

---

# EECS 545 Final Report:

## Self Supervised Object Detection With Multimodal Image Captioning

---

xxx xxxxxxxxx, xxxxxx xxxxxx, Kemmannu Vineet Rao, xxxxxxx xxxxxx xxxxx, xxx xxx xxx  
{xxxxxxx,xxxxxxx,kemmannu,xxxxxx,xxx}@umich.edu

### 1 Abstract

We worked on the problem of self-supervised object detection utilizing multimodal data representation learning. We propose a novel self supervised pipeline that uses natural language supervision as a pre-training task to localize objects given an image. Further, we integrate a noise robust object detector to detect an object in noisy box settings.

*Code is available at: [here](#).*

### 2 Significance

Performance of object detection models trained on labeled datasets is heavily reliant on how the training data is represented which doesn't always generalize well to the test data and real-world examples. Reducing the dependency on labeled data ensures the model performs well in all scenarios.

Compared to standard image classification, the performance of automated object localization tasks is sub-standard compared to human performance. Progress is limited by the availability of accurately labeled data which requires substantial human effort. Conventional object detection models also require high levels of computation as they have to be trained with high resolution images and with large convolutional networks

Training a deep learning model with supervision from scratch requires a lot of training data. It alleviates the need for human labor in terms of annotations. It can also facilitate the learning process when the fine-grained annotation is extremely labor intensive and time consuming to even obtain the whole labeled data that is required by the fully-supervised approaches. We hope to reduce this heavy reliance on large amounts of human annotated training data by learning a multimodal representation pre-trained on Redcaps dataset and use the captions generated to artificially label the images in a self-supervised method. The major advantage is reducing the reliance on large amounts of training data and requires lesser computational power. Improving the performance of object localization could have far reaching benefits various fields such as autonomous vehicle control

### 3 Related Work

The Redcaps paper [2] uses image captioning on Reddit data as a pre-training task to learn good visual representations. Using these representations, the authors finetune models on several downstream datasets for various downstream tasks. They use an image captioning model - VirTex-v2 [1] pre-trained on the RedCaps dataset as a backbone for various downstream tasks such as fine-grained image classification, object detection and semantic segmentation. Our approach differs from this in that we are performing object detection, which outputs bounding boxes, and our approach involves training the full object detection model on pseudo-labels, rather than only pre-training the backbone CNN on a different task.

Desai *et al.*[1] introduced an image captioning model called VirTex. It is a pre-training approach to learn visual features via language supervision. The model consists of a visual backbone that extracts image features and a textual head that predicts captions via bidirectional language modelling. The Transformers then perform masked multiheaded self-attention over caption features, and multiheaded attention over image features and outputs an caption based on beam search and nucleus sampling See Figure 1.

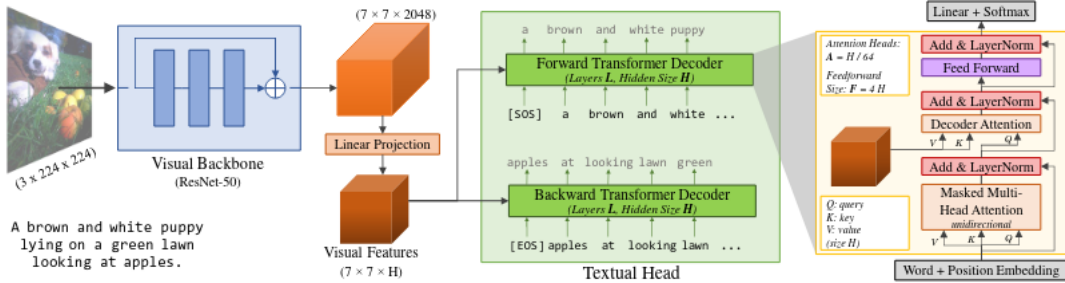


Figure 1: VirTex model architecture [1] consists of a visual backbone (ResNet-50), and a textual head (two uni- directional Transformers). The visual backbone extracts image features, and textual head predicts captions via bidirectional language modeling (bicaptioning). The Transformers perform masked multiheaded self-attention over caption features, and multiheaded attention over image features.

