Transformer-Encoder Detector Module: Using Context to Improve Robustness to Adversarial Attacks on Object Detection

Faisal Alamri
Department of Computer Science
The University of Exeter
Email: fa269@exeter.ac.uk

Sinan Kalkan
Department of Computer Engineering
Middle East Technical University
Email: skalkan@ceng.metu.edu.tr

Nicolas Pugeault
School of Computing Science
University of Glasgow
Email: nicolas.pugeault@glasgow.ac.uk

***Abstract*—Deep neural network approaches have demonstrated high performance in object recognition (CNN) and detection (Faster-RCNN) tasks, but experiments have shown that such architectures are vulnerable to adversarial attacks (FFF, UAP): low amplitude perturbations, barely perceptible by the human eye, can lead to a drastic reduction in labelling performance. This article proposes a new context module, called *Transformer-Encoder Detector Module*, that can be applied to an object detector to (i) improve the labelling of object instances; and (ii) improve the detector's robustness to adversarial attacks. The proposed model achieves higher mAP, F1 scores and AUC average score of up to 13% compared to the baseline Faster-RCNN detector, and an mAP score 8 points higher on images subjected to FFF or UAP attacks due to the inclusion of both contextual and visual features extracted from scene and encoded into the model. The result demonstrates that a simple ad-hoc context module can improve the reliability of object detectors significantly.**

I. Introduction

Recognising objects in a visual scene is an effortless task for humans, one that has challenged computer vision since its inception. The advent of deep neural network approaches over the last decades has delivered major improvements on this task on all benchmarks, although some difficulties remain [39].

One notable weakness of deep neural network approaches to vision, first demonstrated by Szegedy *et al.* [34], is that perturbations of very small amplitude, barely visible to the human eye, can be found that lead to major decrease in the neural networks' labelling accuracy—so called *adversarial attacks*. This seminal finding was confirmed by following studies, which showed that such attacks could be designed to be independent on the specific neural network or even neural architecture employed [23], and that such attack patterns could be data independent [24].

This apparent weakness is in contrast to how robust human perception of objects is to a variety of noise and perturbations. One plausible reason for this difference in robustness between human and artificial perception is *context* [3]. Contextual information plays an important role in visual recognition for both human and computer vision systems [3], [28]. Imagine there are an *office desktop*, a *chair*, a *PC*, a *monitor* and a *keyboard*, then we ask the question "*what is the missing object in this scene?*". Some may say a *cup*, or a *pen*, but when looking deeply and understanding the context, you are likely to say the missing object is a *mouse*. Such guesses are made even without seeing or gazing at the scene due to leveraging contextual information. This shows how context carries rich information about visual scenes. In terms of object recognition, the context could be defined as cues captured from a scene that presents knowledge about objects locations, size and object-to-object relationships. Due to the importance of contextual information and how it improves detection, it has been widely studied [7], [11], [14], [25], [40]. Contextual information can be captured from images explicitly as attempted in [1], or implicitly as proposed in this paper upon the success of the Transformer model proposed by [35] for Natural Language Processing.

This paper describes a contextual detection module, proposed as an add-on to classical object detection architectures such as Faster-RCNN, named *Transformer-Encoder Detector Module* (TEDM). This model implicitly encodes contextual statistics of objects and uses attention mechanism to improve the labelling of image regions. It improves the detection performance of a state-of-the-art object detector, as evaluated on natural images and perturbed images. This model does rescore, relabel, and correct detector predictions, in contrast to [8], [11], which only rescore detected objects' confidences.

This paper is organised as follows: First, we review the Transformer Model, and how it works. CNN-based detectors (Section II). In Section III, we explain how the proposed model is built and how features are encoded in the pipeline. Finally, the proposed model is examined on MSCOCO2017 dataset in comparison with Faster RCNN detector and the Relabelling Model proposed by [1] in Section IV. Natural and perturbed images are both used to evaluate the proposed model.

