objects since this meant that there were no object pairs present and therefore no spatial relationships to predict. The bounding boxes of the detected objects were then 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 to a text file and the corresponding frames, marked with the detected objects' class and bounding box labels, were also saved. An option was also given for the full prediction to be optimised. The optimisation process is outlined in Algorithm 5 and was inspired by [16]. The full algorithm used to generate a full set of predictions of each frame of the unlabelled videos dataset is outlined in Algorithm 4. An example of a prediction outputted by the system, along with its accompanying labelled frame in Figure A.4.

3.6.2 Filtering the Frames

Once we had the full set of predictions, a script was developed to allow us obtain specific lists of predictions filtered by preposition, object and subject (for example we could filter out all predictions satisfying: $\langle person - \mathbf{any} - any \rangle$ or $\langle any - \mathbf{near} - car \rangle$). Aside from providing us with useful insight with regards to the model's performance on the dataset, the script also allowed us to count the amount of times that the model gave a confidence score higher than 0.5 for each preposition (see Figure A.5). Additionally, the script could also show us the amount of times

that it gave confidence scores of 0.5-0.6, 0.6-0.7, 0.7-0.8, 0.8-0.9 and 0.9-1 for each preposition (see Figure A.6). This level of control was important when it came to evaluating the model's performance on each individual preposition.

3.6.3 The Human Evaluation

In order to properly judge the model's performance on the Unlabelled Videos dataset, the predictions underwent a human evaluation. In this case, 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 two groups of 12 and 16 people respectively. To prepare this sample, a random selection of a maximum of two different predictions per range of confidence scores for every preposition (Two per histogram bin in Figure A.6) were made. If two were not available then either a single prediction or no prediction was included, depending on what was available. These predictions were then evenly split between the two questionnaires to ensure that they did not take too much time for the people to complete. In the questionnaire, the participants were asked to state whether they agreed or disagreed with each of the predictions (see Figure A.7). 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. These solutions included using a different set of geometric features for specific prepositions and filtering out preposition predictions that coincided with others. These are discussed in detail in the Evaluation section.

4. Results and Evaluation

4.1 Experiment 1: Evaluating the Model on VRD

After carrying out the experiment as outlined in the Methodology, 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. Additionally, the distribution of instances among the prepositions of each dataset are very different (see Figures A.1 and A.8), meaning that the model is bound to lose accuracy when being tested on VRD. Despite this, the amount that the accuracy decreased was not alarming.

4.2 Experiment 2: Testing the Model on Unlabelled Videos

4.2.1 Analysing the Model's Overall Performance

The human evaluation 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 are calculated. These values are shown in Table 4.1.

Preposition	Precision	Recall	F1	Mean Human Evaluation 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 4.1: Overall Performance of the Model on the Unlabelled Videos Dataset

4.2.2 Analysing the Model's Performance on Specific Prepositions

In this section, we will analyse the model's performance on each of the prepositions that appeared in our set of frames collected from videos. We will also 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 evaluation score was lower than the model's confidence score then the example was deemed to be incorrect.

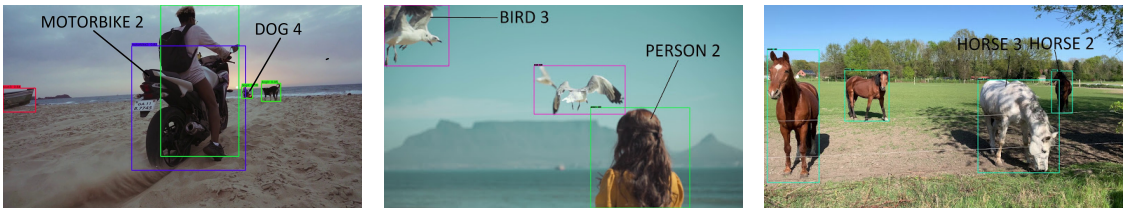
For the other prepositions, namely *across*, *along*, *beyond*, *below*, *none*, *opposite* and *outside of*, the model was not able to find any instances with a confidence higher than 0.5 so we will not discuss them in this section.

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. In a later section, we will then pick up a number of good instances of these under-represented prepositions from our video set and add them to the training set after which we will retraining the model to see if the f1 scores for these prepositions will increase.

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

Above: The model was trained on 53 instances of above and gave a precision of 0.667, a recall of 0.15 and an f1 score of 0.246. This number of training instances, together with the low recall and f1 scores made it the perfect under-represented preposition for us to test our hypothesis on. We will do this in a later section.



False Positive
$\langle motorbike2 - \mathbf{above} - dog4 \rangle$

True Positive
$\langle bird3 - \mathbf{above} - person2 \rangle$

True Negative
$\langle horse2 - \mathbf{above} - horse3 \rangle$

Figure 4.1: *Above* prediction samples

The model's prediction of $\langle motorbike2 - \mathbf{above} - dog4 \rangle$ was marked as incorrect by the human evaluation. The model gave it a confidence score of 0.675 while the human evaluation score of 0.1875 disagreed with the model's score. The model correctly gave a confidence score of 0.554 to $\langle bird3 - \mathbf{above} - person2 \rangle$, agreeing with the human evaluation score of 0.917.

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]. We will analyse the effect that each of these features had on the aforementioned predictions.

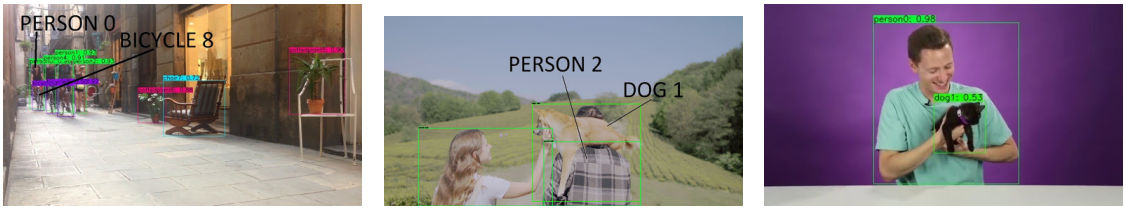
When running an alternative model that does not make use of language features on the same image, $\langle motorbike2 - \mathbf{above} - dog4 \rangle$ is still given a confidence score of above 0.5 so the language features are not the problem.

The F18 and F19 features gave values of 0.914 and -0.164 respectively for the incorrect $\langle motorbike2 - \mathbf{above} - dog4 \rangle$ prediction, and 2.752 for the correct $\langle bird3 - \mathbf{above} - person2 \rangle$ prediction. These features also had values of 0.867 and -0.1 respectively for $\langle horse2 - \mathbf{above} - horse3 \rangle$, which was correctly given a low confidence score of 0.24 for above. This proved that a higher value for the F18 feature indicated that the object is above the subject while a F19 value of less than 0 seemed to indicate that the object is not above the subject. This F19 feature was being overlooked by the model, since even when it had a negative value, above was still given a high confidence score.

Since our model did not make use of the F12 feature, a new model that included this feature was trained to see how this would affect the above examples. While this model had lower overall precision, recall and f1 scores of 0.545, 0.113 and 0.187 respectively for the above preposition, it correctly gave a low confidence score of 0.002 to the previously incorrect $\langle motorbike2 - \mathbf{above} - dog4 \rangle$ prediction. It also maintained an above confidence score of higher than 0.5 for the correct $\langle bird3 - \mathbf{above} - person2 \rangle$ prediction.

This gives us 2 possible methods to filter out the incorrect predictions for above by our model. One method would be to disregard instances with F19 values of lower than 0 since they led to false positives in our video set. The other method would be to use the alternative model with the added F12 feature to search for above instances instead of our original model.

Against: The model was trained on 193 instances of *against* and gave a precision of 0.475, a recall of 0.197 and an f1 score of 0.278. Like above, the number of training instances and the low recall and f1 scores make it an ideal underrepresented preposition for us to attempt to retrain a model on later.



False Positive
$\langle person0 - \textbf{against} - bicycle8 \rangle$

False Positive
$\langle person2 - \textbf{against} - dog1 \rangle$

True Positive
$\langle person0 - \textbf{against} - dog1 \rangle$

Figure 4.2: *Against* prediction samples

The incorrect $\langle person0 - \textbf{against} - bicycle8 \rangle$ prediction, with a confidence score of 0.842, was one of a series of predictions where the model gave a high confidence score to both *against* and *on* simultaneously. The human evaluation deemed *against* to be unsuitable for these object pairs so it is clear that the model would benefit from the use of a probabilistic modelling of the interdependencies between prepositions to prevent *against* and *on* from being given high confidence scores on the same object pairs.

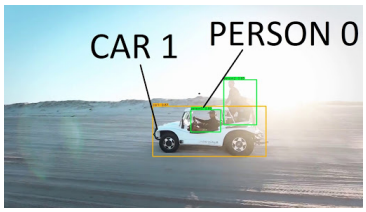
To analyse the remaining *against* predictions, let us look at their features. The most important features for predicting against are F5 (area of object and subject bounding box overlap normalised by minimum bounding box area), F11 (distance between bounding box centroids normalized by the union of the bounding boxes) and F12 (distance between bounding box centroids normalized by the union diagonal). The F5 feature for $\langle person2 - \textbf{against} - dog1 \rangle$, was higher than that of $\langle person0 - \textbf{against} - dog1 \rangle$ explaining why it incorrectly received a higher confidence score of 0.719 compared to the latter's score of 0.517. This shows that the against training set could benefit from increasing the weighting of the F11 and F12 geometrical features since objects whose centroids are closer together are more likely to be described as *against* by humans. More training examples are required to do this and ensure that these features, as well as the language features, carry the correct weight when making predictions.