Object detection is the task of locating objects within an image and drawing their bounding boxes. Traditional approaches involve training from scratch on big datasets of images and human annotated data. We hope our proposed approach can eliminate the need for much of this labeled data, with the bulk of the training done with unlabeled images.

FCOS [7] is a fully convolutional one-stage object detector. It is an anchor free and proposal free method. It reformulates object detection in a per-pixel prediction fashion. It predicts centerness which improves the quality of the bounding boxes. It also performs multilevel prediction using the feature pyramid network.

Recent work on unsupervised learning for object detection (DetCo [8], patch Re-ID [3]), typically proposes novel pre-training tasks that produce models which can then be fine-tuned on object detection datasets. Our approach is similar, however our pre-training task is more directly related to object detection as opposed to contrastive learning.

There has also been work on creating visual explanations [6], where heatmaps can be generated to approximately show which pixels in an image correspond to the predicted label by calculating the gradient of the cost function with respect to pixels of the input image. We hope to utilize a similar approach to determine which pixels in an image correspond to words in the caption. From this we can generate a bounding box around the highlighted pixels for each pseudo-label we obtain. In this way, we can perform object detection with minimal amount(1%-2%) of labelled data.

## 4 Proposed Method

We solved this problem using a novel pre-training approach which outputs image captions which we will treat as pseudo-labels. We then used the pseudo-labels to generate class activation maps for that specific query to spatially locate the corresponding object in the image. Finally, we used the generated location information to train the FCOS[7] object detection model without requiring any ground truth object annotations. We also modified the box loss function to use a smoothened L1 loss to account for noisy boxes.

For the first step, we would like to utilize the rich representations that come from multimodal training. In particular, we want a model that can perform image captioning. These captions can provide a lot of information about what objects might appear in the image. Fortunately, Desai *et al.*[2] have already trained a VirTex-V2 model on their large scale curated RedCaps dataset which we use to generate image captions.

We modified the VirTex-v2 model to output captions along with logits for each corresponding word in the caption (See Figure 6).

More specifically, a sub-prompt (sub-reddit) and a caption prompt are fed in as inputs along with the unlabelled image. We then consider the next word predictions which we can use for labelling. For our task, the sub-prompt we used was “I took a picture of ” and prompt “itap ” (See Figure 2). In the sub-reddit “I took a picture of ” users upload ordinary life images and captions describing them.

The specific method to generate bounding boxes for proposed captions begins after the VirTex model is given the prompt “I took a picture of :// ” “ITAP a/an ”. The model used in [2] generates captions one word at a time, so after giving the prompt as input, we can extract the logits for what the model thinks should come next. We then filter the most likely words above a certain logit threshold (probability of that word appearing after the prompt based on the image). Since we are looking for objects, we filter out just the nouns with parts-of-speech tagging and ignore everything else like adjectives. Lastly we use the Wordnet Dataset[5] to associate each generated noun with the class label that shares the most semantic similarity. This ensures words like girl get mapped to the person class, husky gets mapped to the dog class, etc.

By using a technique similar to GradCAM [6], we can then compute the gradient of the input image with respect to the words of interest. This provides a heatmap showing which pixels in the image correspond with each word. From this heatmap we can then create a pseudo-label bounding box. Using an intensity threshold, the maximum and minimum coordinates of pixels above the specified intensity are marked as the edges of the bounding box corresponding to that noun. As an example, if the caption contained the word cat; we would expect the heatmap generated from this word to be comprised of pixels of the cat. Thus our generated bounding box should roughly correspond to the ground truth bounding box of the cat. Finally, since many words correspond to the same object in an image, bounding boxes are merged through averaging if they share a class label and have an IoU greater than another specified threshold. These resulting boxes are then kept as the pseudo-labels for the data. We then experimented with different values of various thresholds mentioned above, type of GradCAM method used and using average or median to remove duplicate bounding boxes to get our best performing model. Using our best performing model, we generated our pseudo classes and pseudo bounding boxes and trained an object detection model with unlabeled data.