II. Related Work

A. Adversarial attacks

The vulnerability of deep neural network architectures to small amplitude perturbations optimised to damage the

This work is accepted for the 25th International Conference on Pattern Recognition (ICPR'2020).

networks' performance has first been found by Szegedy *et al.* [34]. They demonstrated that image perturbations of small amplitude, barely visible to the eye, when optimised to cause the highest misclassification error, can achieve large decreases in performance. Moreover, such perturbations can decrease performance across network architectures and even on different datasets.

Mohsen *et al.* [23] proposed an approach to find Universal Adversarial Perturbations (UAP), single patterns of perturbation that lead to large misclassification across all inputs of the network, and demonstrated its efficiency on various classifiers (e.g. CaffeNet [18], VGG-16 [32], VGG-19 [33]), claiming that the addition of the perturbations drops the tested classifiers' performance by almost 90% as tested on ImageNet validation dataset. Therefore, this leads to the result that the single universal perturbation vectors can cause most of the natural images to be misclassified. Therefore, it can be concluded that using UAP perturbations optimised on a network architecture X (e.g., VGG-F) will also lead to misclassification when applied to images classified with another network architecture Y (e.g., CaffeNet).

Moreover, Reddy *et al.* [24] claim that their proposed model, named Fast Feature Fool (FFF), is transferable to various different models (i.e. they use the same models examined in [23]). The main and important difference between UAP and FFF is that FFF is data-independent. In other words, it can be applied on any model even if the trained and tested models are trained on different datasets (i.e., no prior knowledge about the target CNNs is needed). It is also reported that when FFF is trained on X dataset and evaluated on Y dataset, it can still impact the CNN models, leading to an increase in the fooling rate compared to UAP, which is a data-dependent method.

B. Visual context for object detection

Contextual information can help improve detection performance due to the knowledge it provides from scenes [14], [20]. It is defined in [2] as *any data obtained from an object's own statistical property and/or from its vicinity, including intra-class and inter-class details*. Such a definition is claimed due to the information observed while studying the importance of context in digital images.

It is said that contextual information is a tool used more with multiple objects so that relationships among objects can be deeply understood [12]. Roozbeh *et al.* [25] also state that in digital images, objects with clear appearance (e.g. large objects) are easy to detect, whereas some small objects are harder. Lubor *et al.* [27] also claim that contextual information, therefore, can be a solution here as it provides stronger cues in detecting small objects due to the context where those objects are present. Hence, contextual information is described as *"a natural way to improve detection"* [4], [7].

Contextual information in the field of object detection can help to understand and explore object vicinity (*i.e.*, scene-level context) as applied in [5], [38], and also provides object-object relationships (*i.e.*, object-level context) as in [10], [29]. Moreover, Contextual information has been also studied in different areas, such as object localisation [11], image segmentation [15], image annotation [21], scene modeling [7] and cognitive robotics [40].

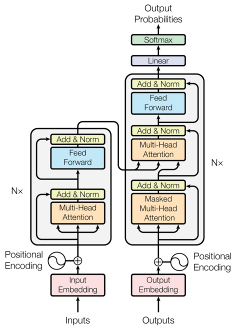

Fig. 1. The Transformer Architecture, reproduced from [35]

Context can be classified upon the sources of information extracted from images. Biederman *et al.* [6] state that there are five categories of object-environment dependencies, which are: *"(i)* ***interposition*** *objects interrupt their background, (ii)* ***support****: objects often rest on surfaces, (iii)* ***probability****: objects tend to be found in some environments but not others, (iv)* ***position****: given an object in a scene, it is often found in some positions but not others, and (v)* ***familiar size****: objects have a limited set of sizes relative to other objects"*. Galleguillos *et al.* [14] grouped those relationships into three main categories: (i) **Semantic** (Probability), (ii) **Spatial** (interposition, support and position) and (iii) **Scale** (familiar size).

C. Attention and the Transformer model

Vaswami *et al.* [35] proposed the Transformer architecture for natural language processing, as a way to encode word context from documents corpora using dot product attention between word vectors. The Transformer consists of a set of encoders and decoders, which are composed of a stack of layers. In terms of encoders: each has two major components, which are (1) self-attention layer and (2) feed-forward neural network. On the other hand, the decoder has two similar components and an additional one: (1) self-attention layer, (2) encoder-decoder layer, and (3) feed-forward neural network. The Transformer architecture is shown in Figure 1. We refer the reader to [35] for further information. The Transformer architecture, due to its performance and speed, has been used in a variety of approaches in NLP, such as [13], [30], [36], [37], and has recently been applied to computer vision problems [9].