Around: The model was trained on 12 instances of *around* and gave a precision

of 0.667, a recall of 0.333 and an f1 score of 0.444.

False Positive
$\langle train2 - \textbf{around} - person1 \rangle$

False Positive
$\langle train1 - \textbf{around} - person2 \rangle$



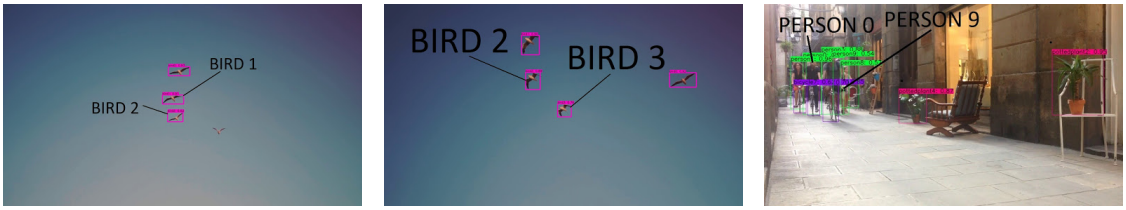
False Negative
$\langle car1 - \textbf{around} - person0 \rangle$

Figure 4.3: *Around* prediction samples

The most important features for predicting around are F0 (area of object normalized by image area), F1 (area of subject normalized by image area), and the language features. The fact that F0 and F1 were not used in the model could explain the lack of *around* predictions. The language features accounted for the *around* predictions made by the model since when an alternative model without language features was used instead, the object pairs no longer received an around confidence score of over 0.5.

Another model was trained with the F0 and F1 features added to it but this did not make any changes. In fact, the overall precision, recall and f1 scores for around remained identical when using this new model. The $\langle car1 - \textbf{around} - person0 \rangle$ prediction still had a confidence score of lower than 0.5 while the $\langle train2 - \textbf{around} - person1 \rangle$ and $\langle train1 - \textbf{around} - person2 \rangle$ confidence scores remained high. An additional model with the F0 and F1 features but no language features was trained. This model once again eliminated the $\langle train2 - \textbf{around} - person1 \rangle$ and $\langle train1 - \textbf{around} - person2 \rangle$ predictions but did not successfully pick up the $\langle car1 - \textbf{around} - person0 \rangle$ example.

At The Level Of: The model was trained on 316 examples and received a precision score of 0.517, a recall score of 0.438 and an f1 score of 0.473 on the training set.



False Positive $\langle bird2 - \textbf{at the level of} - bird1 \rangle$

True Negative $\langle bird2 - \textbf{at the level of} - bird3 \rangle$

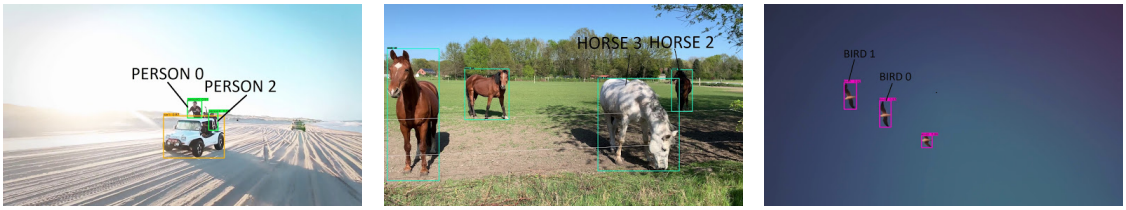
True Positive $\langle person9 - \textbf{at the level of} - person0 \rangle$

Figure 4.4: *At The Level Of* prediction samples

The most important features for predicting this preposition are F5 (area of object overlap normalized by minimum area), F20 (object area ratio - maximum to minimum), F18 (ratio between subject top edge - object top edge to object bottom edge - object top edge) and F3(area of object normalized by the union of areas). Removing the language features lowered the confidence score of the incorrect $\langle bird2 - \textbf{at the level of} - bird1 \rangle$ prediction from 0.906 to 0.388, suggesting that the original model developed a bias towards birds being at the same level of each other. Despite this, it also reduced the confidence of the correct $\langle person9 - \textbf{at the level of} - person0 \rangle$ prediction from 0.801 to 0.257 suggesting that the language features were detrimental in examples containing birds.

Aside from the language features, F18 values closer to 0 seemed to increase the confidence score of the *at the level of* predictions. $\langle bird2 - \textbf{at the level of} - bird1 \rangle$ received a confidence score of 0.906 with an F18 value of -0.2, $\langle person9 - \textbf{at the level of} - person0 \rangle$ received a confidence score of 0.801 with an F18 value of -1.19 and $\langle bird2 - \textbf{at the level of} - bird3 \rangle$ received a confidence score of 0.04 with an F18 value of 1.9. Interestingly, the F5 and F20 feature values which were both based on the objects' areas did not seem to affect the confidence scores.

Behind: The model achieved a precision of 0.713, a recall of 0.587 and an f1 score of 0.644 on the SpatialVOC2k dataset after being trained on a set of 504 examples.



False Positive
$\langle person0 - \textbf{behind} - person2 \rangle$

True Positive
$\langle horse2 - \textbf{behind} - horse3 \rangle$

True Negative
$\langle bird0 - \textbf{behind} - bird1 \rangle$

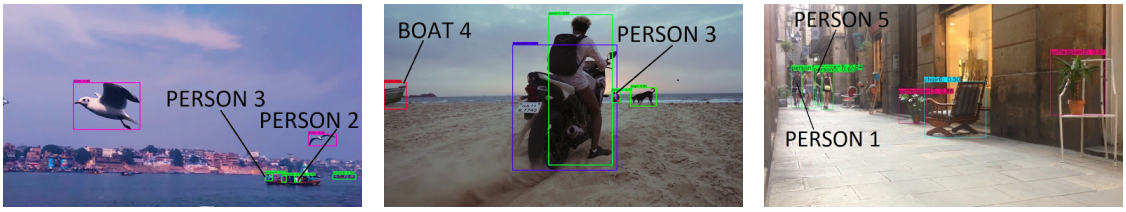
Figure 4.5: *Behind* prediction samples

The most important features for detecting *behind* relationships are depth features, language features, F18 (ratio between subject top edge - object top edge to object bottom edge - object top edge) and F28 (object centroid y-axis position relative to subject centroid y-axis position as a vector normalized by the union of their bounding boxes). Depth features were not used since this would have required an additional model that would have to be trained to collect this object depth data from videos.

When predicting the above images using an alternative model without language features, the confidence score of the incorrect $\langle person0 - \textbf{behind} - person2 \rangle$ prediction was reduced to 0.4 while the correct $\langle horse2 - \textbf{behind} - horse3 \rangle$ remained above 0.5. This suggests that a bias was introduced by the language features that made the model give confidence scores that were too high to behind relationships between people.

While the F18 feature did not seem to have much effect on the model's *behind* predictions, object pairs with a F28 value of between -1 and -2 displayed higher confidence scores for behind. Indeed, $\langle person0 - \textbf{behind} - person2 \rangle$ and $\langle horse2 - \textbf{behind} - horse3 \rangle$ were both given high confidence scores and had similar F28 values of -1.52 and -1.46 while $\langle bird0 - \textbf{behind} - bird1 \rangle$ was given a confidence score of 0.037 and had a much higher F28 value of 1.59.

Far From: The model was trained on 242 training instances, obtaining a precision of 0.699, a recall of 0.587 and an f1 score of 0.638. Since the number of training instances was below 300, we will attempt to find a way to filter out incorrect predictions so that we can add the remaining correct predictions to the training set. This enlarged training set could then be used to retrain the model to improve the precision, recall and f1 scores.



False Positive
$\langle person2 - \mathbf{far\ from} - person3 \rangle$

True Positive
$\langle boat4 - \mathbf{far\ from} - person3 \rangle$

True Negative
$\langle person5 - \mathbf{far\ from} - person1 \rangle$

Figure 4.6: *Far From* prediction samples

Similar to *behind*, one of the most important features when making *far from* predictions is the depth feature, which we are not collecting. Our results could benefit from the addition of a model that can collect these features. Additionally the 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) and F28(object centroid y-axis position relative to subject centroid y-axis position as a vector normalized by the union of their bounding boxes) geometrical features are also important in making *far from* predictions.