## 5 Evaluation

We first compare the bounding boxes generated by our novel training pipeline to the ground-truth labels of the PASCAL V07 dataset [4] to compare the yielded results. The metric that we use is generic mAP(mean average precision) which computes the average precision of the proposed bounding boxes by calculating the precision where a bounding box is considered correct when it has an IoU of greater than **50%** with the ground truth bounding box and the same associated class label.

We then compare how the FCOS model [7] trained with our self-supervised method performs against the same model trained from scratch with supervision by benchmarking the performance PASCAL V07 dataset [4] used in SOTA object detection tasks. We also compare the mAP achieved by our model with a weakly supervised object detection method that achieves a mAP of **24%**.

We hope to extrapolate the computation savings and stress that the model pre-trained on a multimodal dataset would perform better with minimal amount of fine-tuning. Additionally, we want to show that the model pre-trained on Redcaps specifically will perform better than the models pre-trained on the other datasets since the images in Reddit data gives us more linguistic diversity (we can create more object classes).

## 6 Experimental Results

Our first task was to download the redcaps image captioning model and get it to run locally. By default the model only returns the predicted caption string, however we also need the logits (log probability) for each word in order to use the GradCAM technique. We also passed a sub-prompt “I took a picture of ” and prompt “itap “. After modifying the code to return the logits for the predicted word that could appear after the sub-prompt and prompt, we noticed that the language tokens do not have a one to one correspondence to words in the caption. Some words are composed of several tokens. Thus we also needed to implement