III. Proposed Method

A novel model is proposed in this paper, which obtains a better performance compared to the Faster-RCNN detector whether perturbations are applied to the images or not. The proposed model is built upon the success of the Transformer model proposed by [35]. However, only the Transformer-Encoder is adapted, as shown in Figure 2, where the proposed model, named *Transformer-Encoder Detector Module* (TEDM) architecture is illustrated.

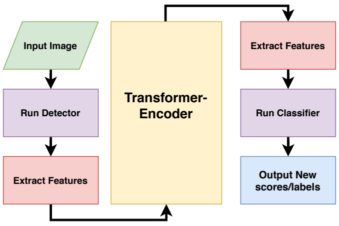

Fig. 2. Transformer-Encoder Detector Module Architecture

First, an image is passed to the baseline detector (i.e., Faster-RCNN) for feature extraction. Faster-RCNN, proposed by Shaoqing *et al.* [31], uses a two stages for detection. In the first stage, it uses a CNN network to extract regions in the image that are likely to contain an object. The second stage, moreover, performs detection (classification & localisation) for each region proposed by the first stage. Faster-RCNN has been implemented in a variety of articles studying the importance of contextual information (e.g. [16], [17]), and it is still one of the state-of-the-art detection methods, and due to some of its advantages (e.g. speed, accuracy), it is also used as the baseline detector in this paper.

Features representing image regions for the detected objects are extracted. As the dimension of the features extracted is $4,096$, they are mean-pooled to $2,048$ to suit further processing. The pooled features and the boundary boxes obtained from the detector representing the detected regions are passed into the Transformer-Encoder with a dimension of $2,048$. Boundary boxes are included to encode the spatial features to extract the contextual information among regions implicitly. Since there is no recurrence in the model, the positions of the inputs are needed, thus positional encoding is applied, which "injects some information about the relative or absolute position of the tokens in the sequence", using sine and cosine functions of different frequencies. In other words, it is used to obtain the context of each feature inputted [35].

The Transformer Model, moreover, is then applied, which is already pre-trained on MS COCO 2014, adapted from [19], [35]. In details, the Transformer-Encoder takes the passed features and processes them with the attention mechanism to output features with a dimension of 512. A feed-forward NN classifier is then applied. The classifier produces the new scores and predicted labels for each region. We use a *trainscg* (scaled conjugate gradient back-propagation) Neural Network approach, as implemented in MATLAB [22]. Scaled conjugate gradient (SCG), a supervised learning algorithm, is a network training function used to update weight and bias value according to the scaled conjugate gradient method [26]. *trainscg* was implemented as explained in [22]. The standard network consists of a two-layer feed-forward network, with a sigmoid activation function in the hidden layer, and a softmax function in the output layer. The number of hidden units defined is 200, as different numbers were tested, and 200 achieved the highest performance for this data.

This method, to the best of our knowledge, is the first to include the Transformer-Encoder to benefit from the use of attention and the positional encoding operation to apply for object detection performance in both natural and perturbed images. The positional encoding, as said, helps to simply encode the semantic and spatial contexts for each region inputted implicitly. *TEDM* is examined, as below, on the MS COCO 2017 *val* dataset in comparison with Faster RCNN (i.e., baseline detector) on two types of images: (1) Natural images, where no perturbations are added, and (2) perturbed images, as presented in Section IV.

IV. Experiments

This section presents the results obtained from the application of the new proposed method (*TEDM*). Images before any attacks, called natural images, are used to examine the impact of this model in comparison with Faster-RCNN, as in Experiment one (Section IV-A). Experiment Two (Section IV-B) illustrates the examination of the *TEDM* vs. Faster RCNN on images under adversarial attacks.