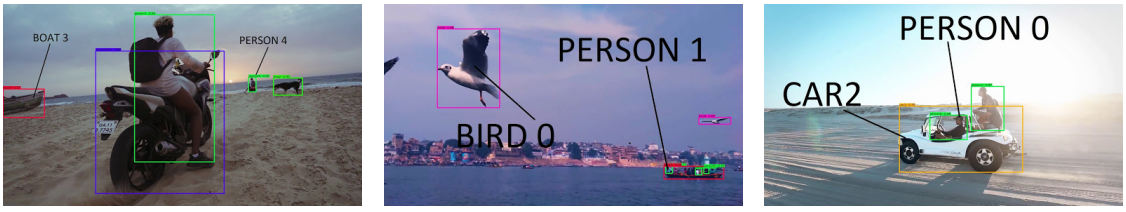
Language features seemed to have played a big part in a lot of our model's predictions of this preposition. When they were removed, the incorrect $\langle person2 - \mathbf{far\ from} - person3 \rangle$ prediction's confidence score went from 0.53 down to 0.024. However, the correct $\langle boat4 - \mathbf{far\ from} - person3 \rangle$'s confidence score also reduced from 0.706 to 0.398. This shows that the model could benefit from a larger set of correct far from examples with more varied objects to reduce the bias introduced

by the language features.

Large F18, F30 and F28 values were also present in most of the far from predictions, except for some incorrect predictions such as the above $\langle person2 - \textbf{far from} - person3\rangle$. These predictions were likely the result of the language features but the small union of their areas, leading to larger F30 and F28 values may have also contributed to their high confidence scores.

Using an alternative model that included the addition of the F15 feature (euclidean distance between bounding boxes normalized by the size of the image) and the removal of the F14 feature (euclidean distance between bounding boxes normalized by the union of their bounding boxes) proved to have positive results. Even though the precision, recall and f1 scores reduced to 0.667, 0.512 and 0.579 respectively, the incorrect $\langle person2 - \textbf{far from} - person3\rangle$ prediction received a confidence score lower than 0.5 while the correct $\langle boat4 - \textbf{far from} - person3\rangle$ prediction received a higher confidence score of 0.947.

In: The model was trained on 21 instances of *in* with precision, recall and f1 scores of 0.478, 0.524 and 0.5 respectively. *In* is another ideal candidate for retraining by adding our model's correct predictions to the training set. Let us attempt to find a way to filter out the incorrect predictions that our model has made.



False Positive
$\langle person4 - \textbf{in} - boat3\rangle$

False Positive
$\langle person1 - \textbf{in} - bird0\rangle$

True Positive
$\langle person0 - \textbf{in} - car2\rangle$

Figure 4.7: *In* prediction samples

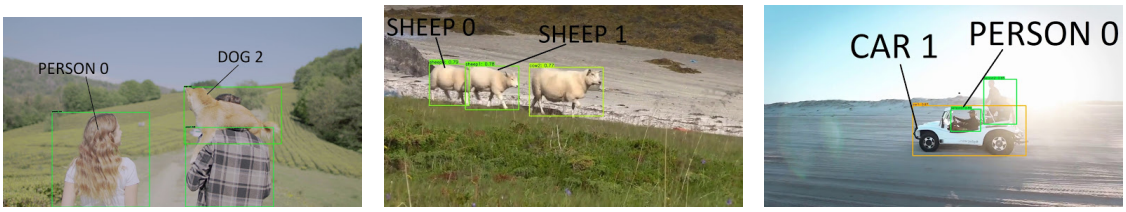
The most important features when prediction *in* are the language features, F5 (area of object and subject overlap normalized by the minimum bounding box

area), 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).

From the results, it was clear that the language features played a big part in the predictions. This is evident since the incorrect $\langle person4 - \textbf{far from} - boat3\rangle$ and $\langle person1 - \textbf{far from} - bird0\rangle$ predictions did not have overlapping areas and had very large values of F19 and F30, going against most of the correct in predictions. In fact when running a model without language features on these examples, their confidence scores reduced from 0.508 to almost 0 and from 0.685 to almost 0 respectively. The correct $\langle person0 - \textbf{far from} - car2\rangle$ prediction's confidence score also reduced to 0.1 which, although low, is much higher than that values that the incorrect predictions reduced to.

Lower F19 and F30 values were preferred and led to higher confidence scores for the *in* preposition, explaining why the correct examples maintained a higher confidence level when the language features were removed. More training examples would be needed to reduce the bias introduced by the language features in order to consistently collect correct in predictions.

In Front Of: The model was trained on 477 examples and obtained a precision of 0.703, a recall of 0.472 and an f1 score of 0.565.



False Positive $\langle dog2 - \textbf{in front of} - person0\rangle$ | True Positive $\langle sheep1 - \textbf{in front of} - sheep0\rangle$ | True Negative $\langle car1 - \textbf{in front of} - person0\rangle$

Figure 4.8: *In Front Of* prediction samples

The most important features for the model when predicting *in front of* are

the depth features, which we do not have access to, 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).

All of the predictions that were marked as positive had similar F5 and F30 values. The positive $\langle sheep1 - \textbf{in front of} - sheep0\rangle$ and $\langle dog2 - \textbf{in front of} - person0\rangle$ predictions had F5 and F30 values of -0.37 and 1.78, and -0.70 and 1.52 respectively while the negative $\langle car1 - \textbf{in front of} - person0\rangle$ prediction had values of 0.54 and 0.27. Since half of person0 in the incorrect $\langle dog2 - \textbf{in front of} - person0\rangle$ prediction is out of the frame, the object's bounding box does not represent the object's true area. If we had a model that could enlarge the bounding box according to the object's true size then the F5 and F30 values of the object pair would change and in front of would likely not have been given such a high confidence score.

Near: The model was trained on 1048 examples, obtaining a precision of 0.747, a recall of 0.852 and an f1 score of 0.796.

False Positive
$\langle car1 - \textbf{near} - person0\rangle$

True Positive
$\langle person1 - \textbf{near} - person2\rangle$

True Negative
$\langle boat4 - \textbf{near} - person3\rangle$

Figure 4.9: *Near* prediction samples

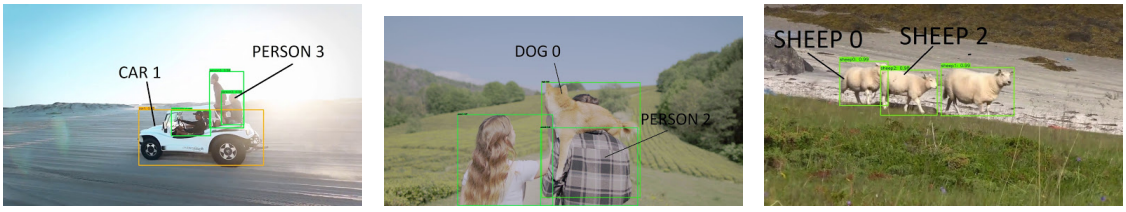
The most important features for the model to use when making *near* predictions are the language features and the F23 (object centroid x-axis position relative to subject centroid x-axis position as a euclidean unit vector), F22 (object centroid position relative to subject centroid position as a vector) and F13 (distance to size ratio between bounding boxes) geometrical features that our model did not include.

Using a model without language features gives negative results. The incorrect

$\langle car1 - \textbf{near} - person0 \rangle$ prediction's confidence score increased from 0.669 to 0.782 while the correct $\langle person1 - \textbf{near} - person2 \rangle$ prediction's confidence score is reduced slightly.

An alternative model was trained with the introduction of the F23 and F22 geometrical features in an attempt to filter out similar incorrect predictions to $\langle car1 - \textbf{near} - person0 \rangle$ where the object and subject centroids are too close together. This model gave a precision of 0.755, a recall of 0.839 and an f1 score of 0.795 and succeeded in reducing the $\langle car1 - \textbf{near} - person0 \rangle$ confidence score from 0.669 to 0.516 while also leaving the correct $\langle person1 - \textbf{near} - person2 \rangle$ and $\langle boat4 - \textbf{near} - person3 \rangle$ predictions almost unchanged. This shows that the model with the added F23 and F22 features is more suitable for accurately identifying *near* relationships.

Next To: The model was trained on 695 next to training examples and achieved a precision score of 0.679, a recall score of 0.659 and an f1 score of 0.669.



False Positive
$\langle car1 - \textbf{next to} - person3 \rangle$

False Positive
$\langle dog0 - \textbf{next to} - person2 \rangle$

True Positive
$\langle sheep2 - \textbf{next to} - sheep0 \rangle$

Figure 4.10: *Next To* prediction samples

The most important features for the model to make *next to* predictions are F3 (area of object normalized by the union of bounding boxes), F18 (ratio between subject top edge - object top edge to object bottom edge - object top edge) and F15 (euclidean distance between bounding boxes normalized by image).

Our initial model did not contain F15 but used F14 (euclidean distance between bounding boxes normalized by the union of bounding boxes) instead. The F14 feature seemed to hold too much importance since predictions were made based on

this feature alone regardless of the F3 and F18 values which are equally important when predicting this preposition. Therefore, a new model was trained with F15 replacing F14 in the hope that this would remove the strong bias towards the F14 feature. This model gave similar results and every example with a high confidence score for *next to* had similar F15 values of between -0.55 to -0.57 regardless of their F3 and F18 values, resulting in many incorrect predictions. A possible way of filtering out incorrect examples would be to remove any predictions with a high F18 values, (such as the incorrect $\langle car1 - \mathbf{near} - person3\rangle$ and $\langle dog0 - \mathbf{near} - person2\rangle$ predictions) since this would mean that their vertical positions are too different for them to be considered as *next to* each other.