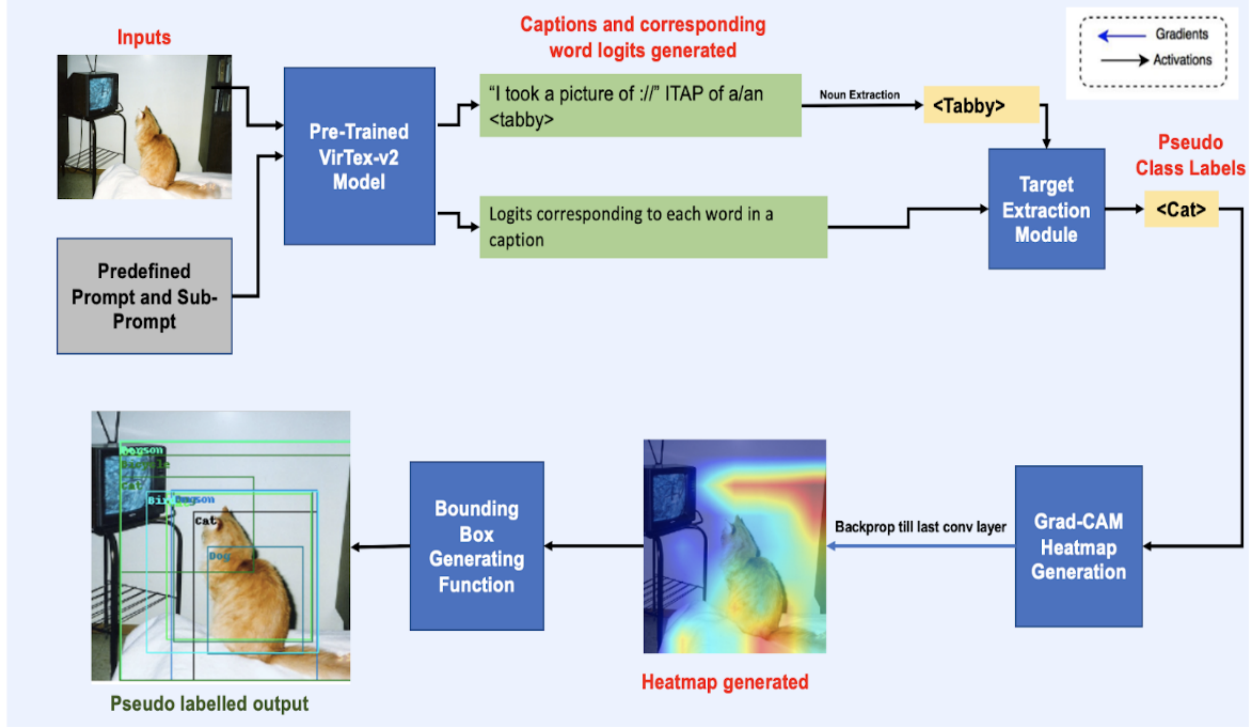


Figure 2: In our novel proposed method, we input an unlabelled image to the pretrained VirTex-v2 model to generate captions along with logits corresponding to each token in the caption. Using a noun extraction module, we will extract all the nouns from the caption. Now, using the nouns and their corresponding logits we pass it through a target extraction module to generate our pseudo class labels by comparing similarity between each word and the class labels of our downstream dataset. Using these pseudo class labels, we pass it through a class activation method namely GradCAM to generate heatmaps. Using these heatmaps, we draw bounding boxes on the image and obtain their respective coordinates. We also perform a post-processing step to discard redundant bounding boxes with the same class label according to the logit score threshold.

a function that converts from tokens to words or word parts. With these steps completed we were able to predict a caption string and extract the logit values corresponding to each word or word part. We introduced a logit threshold here to limit our logit list to high probability values. The redcaps image captioning model

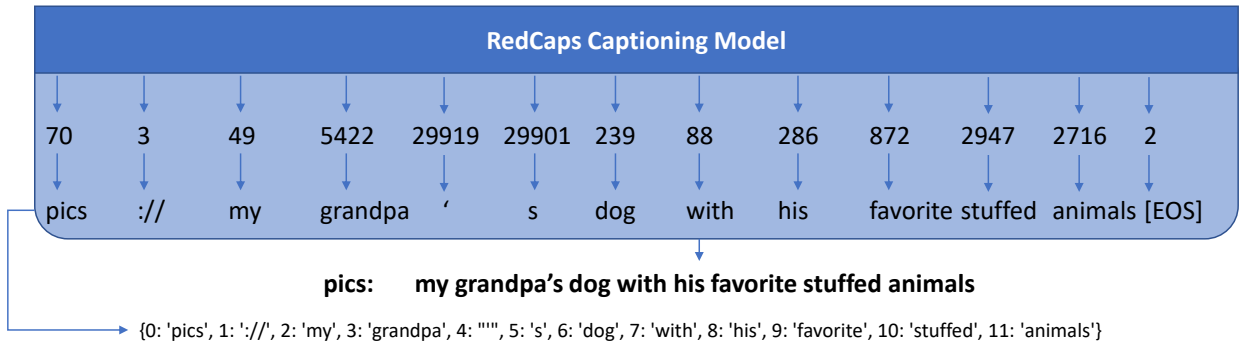


Figure 3: We modify the RedCaps Captioning Model to return the internal prediction logits, as well as a mapping from word parts to tokens.

provided us with  $n$  logits for each image. To generate heatmaps and bounding-boxes using the gradient CAM method, we must provide a target concept. We chose to extract nouns from the generated logits by using a prompt "i took a picture of a " followed by the logit. The Parts of Speech (POS) tagging then extracted the nouns from the list of logits using the averaged perceptron tagger from the nltk package. We then used the WordNet database [5] to generate semantic similarity scores between the chosen nouns and the target classes from the image dataset. We introduced another threshold here to discard pseudo class labels which did not have a minimum similarity score with the target classes. To view the results of this experiment for two different images , refer Figure: 4



Example Experimental Target Extraction				
Input Image	Thresh	Logits	Nouns	Target(n)
	Logit: -7.13 Sim: 0.62	'picture', 'dog', 'person', 'man', 'girl', 'friend', 'beautiful', 'hand', 'boy', 'guy', 'blue', 'baby', 'beach', 'street', 'woman', ...	'dog', 'person', 'man', 'girl', 'friend', 'boy', 'guy', 'baby', 'bird', 'woman', 'friends', 'bike', 'boat', ...	person(40), dog(2), bicycle(2), sheep(2), bird(1), boat(1), car(1), cat(1), chair(1), cow(1), horse(1)
	Logit: -4.24 Sim: 0.37	'man', 'girl', 'friend', 'boy', 'guy', 'woman', 'couple', 'horse', 'kid', ...	'man', 'girl', 'friend', 'boy', 'guy', 'woman', 'horse', 'kid', ...	person(7), horse(1)
	Logit: -7.13 Sim: 0.62	'dog', 'beach', 'street', 'pup', 'bull', 'husky', 'bulldog', 'horse', 'cow', 'doggo', 'goat', 'sheep', 'stray', ...	'dog', 'beach', 'street', 'pup', 'bull', 'husky', 'bulldog', 'horse', 'cow', 'goat', 'sheep', 'stray', ...	dog(14), sheep(7), cat(4), bird(3), horse(1)
	Logit: -4.24 Sim: 0.37	'cat', 'dog', 'beautiful', 'beach', 'street', 'pup', 'cute', 'pig', ...	'cat', 'dog', 'pig', 'bull', 'husky', 'bulldog', 'horse', 'dogs', ...	dog(6), sheep(4), cow(2), cat(1), horse(1)

Figure 4: Two Representative cases showing Target Extraction using WordNet

**Sim:** similarity score between classes in the dataset and words in the generated captions

**Logit:** Probability of token appearing in the caption

Now that we have our pseudo class labels, the next step was to use GradCAM to obtain heatmaps. The GradCAM technique requires two specific inputs, the layer/layers that needs to be visualized and the specific output/outputs(in our case the "tokens "or "words "present in the generated caption) for which heat map is required.The authors [2] suggests to remove the textual-head transformer architecture and just use the visual backbone to extract the visual features for any of the downstream tasks after the pre-training step . This led to us choosing to use GradCAM on the last layer of the ResNet-50 of the visual backbone part of the VirTex-v2 model as it is a well known fact that last residual block in ResNet learns most of the visual representations.

We implemented a class wrapper for the VirTex Model as our first step to implement GradCAM and then using the logit2word dictionary outputted by the VirTex Model, we inputted the target words that corresponded to that specific image region and generated heatmap visualizations for the same. To see the heatmaps generated by each pseudo class label refer to Figure: 5

After this, using these heatmap visualizations, we generated bounding boxes and bounding box coordinates by using a threshold of where maximum heat and minimum heat is present in the heatmap. After this, we also performed a post-processing step, where if multiple bounding boxes are of the same class, they are merged together through an averaging function.

Finally, with the pipeline up and running, we performed a hyperparameter search on a reduced dataset of 100 images to find the optimal configuration by evaluating using the mAP metric. These include the thresholds mentioned so far in addition to which GradCAM method is used(GradCAM vs. GradCAM++), the bounding box merging method(Mean vs. Median), and Eigen Smoothing. Our final configuration yielded a mAP score of 17.43%. When compared to SOTA weakly-supervised object detection methods that reach maximum mAP scores of **29%** our method performs relatively well. With some more improvement in GradCAM localization technique and extraction of words, we can achieve higher levels of mAP and achieve