On: The model was trained on 128 examples of on and obtained a precision score of 0.857, a recall score of 0.703 and an f1 score of 0.773.

False Positive $\langle person2 - \mathbf{on} - train1\rangle$

True Positive $\langle person1 - \mathbf{on} - bicycle2\rangle$

True Positive $\langle person0 - \mathbf{on} - motorbike2\rangle$

Figure 4.11: *On* prediction samples

The most important features for the model when predicting *on* are F11 (distance between bounding box centroids normalized by the union of bounding boxes), F30 (object centroid y-axis position relative to subject centroid y-axis position as a unit vector normalized by the union of bounding boxes) and the language features.

The effect of the language features can be seen in the $\langle person2 - \mathbf{on} - train1\rangle$ ex-

ample. Its F11 and F30 values were different to the correct $\langle person1 - \mathbf{on} - bicycle2 \rangle$ and $\langle person0 - \mathbf{on} - motorbike2 \rangle$ predictions, -0.13 and -1.8 compared to -1.38 and -0.002, and -1.92 and 0.36 respectively. Using an alternative model without the language features reduced $\langle person2 - \mathbf{on} - train1 \rangle$'s confidence score from 0.655 to almost 0. Despite this, the confidence scores of the correct $\langle person1 - \mathbf{on} - bicycle2 \rangle$ and $\langle person0 - \mathbf{on} - motorbike2 \rangle$ predictions were also reduced to low values, showing that the language features were still required. More varied training examples are required to lower the bias introduced by the language features. Incorrect examples could be filtered out by removing positive examples with unsuitable F11 and F30 values.

Under: The model was trained on 158 examples and gave a precision score of 0.796, a recall score of 0.715 and an f1 score of 0.753.

False Positive
$\langle bird0 - \mathbf{under} - bird1 \rangle$

False Positive
$\langle person2 - \mathbf{under} - person0 \rangle$

True Positive
$\langle bicycle1 - \mathbf{under} - person3 \rangle$

Figure 4.12: *Under* prediction samples

The most important features for the model to make under predictions are language features, depth features (which we do not have access to), F1 (area of subject normalized by image size), F11 (distance between bounding box centroids normalized by the union of bounding boxes) and F30 (object centroid y-axis position relative to subject centroid y-axis position as a unit vector normalized by the union of bounding boxes).

In our examples, 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 these features, lower F30 values led to higher *under* confidence scores while F1 and F11 values did not seem to have an effect. 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.

4.2.3 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 (see Figure A.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 full results are shown in Table 4.2.

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 evalua-

tion. 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	**Added Examples**	**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	**Added Examples**	**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	**Added Examples**	**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 4.2: Effect of Adding Additional Examples for the Under-Represented Prepositions

5. Closure

5.1 Future Work

Object tracking between video frames could also be incorporated. In this way, the spatial relationships between these objects could also be tracked over a number of frames of the video. A majority vote could then be used to predict the preposition that is best suited to describe the spatial relationship between each pair of objects over the short or long term of the video.

Another future improvement to the system would be the use of the *depth* and *pose* features offered by SpatialVOC2K. This would improve the system's accuracy when making predictions on prepositions like ***opposite***, ***behind*** and ***in front of*** [5], but would require training specialised models to recognise object *depth* and *pose* from unseen video frames.

The ideas explored in this project could be applied in future computer vision research that requires the creation of custom datasets or the enrichment of existing ones. Additionally, if a high-powered GPU is available, a similar system to the one that was created in this project could be used to detect spatial relationships between object pairs identified from a live video feed, allowing it to be used in real time applications such as robotics in the healthcare and maintenance industries, autonomous vehicles and in aids for the visually impaired.

5.2 Conclusion

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. A systematic way to clean the outputted set of examples for each preposition was therefore discussed in Section 4.2.2 in an effort to improve the Spatial Relationship model. Two possible solutions were deduced. 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 in its applications, in particular in dataset enrichment which we described in Section 4.2.3. 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 underrepresented *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.

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.

A. Appendix

A.1 Supplemental Dataset Material

1	aeroplane	11	dining table
2	bicycle	12	dog
3	bird	13	horse
4	boat	14	motorbike
5	bottle	15	person
6	bus	16	potted plant
7	car	17	sheep
8	cat	18	sofa
9	chair	19	train
10	cow	20	tvmonitor

Table A.1: Full list of VOC object classes

1	person	21	elephant	41	wine glass	61	dining table
2	bicycle	22	bear	42	cup	62	toilet
3	car	23	zebra	43	fork	63	tv
4	motorcycle	24	giraffe	44	knife	64	laptop
5	airplane	25	backpack	45	spoon	65	mouse
6	bus	26	umbrella	46	bowl	66	remote
7	train	27	handbag	47	banana	67	keyboard
8	truck	28	tie	48	apple	68	cell phone
9	boat	29	suitcase	49	sandwich	69	microwave
10	traffic light	30	frisbee	50	orange	70	oven
11	fire hydrant	31	skis	51	broccoli	71	toaster
12	stop sign	32	snowboard	52	carrot	72	sink
13	parking meter	33	sports ball	53	hot dog	73	refrigerator
14	bench	34	kite	54	pizza	74	book
15	bird	35	baseball bat	55	donut	75	clock
16	cat	36	baseball glove	56	cake	76	vase
17	dog	37	skateboard	57	chair	77	scissors
18	horse	38	surfboard	58	couch	78	teddy bear
19	sheep	39	tennis racket	59	potted plant	79	hair drier
20	cow	40	bottle	60	bed	80	toothbrush

Table A.2: Full list of COCO object classes with VOC classes highlighted

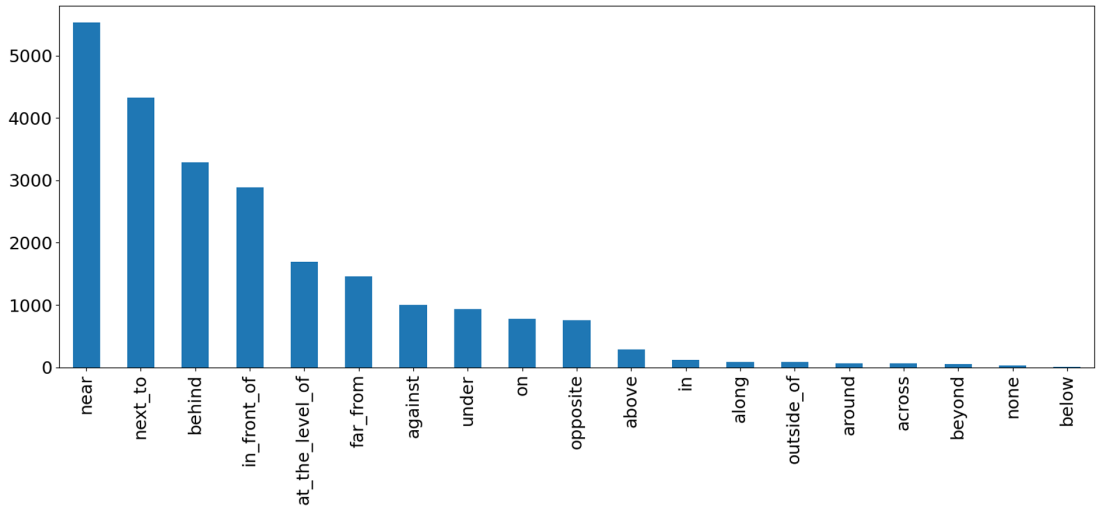


Figure A.1: Amount of times that each preposition is listed as being suitable to describe the spatial relationship between an object pair in the SpatialVOC2K dataset

a cote a cote de a cot de	next to
a l'exterieur de	outside of
au dessous de	below
au dessus de	above
au niveau de	at the level of
aucun	none
autor de	around
contre	against
dans	in
derriere	behind
devant	in front of
en face de	opposite
en travers de	across
le long de	along
loin de loin	far from
par dela	beyond
pres pres de	near
sous	under
sur	on

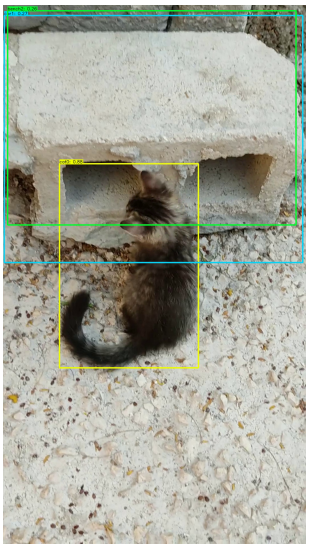
Table A.3: Translation of each SpatialVOC2K preposition from French to English