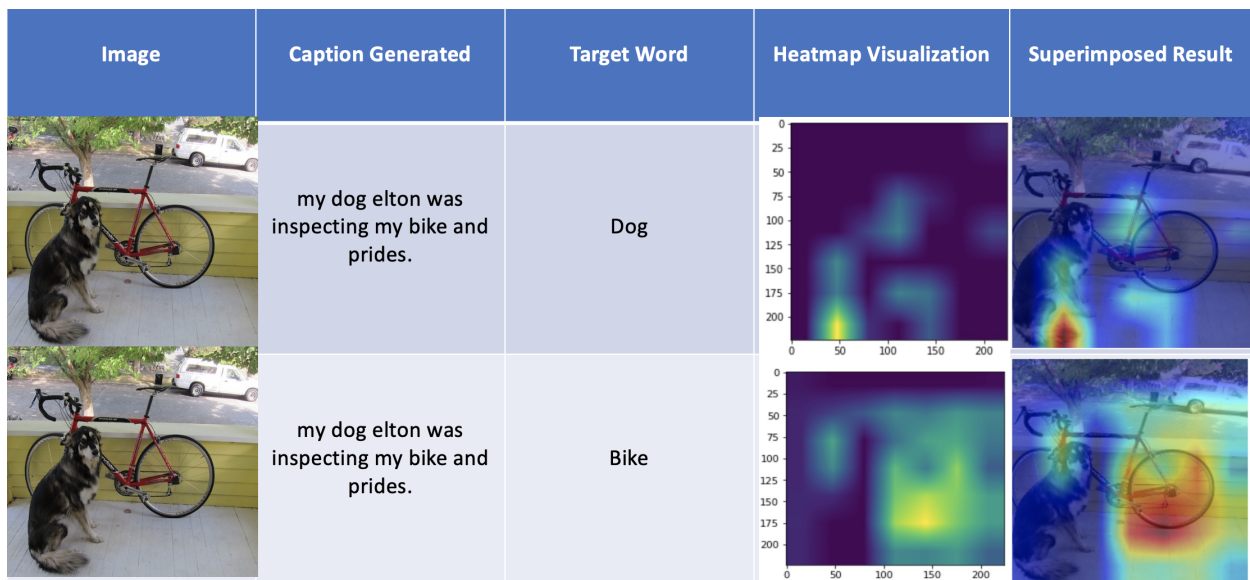


Figure 5: Heatmap visualizations for an image with the same caption generated but different target words

mAP results closer to the baseline. To see the results of the bounding boxes produces by our pipeline compared to the ground truth boxes refer to Figure: 6

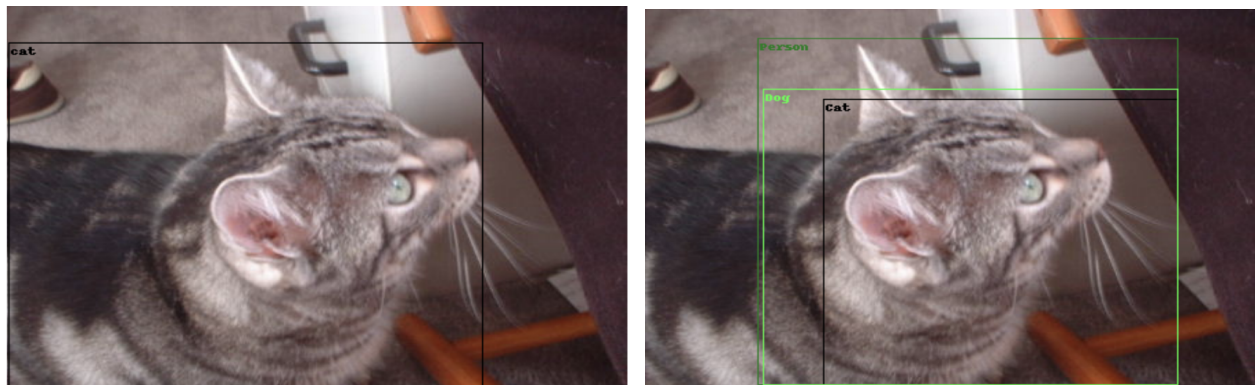






Figure 6: Experimental results comparing bounding boxes produces by our pipeline with the ground truth boxes

Ground Truth bounding box (left) and predicted bounding box from our pipeline (right)

To see the results obtained from testing different hyperparameter settings refer to Figure: 8

After this step, we used the pseudo classes and pseudo bounding boxes produced by our best performing model and fed it as training data to our FCOS object detection model. We modified the box loss function to use a smoothened L1 loss to account for noisy box labels. After passing this through our object detector we achieve the highest mAP of **21.57%** when evaluated against ground truth labels. To see the detailed result of mAP score please refer to Figure: 7

## 7 Conclusion

With a pipeline combining image captioning, gradient heatmap generation, and hand crafted methods to fit each part together, we were able to create an object detector that was trained with almost no ground truth

```

[ ] Total inference time: 184.6s
0.00% = aeroplane AP
0.00% = bicycle AP
17.94% = bird AP
0.00% = boat AP
0.00% = bottle AP
31.11% = bus AP
37.64% = car AP
7.04% = cat AP
10.63% = chair AP
17.34% = cow AP
59.33% = dog AP
33.86% = horse AP
58.71% = person AP
0.00% = sheep AP
0.00% = sofa AP
71.54% = train AP
mAP = 21.57%
<Figure size 640x480 with 1 Axes>

```

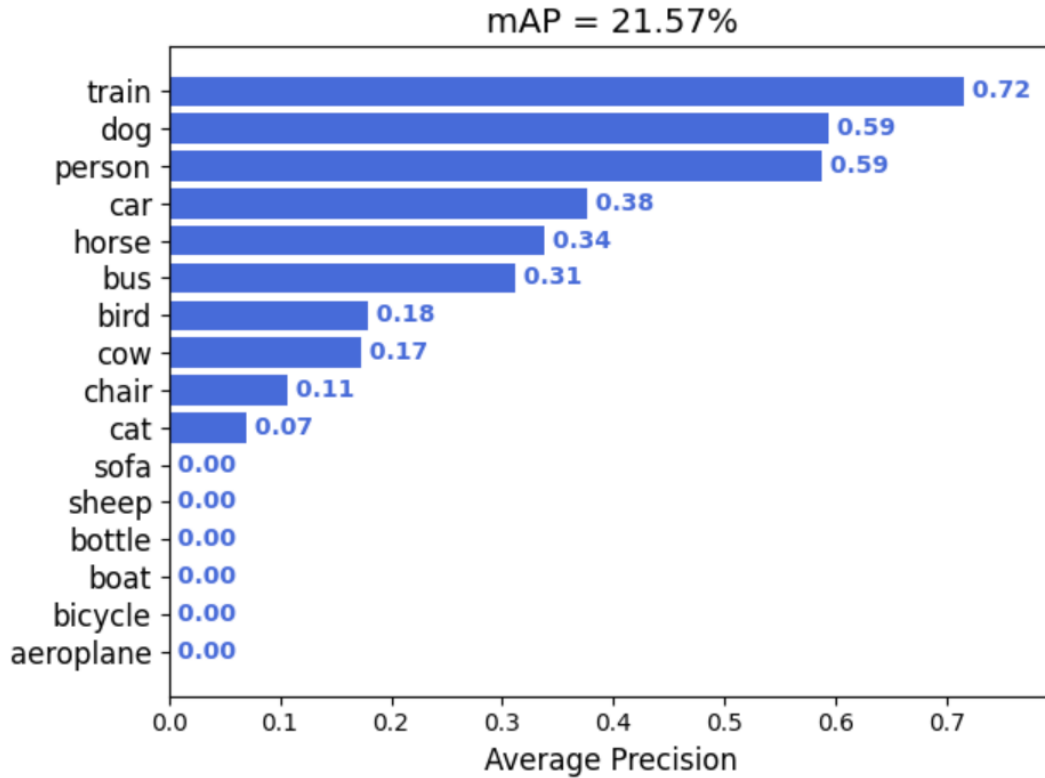


Figure 7: mAP score when we evaluated our predicted boxes from the FCOS Object Detector with the ground truth boxes

labels. We were able to achieve the best mAP score of **17.43%** when compared to the original ground truth bounding boxes. After training an object detector on this, pseudo training data we achieve the best mAP of