A.2 Full Geometrical Feature List

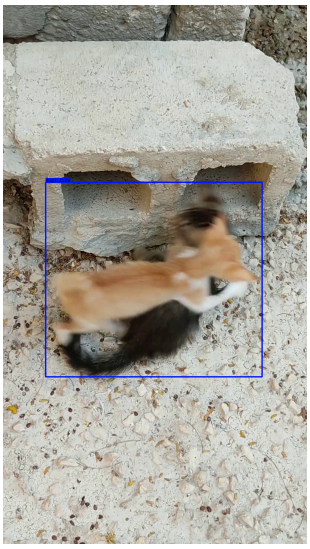
1	AreaObj1_Norm_wt_Image	21	AreaOverlap_Norm_wt_Min
2	AreaObj2_Norm_wt_Image	22	DistanceSizeRatio
3	AreaObj1_Norm_wt_Union	23	relativePosition
4	AreaObj2_Norm_wt_Union	24	InvFeatXminXmin
5	objAreaRatioTrajLand	25	InvFeatXmaxXmin
6	objAreaRatioMaxMin	26	InvFeatYminYmin
7	UnionDiag_Norm_wt_ImageDiag	27	InvFeatYmaxYmin
8	Obj_1Diag_Norm_wt_ImageDiag	28	AspectRatioObj_1
9	Obj_2Diag_Norm_wt_ImageDiag	29	AspectRatioObj_2
10	ImageDiag_Norm_wt_UnionDiag	30	unitEuclideanVector_x
11	Obj_1Diag_Norm_wt_UnionDiag	31	unitEuclideanVector_y
12	Obj_2Diag_Norm_wt_UnionDiag	32	EuclDist_norm_wt_ImageBB
13	objDiagonalRatioTL	33	EuclDist_Norm_wt_UnionBB
14	objDiagonalRatioMaxMin	34	vecTrajLand_Norm_wt_UnionBB_x
15	DistBtCentr_Norm_wt_ImageDiag	35	vecTrajLand_Norm_wt_UnionBB_y
16	DistBtCentr_Norm_wt_UnionDiag	36	unitVecTrajLand_Norm_wt_UnionBB_x
17	DistBtCentr_Norm_wt_UnionBB	37	unitVecTrajLand_Norm_wt_UnionBB_y
18	AreaOverlap_Norm_wt_Image	38	unitVecTrajLand_x
19	AreaOverlap_Norm_wt_Union	39	unitVecTrajLand_y
20	AreaOverlap_Norm_wt_Total		

Table A.4: Full list of SpatialVOC2K geometrical features with features that are also in the partial list highlighted

A.3 Additional Diagrams for Object Detection



The cat is correctly detected with confidence 0.88 while the brick is identified as both a bench and a car with low confidences of 0.26 and 0.27



Two cats identified as a single dog with confidence 0.29

Figure A.2: Examples of low confidence object detections

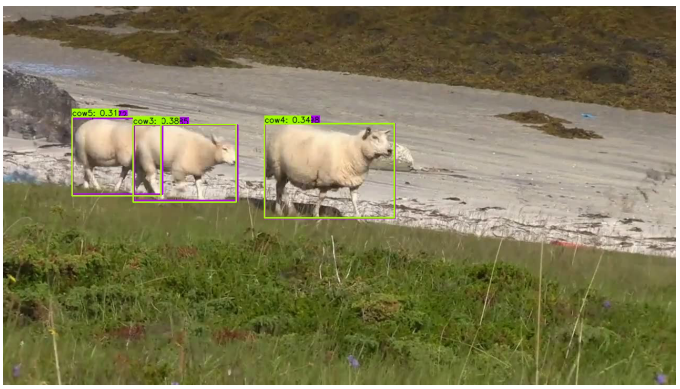


Figure A.3: Example of a double detection. The sheep are correctly detected as sheep (in purple) with confidence scores higher than 0.5. They are also detected as cows (in green) with confidence scores lower than 0.5. The incorrect cow predictions will be removed.

A.4 Algorithms

Algorithm 1: Composing the full list of features. (x) means that the task is only performed when training model (x)

Data: List of geometrical features between object pairs, list of corresponding object/subject classes

1 Load list of geometrical features
2 Binarize list of prepositions
3 Binarize list of object/subject classes (2,5)
4 Load word embeddings for each object/subject class (3,4,7,8)
5 Make full list of features by joining the geometrical and language features

Algorithm 2: Spatial Relationship Models Training

Data: full feature list, corresponding list of acceptable prepositions

1 Load full list of features
2 Binarize list of prepositions
3 Split list of features into train/test set
4 Normalise features using the saved scalar model
5 Create MLPClassifier instance using the selected hyper-parameters
6 Train the classifier on the training+validation set
7 Make predictions on the test set
8 Calculate the model's overall and label based performance scores.

Algorithm 3: Testing Model on VRD

Data: Filtered VRD dataset, multi-label spatial relationship classifier

1 Binarize list of correct prepositions from VRD dataset
2 Generate full list of features from VRD dataset using Algorithm 1 for model 7
3 Normalize features using the corresponding saved scaler for model 7
4 Make predictions from the list of normalized features
5 Calculate total model score:
6 **If** any of VRD's list of prepositions for an object pair is given a score of 0.5 or above by the model: mark as correct.
7 **Else**: mark as incorrect
8 **Return** correct/total

Algorithm 4: Running the Model on the Unlabelled videos dataset

```
Data: Unlabelled Videos dataset, multi-label spatial relationship classifier
for each video in the dataset do
    for each frame of the video (at 10FPS) do
        Detect objects in the frame and filter out low confidence detections
          and double detections
        if less than 2 detected objects then
            Continue to next frame
        end
        Save frame marked with bounding boxes and object classes
        Generate features as in Algorithm 1
        Normalize features using the saved scaler for model 7
        Make predictions from the list of normalized features
        Add predictions to dictionary
    end
end
Save dictionary
(optional) Optimise frames using Algorithm 5
```

Algorithm 5: Optimising the Predictions

```
Data: List of predictions of the multi-label spatial relationship classifier
      on the Unlabelled Videos dataset
i = 1;
while i < number of frames in dataset do
    new_key_frame = False;
    key_frame = frame i;
    prepositions_1 = list of prepositions in key_frame with confidence over
     0.5;
    frame_set = [ ];
    while new_key_frame is False do
        prepositions_2 = list of prepositions in frame i+1 with confidence
         over 0.5;
        if prepositions_1 == prepositions_2 then
            add frame i+1 to frame_set;
        end
        else
            delete all frames in frame_set except the frame that has the best
             average preposition confidence score;
            new_key_frame = True;
            break;
        end
        i++;
    end
end
```

A.5 Spatial Relationship Model Training Full Results

Model 1: Overall Accuracy 0.449629522361402

Preposition	Instances	Precision	Recall	F1
above	53	0.4285714285714	0.0566037735849	0.1000000000000
across	9	nan	0.0000000000000	0.0000000000000
against	193	0.5500000000000	0.1139896373056	0.1888412017167
along	18	nan	0.0000000000000	0.0000000000000
around	12	0.6000000000000	0.2500000000000	0.3529411764705
at the level of	316	0.4827586206896	0.3544303797468	0.4087591240875
behind	504	0.6085918854415	0.5059523809523	0.5525460455037
beyond	1	nan	0.0000000000000	0.0000000000000
below	0	nan	0.0000000000000	0.0000000000000
far from	242	0.5628140703517	0.4628099173553	0.5079365079365
in	21	0.6666666666666	0.1904761904761	0.2962962962962
in front of	477	0.6554307116104	0.3668763102725	0.4704301075268
near	1048	0.7232219365895	0.8053435114503	0.7620767494356
next to	695	0.6542207792207	0.5798561151079	0.6147978642257
none	6	nan	0.0000000000000	0.0000000000000
on	128	0.7200000000000	0.5625000000000	0.6315789473684
opposite	121	nan	0.0000000000000	0.0000000000000
outside of	15	0.0000000000000	0.0000000000000	0.0000000000000
under	158	0.6014492753623	0.5253164556962	0.5608108108108
Overall		**0.64828897338**	**0.52029058962**	**0.57727973838**

Model 2: Overall Accuracy 0.50321844906234

Preposition	Instances	Precision	Recall	F1
above	53	0.2857142857142	0.0754716981132	0.1194029850746
across	9	0.0000000000000	0.0000000000000	0.0000000000000
against	193	0.3818181818181	0.2176165803108	0.2772277227722
along	18	0.2857142857142	0.1111111111111	0.1600000000000
around	12	0.6666666666666	0.3333333333333	0.4444444444444
at the level of	316	0.5128205128205	0.2531645569620	0.3389830508474
behind	504	0.6971046770601	0.6210317460317	0.6568730325288
beyond	1	nan	0.0000000000000	0.0000000000000
below	0	0.0000000000000	0.0000000000000	0.0000000000000
far from	242	0.6173913043478	0.5867768595041	0.6016949152542
in	21	0.5000000000000	0.4761904761904	0.4878048780487
in front of	477	0.6555555555555	0.4947589098532	0.5639187574671
near	1048	0.7486772486772	0.8101145038167	0.7781851512373
next to	695	0.6796747967479	0.6014388489208	0.6381679389312
none	6	nan	0.0000000000000	0.0000000000000
on	128	0.8846153846153	0.7187500000000	0.7931034482758
opposite	121	0.3666666666666	0.0909090909090	0.1456953642384
outside of	15	0.0000000000000	0.0000000000000	0.0000000000000
under	158	0.8203125000000	0.6645569620253	0.7342657342657
Overall		**0.69157706093**	**0.58786112519**	**0.63551529664**