Method	Duplicate Removal	Smoothing	mAP(↑)
GradCAM	Mean	✗	10.64%
<b>GradCAM++</b>	<b>Mean</b>	✗	<b>17.43%</b>
GradCAM++	Median	✓	12.40%

Figure 8: A table mentioning the different mAP values achieved while testing models with different hyper-parameter settings

**21.57%**. Although there is further work to be done, we have shown that our method of autonomous data generation is feasible and yields fruitful results.

## 8 Future Work

One of the main ways to improve our method would be to research improvements for bounding box noise. Many bounding boxes are generated for each class and merging them into a single accurate bounding box is an ongoing challenge. We can come up with better ways to get rid of redundant boxes. Furthermore, it is difficult to differentiate between multiple instances of a single class in an image and multiple bounding boxes generated for the same instance - we currently threshold IoU of boxes to determine instance separation. Extending our method to perform better while trying to evaluate multiple objects in a single image.

Another task left to do would be to make improvements to label assignment. Some words, like names, are hard to assign to a particular class from text alone. Additionally, since the dataset for image captioning is reddit, slang words or abbreviations appear frequently and are harder to deal with.

We could also scale up the object detector model and increase the resolution of images and we hope that the mAP score scales accordingly.

## 9 Author Contributions

Max Hamilton investigated the RedCaps code and made modifications to support GradCAM. This included returning internal variables like the per-token logits and a mapping from each logit to its corresponding part of the caption. Kshama Nitin Shah and Vineet Rao then integrated this captioning model with GradCam to produce heatmaps. Kshama investigated the layers on which GradCAM would yield most fruitful results, Vineet implemented the Class Wrapper and made necessary changes to the GradCAM code to integrate GradCAM with Virtex. Kshama and Vineet modified the predict function of the Virtex model so that the input to the Virtex and GradCAM models would be compatible. Madhav implemented code to parse the caption and detect target words to use GradCAM on. These are the words that should correspond to objects in the image. All authors integrated the prompt engineering. Wen Jay worked on creating a bounding box generation method from the GradCam heatmap. This serves as the pseudo-label for the object detection task. All co-authors equally contributed to this project and to writing the report.

## References

- [1] K. Desai and J. Johnson. Virtex: Learning visual representations from textual annotations. *CoRR*, abs/2006.06666, 2020.
- [2] K. Desai, G. Kaul, Z. Aysola, and J. Johnson. RedCaps: Web-curated image-text data created by the people, for the people. In *NeurIPS Datasets and Benchmarks*, 2021.
- [3] J. Ding, E. Xie, H. Xu, C. Jiang, Z. Li, P. Luo, and G.-S. Xia. Unsupervised pretraining for object detection by patch reidentification, 2021.
- [4] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results. <http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html>.

- [5] G. A. Miller. Wordnet: A lexical database for english. *Commun. ACM*, 38(11):39–41, Nov. 1995.
- [6] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. *International Journal of Computer Vision*, 128(2):336–359, Oct 2019.
- [7] Z. Tian, C. Shen, H. Chen, and T. He. FCOS: fully convolutional one-stage object detection. *CoRR*, abs/1904.01355, 2019.
- [8] E. Xie, J. Ding, W. Wang, X. Zhan, H. Xu, P. Sun, Z. Li, and P. Luo. Detco: Unsupervised contrastive learning for object detection, 2021.