Model 3: Overall Accuracy 0.51604156107450

Preposition	Instances	Precision	Recall	F1
above	53	0.4285714285714	0.1132075471698	0.1791044776119
across	9	nan	0.0000000000000	0.0000000000000
against	193	0.4920634920634	0.1606217616580	0.2421875000000
along	18	0.5000000000000	0.1111111111111	0.1818181818181
around	12	0.6250000000000	0.4166666666666	0.5000000000000
at the level of	316	0.5271739130434	0.3069620253164	0.3880000000000
behind	504	0.7220708446866	0.5257936507936	0.6084959816303
beyond	1	nan	0.0000000000000	0.0000000000000
below	0	0.0000000000000	0.0000000000000	0.0000000000000
far from	242	0.6744186046511	0.5991735537190	0.6345733041575
in	21	0.5789473684210	0.5238095238095	0.5500000000000
in front of	477	0.5807127882599	0.5807127882599	0.5807127882599
near	1048	0.7578397212543	0.8301526717557	0.7923497267759
next to	695	0.6569444444444	0.6805755395683	0.6685512367491
none	6	nan	0.0000000000000	0.0000000000000
on	128	0.8454545454545	0.7265625000000	0.7815126050420
opposite	121	0.3548387096774	0.0909090909090	0.1447368421052
outside of	15	0.1818181818181	0.1333333333333	0.1538461538461
under	158	0.8442622950819	0.6518987341772	0.7357142857142
Overall		**0.69392882689**	**0.60547389761**	**0.64669064265**

Model 4: Overall Accuracy 0.498842709917215

Preposition	Instances	Precision	Recall	F1
above	53	0.3529411764705	0.1132075471698	0.1714285714285
across	9	0.0000000000000	0.0000000000000	0.0000000000000
against	193	0.4117647058823	0.1813471502590	0.2517985611510
along	18	0.0000000000000	0.0000000000000	0.0000000000000
around	12	0.6250000000000	0.4166666666666	0.5000000000000
at the level of	316	0.5070921985815	0.4525316455696	0.4782608695652
behind	504	0.6812933025404	0.5853174603174	0.6296691568836
beyond	1	nan	0.0000000000000	0.0000000000000
below	0	0.0000000000000	0.0000000000000	0.0000000000000
far from	242	0.7015706806282	0.5537190082644	0.6189376443418
in	21	0.5882352941176	0.4761904761904	0.5263157894736
in front of	477	0.6312684365781	0.4486373165618	0.5245098039215
near	1048	0.7714561234329	0.7633587786259	0.7673860911270
next to	695	0.6845528455284	0.6057553956834	0.6427480916030
none	6	0.0000000000000	0.0000000000000	0.0000000000000
on	128	0.8691588785046	0.7265625000000	0.7914893617021
opposite	121	0.3750000000000	0.0743801652892	0.1241379310344
outside of	15	0.4000000000000	0.1333333333333	0.2000000000000
under	158	0.7800000000000	0.7405063291139	0.7597402597402
Overall		**0.69714337700**	**0.58026693698**	**0.63335836196**

Model 5: Overall Accuracy 0.4515458692346

Preposition	Instances	Precision	Recall	F1
above	53	0.7500000000000	0.0566037735849	0.1052631578947
across	9	nan	0.0000000000000	0.0000000000000
against	193	0.3846153846153	0.0777202072538	0.1293103448275
along	18	nan	0.0000000000000	0.0000000000000
around	12	0.8000000000000	0.3333333333333	0.4705882352941
at the level of	316	0.4822335025380	0.3006329113924	0.3703703703703
behind	504	0.6695652173913	0.4583333333333	0.5441696113074
beyond	1	nan	0.0000000000000	0.0000000000000
below	0	1.0000000000000	0.1000000000000	0.1818181818181
far from	242	0.6125000000000	0.4049586776859	0.4875621890547
in	21	0.8000000000000	0.1904761904761	0.3076923076923
in front of	477	0.6445182724252	0.4067085953878	0.4987146529562
near	1048	0.7098614506927	0.8311068702290	0.7657142857142
next to	695	0.6482558139534	0.6417266187050	0.6449746926970
none	6	nan	0.0000000000000	0.0000000000000
on	128	0.6764705882352	0.5390625000000	0.6000000000000
opposite	121	nan	0.0000000000000	0.0000000000000
outside of	15	0.5000000000000	0.0666666666666	0.1176470588235
under	158	0.6829268292682	0.5316455696202	0.5978647686832
Overall		**0.64957967836**	**0.5235681702**	**0.57980627770**

Model 6: Overall Accuracy 0.51735935124176

Preposition	Instances	Precision	Recall	F1
above	53	0.4000000000000	0.1132075471698	0.1764705882352
across	9	nan	0.0000000000000	0.0000000000000
against	193	0.4427480916030	0.3005181347150	0.3580246913580
along	18	0.2857142857142	0.1111111111111	0.1600000000000
around	12	0.8333333333333	0.4166666666666	0.5555555555555
at the level of	316	0.4749034749034	0.3892405063291	0.4278260869565
behind	504	0.6694214876033	0.6428571428571	0.6558704453441
beyond	1	nan	0.0000000000000	0.0000000000000
below	0	0.0000000000000	0.0000000000000	0.0000000000000
far from	242	0.6371681415929	0.5950413223140	0.6153846153846
in	21	0.6470588235294	0.5238095238095	0.5789473684210
in front of	477	0.6481481481481	0.5136268343815	0.5730994152046
near	1048	0.7635869565217	0.8043893129770	0.7834572490706
next to	695	0.6671686746987	0.6374100719424	0.6519499632082
none	6	nan	0.0000000000000	0.0000000000000
on	128	0.8737864077669	0.7031250000000	0.7792207792207
opposite	121	0.6000000000000	0.0495867768595	0.0916030534351
outside of	15	0.2857142857142	0.1333333333333	0.1818181818181
under	158	0.8195488721804	0.6898734177215	0.7491408934707
Overall		**0.68971061093**	**0.61229092752**	**0.64869900259**

Model 7: Overall Accuracy 0.5270991721574

Preposition	Instances	Precision	Recall	F1
above	53	0.6666666666666	0.1509433962264	0.2461538461538
across	9	nan	0.0000000000000	0.0000000000000
against	193	0.4750000000000	0.1968911917098	0.2783882783882
along	18	0.2500000000000	0.0555555555555	0.0909090909090
around	12	0.6666666666666	0.3333333333333	0.4444444444444
at the level of	316	0.5168539325842	0.4367088607594	0.4734133790737
behind	504	0.7132530120481	0.5873015873015	0.6441784548422
beyond	1	nan	0.0000000000000	0.0000000000000
below	0	nan	0.0000000000000	0.0000000000000
far from	242	0.6995073891625	0.5867768595041	0.6382022471910
in	21	0.4782608695652	0.5238095238095	0.5000000000000
in front of	477	0.7031250000000	0.4716981132075	0.5646173149309
near	1048	0.7472803347280	0.8520992366412	0.7962550156041
next to	695	0.6785185185185	0.6589928057553	0.6686131386861
none	6	nan	0.0000000000000	0.0000000000000
on	128	0.8571428571428	0.7031250000000	0.7725321888412
opposite	121	0.4375000000000	0.0578512396694	0.1021897810218
outside of	15	0.1428571428571	0.0666666666666	0.0909090909090
under	158	0.7957746478873	0.7151898734177	0.7533333333333
Overall		**0.70742027436**	**0.61126034803**	**0.65583425701**

Model 8: Overall Accuracy 0.5160837979388

Preposition	Instances	Precision	Recall	F1
above	53	0.4285714285714	0.0566037735849	0.1000000000000
across	9	nan	0.0000000000000	0.0000000000000
against	193	0.4347826086956	0.2072538860103	0.2807017543859
along	18	0.0000000000000	0.0000000000000	0.0000000000000
around	12	0.7142857142857	0.4166666666666	0.5263157894736
at the level of	316	0.4721311475409	0.4556962025316	0.4637681159420
behind	504	0.7055837563451	0.5515873015873	0.6191536748329
beyond	1	nan	0.0000000000000	0.0000000000000
below	0	0.0000000000000	0.0000000000000	0.0000000000000
far from	242	0.6761904761904	0.5867768595041	0.6283185840707
in	21	0.5454545454545	0.5714285714285	0.5581395348837
in front of	477	0.6864686468646	0.4360587002096	0.5333333333333
near	1048	0.7453468697123	0.8406488549618	0.7901345291479
next to	695	0.6611111111111	0.6848920863309	0.6727915194346
none	6	nan	0.0000000000000	0.0000000000000
on	128	0.8846153846153	0.7187500000000	0.7931034482758
opposite	121	0.3809523809523	0.0661157024793	0.1126760563380
outside of	15	0.1764705882352	0.2000000000000	0.1875000000000
under	158	0.8244274809160	0.6835443037974	0.7474048442906
Overall		**0.69596284387**	**0.60316776482**	**0.64625119311**

Table A.5: Full scores for all models

A.6 Example Prediction

```
person 0 [[6.14496615e-03 4.83749497e-04 2.36012564e-02 4.57818020e-03
  1.03781006e-04 5.86087535e-02 5.82782131e-01 7.95007053e-07
  1.64102378e-02 1.69185042e-01 3.09215509e-03 1.17862589e-01
  5.70808725e-01 2.64078801e-01 1.33055574e-03 7.13852761e-03
  5.97464260e-02 1.12114087e-02 1.07688461e-03]] car 1
person 0 [[1.93644299e-01 2.39123069e-03 2.99918388e-04 1.38573651e-03
  8.09584103e-06 5.37510162e-02 3.49818086e-01 1.25328356e-03
  2.43085805e-03 3.46466495e-02 7.39682442e-06 1.48233495e-01
  8.19384729e-01 3.31822472e-01 2.97968669e-03 2.55928462e-03
  1.16428352e-01 1.03220944e-03 2.44278833e-03]] person 2
car 1 [[4.08089519e-02 2.40531648e-03 2.37814000e-03 2.93075353e-04
  3.09318557e-03 6.37619698e-02 4.56279075e-02 8.72868745e-04
  6.36859017e-05 1.16693627e-01 7.22858496e-06 5.97863290e-01
  4.92002730e-01 2.46902397e-01 1.78896697e-04 2.39954904e-04
  6.64269090e-02 1.72307201e-03 4.92882755e-04]] person 0
car 1 [[8.01229879e-02 1.83999445e-03 3.40097432e-03 1.89272344e-05
  9.69530746e-03 4.28934386e-02 1.94432427e-01 7.06642596e-04
  7.68588565e-05 2.61094131e-02 3.28727318e-06 3.20000954e-01
  6.06227723e-01 2.97292522e-01 1.53297352e-04 5.34006289e-04
  2.51504854e-02 7.89160960e-04 8.74731932e-04]] person 2
person 2 [[8.66233279e-03 4.23443674e-03 1.55516647e-02 5.78491510e-03
  1.67850603e-05 2.33577024e-02 4.50953769e-02 5.94890281e-04
  2.74255500e-04 9.60651471e-03 4.14703503e-06 6.47273122e-01
  8.61207653e-01 3.55391288e-01 4.46128362e-03 6.18442730e-05
  9.24772988e-02 1.31783753e-03 5.03406201e-01]] person 0
person 2 [[7.98676537e-04 1.66029073e-04 2.37543184e-02 2.13463389e-03
  4.18748987e-07 5.30856743e-02 7.69203633e-01 3.74221638e-08
  3.08030106e-02 1.05009912e-01 3.75803811e-04 1.32992863e-01
  7.44024537e-01 4.33489151e-01 2.25415945e-03 1.85818818e-03
  1.92102735e-02 2.22223917e-02 9.07280223e-03]] car 1
```

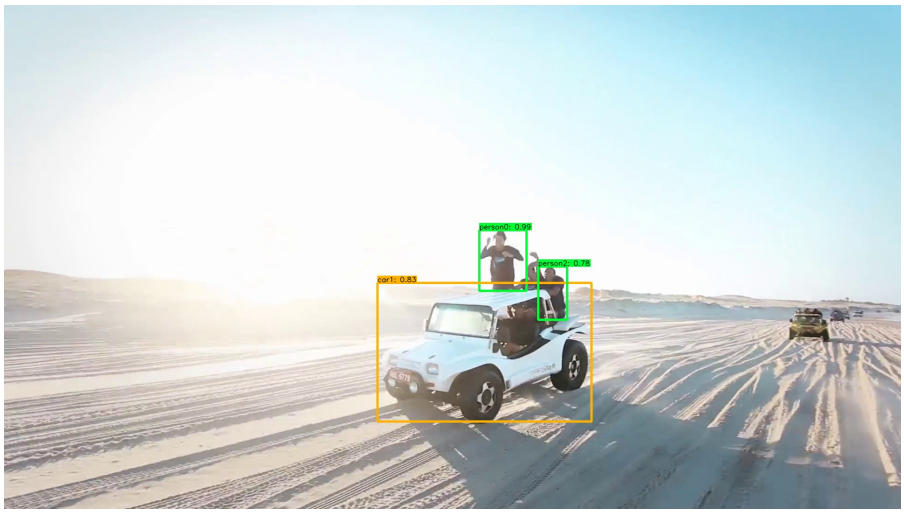


Figure A.4: Example of a prediction outputted by the system

A.7 Additional Data for the Unlabelled Video Dataset

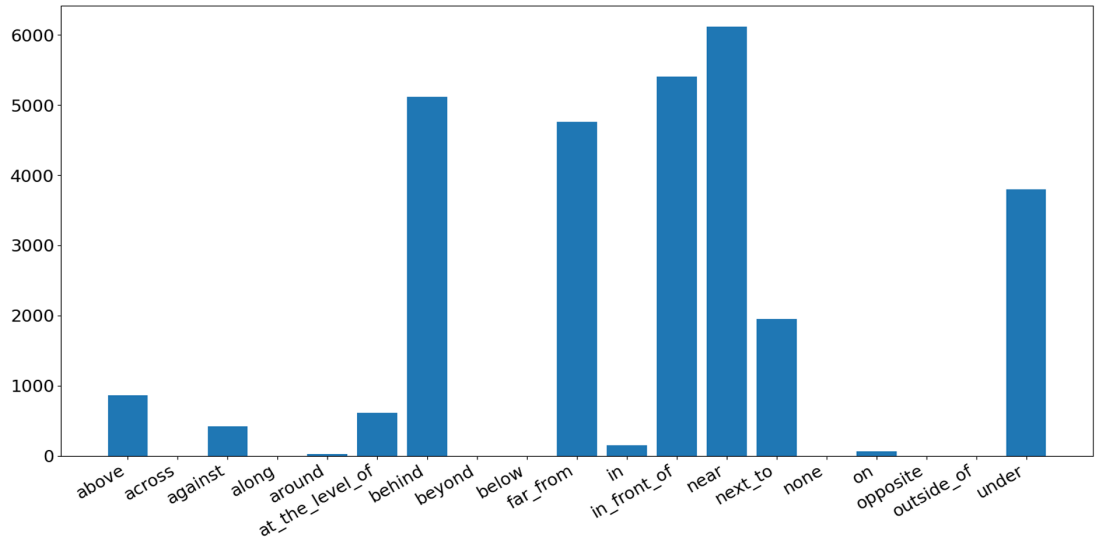
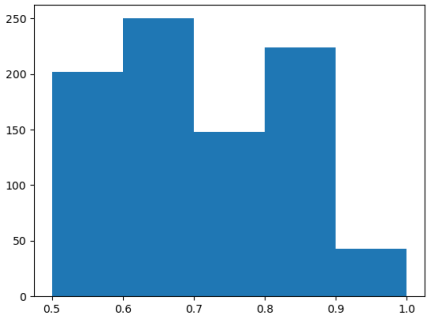
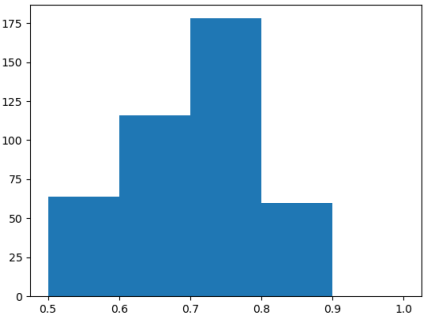


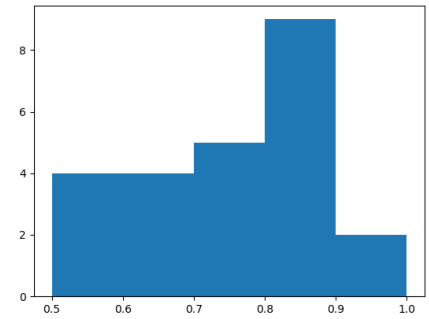
Figure A.5: Number of times each preposition is predicted



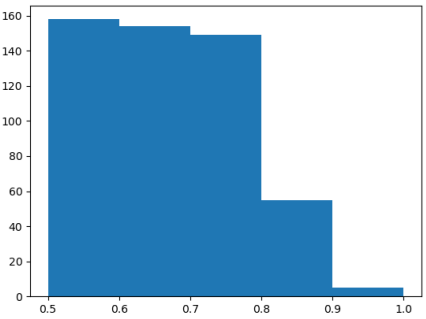
Above



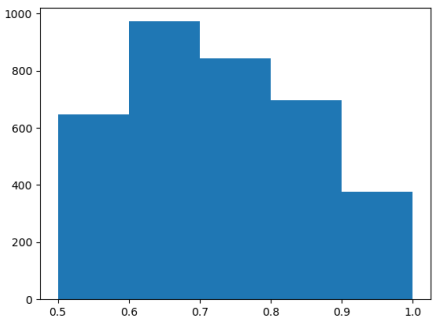
Against



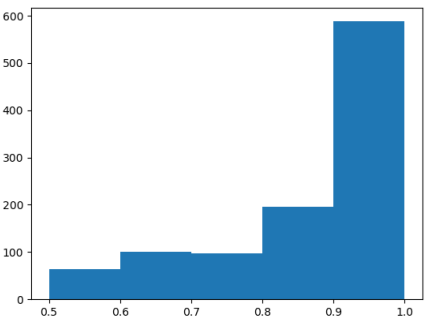
Around



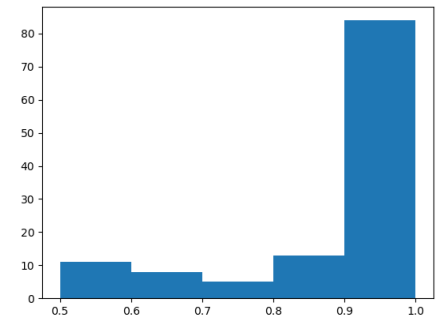
At The Level Of



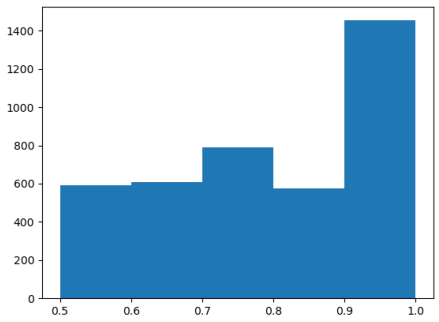
Behind



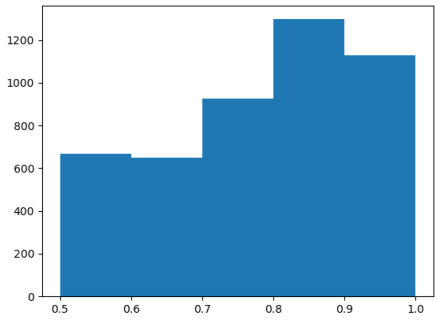
Far From



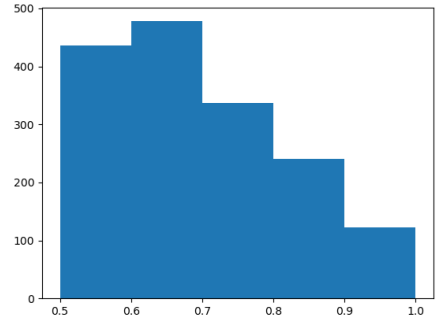
In



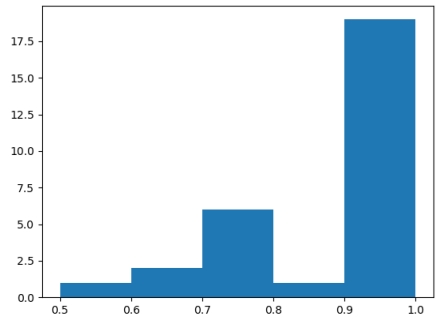
In Front Of



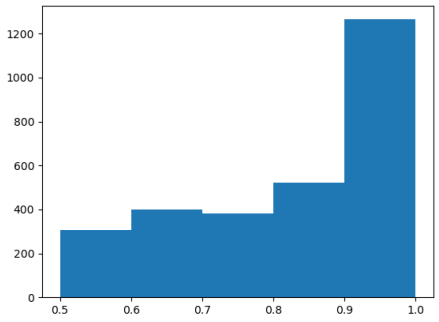
Near



Next To



On

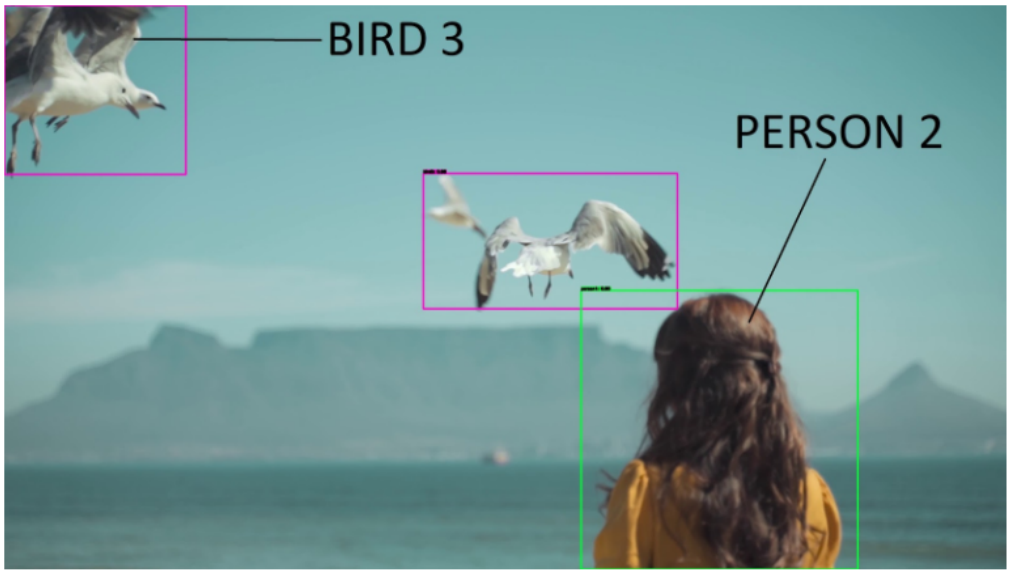


Under

Figure A.6: Distribution of confidence score of predictions on each preposition

A.8 Sample Questionnaire Question

bird3 ABOVE person2 *



- AGREE
- NOT SURE
- DISAGREE

Figure A.7: Sample question from questionnaire

A.9 Additional Data for the VRD Dataset

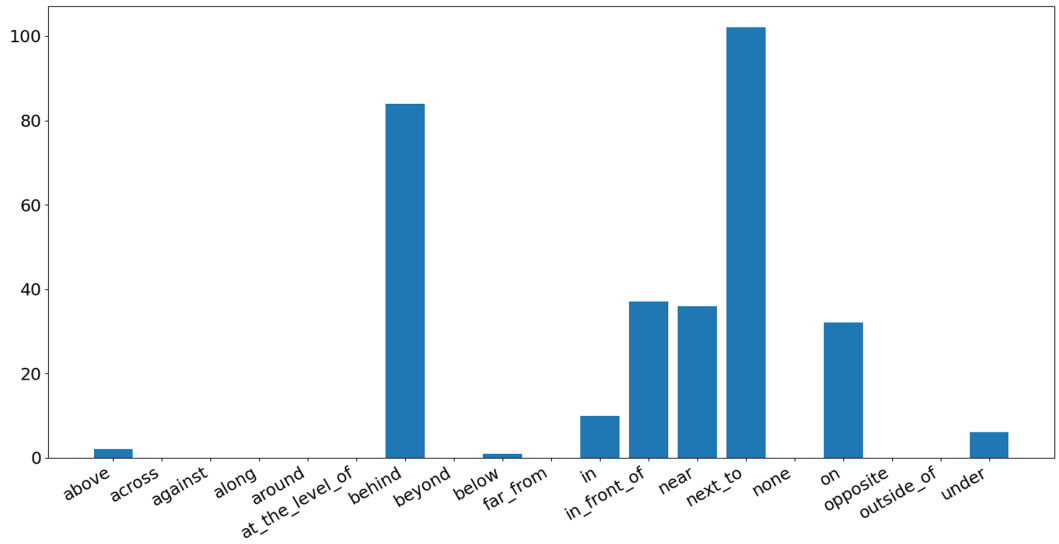


Figure A.8: Number of times each preposition is listed in VRD dataset

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 layer-wise relevance propagation. pages 378–385, 01 2020.

[6] R. B. Girshick. Fast R-CNN. *CoRR*, abs/1504.08083, 2015.

[7] R. B. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. *CoRR*, abs/1311.2524, 2013.

[8] N. Gkanatsios, V. Pitsikalis, P. Koutras, A. Zlatintsi, and P. Maragos. Deeply supervised multimodal attentional translation embeddings for visual relationship detection. In *2019 IEEE International Conference on Image Processing (ICIP)*, pages 1840–1844, 2019.

[9] 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.

[10] M. Haldekar, A. Ganesan, and T. Oates. Identifying spatial relations in images using convolutional neural networks. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pages 3593–3600, 2017.

[11] 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.

[12] T. Lin, M. Maire, S. J. Belongie, L. D. Bourdev, R. B. Girshick, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft COCO: common objects in context. *CoRR*, abs/1405.0312, 2014.

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

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

[15] 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.

[16] 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.

[17] 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.

[18] J. Redmon, S. K. Divvala, R. B. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. *CoRR*, abs/1506.02640, 2015.

[19] S. Ren, K. He, R. B. Girshick, and J. Sun. Faster R-CNN: towards real-time object detection with region proposal networks. *CoRR*, abs/1506.01497, 2015.

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

[21] X. Shang, T. Ren, J. Guo, H. Zhang, and T.-S. Chua. Video visual relation detection. In *Proceedings of the 25th ACM international conference on Multimedia*, pages 1300–1308, 2017.

[22] 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.

[23] Y. Tang, J. Wang, B. Gao, E. Dellandréa, R. Gaizauskas, and L. Chen. Large scale semi-supervised object detection using visual and semantic knowledge transfer. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2119–2128, 2016.

[24] J. Wang and R. Gaizauskas. Don't mention the shoe! a learning to rank approach to content selection for image description generation. 06 2016.

[25] R. Yu, A. Li, V. I. Morariu, and L. S. Davis. Visual relationship detection with internal and external linguistic knowledge distillation. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 1068–1076, 2017.

[26] 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.

[27] Y. Zhu, S. Jiang, and X. Li. Visual relationship detection with object spatial distribution. In *2017 IEEE International Conference on Multimedia and Expo (ICME)*, pages 379–384, 2017.