A. Experiment One: Natural Images

In this section, we are presenting how effective *TEDM* is in comparison with Faster RCNN when applied on natural images.

Note that images used in these experiments are taken from MS COCO 2017 *val* dataset. Only the images with more than one object detected by Faster RCNN are used in order to compare the performance of this proposed model with the *Relabelling Model* proposed by [1], which explicitly extracts contextual features from images. This model improves the detection performance of Faster RCNN detector as it rescores and relabels predictions. using some Refer to [1], [2] further reading about contextual information and how *Relabelling Model* works.

First, two chosen images with only one object presented are used to examine the performance of *TEDM* when a single object is presented. This is because *TEDM* does not solely rely on contextual information as the *Relabelling Model*, but also on the appearance features extracted by Faster RCNN. Therefore, it is expected to work well even with one object presented as illustrated in Figure 3, where the predictions obtained from Faster RCNN and *TEDM* are shown on left and right sides, respectively.

As we can see in the first row in Figure 3, a *person* is detected by Faster RCNN with a confidence of *0.9985* and also predicted by *TEDM*, but with very slightly lower confidence as *0.9883*. Similarly, in the second row, a *kite* is detected by Faster RCNN as the only object presented. Applying *TEDM*, in this case, increases the confidence from *0.9647* to *0.9920*. The two images are shown as each presents one object that belongs to a different category. In all cases, *TEDM* is believed to work very well when a single object is presented. It can be said that *TEDM* is a compatible method producing comparable results as Faster RCNN. Below, more visualised and statistical results are reported where more than one object in the images used are presented.

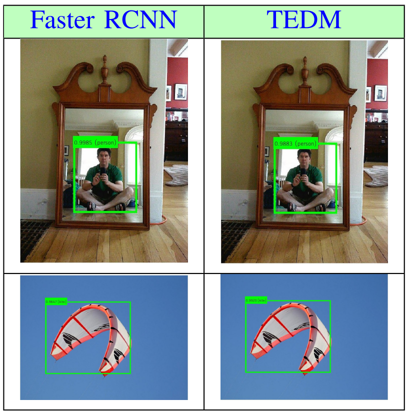

Fig. 3. Results: Faster RCNN vs. TEDM outputs on images with a single Object: Green boxes represent correct detection.

TABLE I
A COMPARISON BETWEEN FASTER RCNN AND TEDM IN TERMS OF AUC SCORES OF SOME MS COCO CLASSES.

Class Name	Faster RCNN	Relabelling Model [1]	TEDM
Person	0.84345	0.84568	**0.95174**
Car	0.81246	0.80474	**0.93287**
Cow	0.82301	0.84035	**0.93246**
Snowboard	0.74229	0.87412	**0.88235**
Sport ball	0.84244	0.84203	**0.95008**
Mean	0.76472	0.78278	**0.89222**

Images with more than one object are used to keep consistent and to report comparable statistical results with the *Relabelling Model*. In Table I, the AUC scores for some objects and all objects average for both *TEDM* and Faster RCNN are presented. It can be noticed that *TEDM* outperforms the baseline detector in all cases shown obtaining a higher average score of *0.8922*. This is even better than the *Relabelling Model*, whose mean of AUC scores is *0.78278*.

Furthermore, Table II shows the mAP$_{0.5}$ and mAP (IoU=[0.5:0.05:0.95]) for *TEDM* and Faster RCNN. As presented, *TEDM* outperforms the baseline. Compared with the *Relabelling Model*, whose mAP and F1 scores are 65.50% and 58.95%, as reported with [1], *TEDM* still performs better.

In addition, as shown in Figure 4, two randomly chosen images taken from MS COCO 2017 validation dataset are used to examine the performance of *TEDM* compared with Faster RCNN. In the first row, seven objects are detected by Faster RCNN. Four objects, which are a *person*, a *TV*, a *keyboard* and a *mouse* are detected correctly. However, the other three, which are a *person*, a *laptop* and *a chair* are incorrectly detected. *TEDM* is applied, which removes two of the incorrectly detected objects, but still predicts the laptop incorrectly, with very low confidence. This result is promising as it shows how *TEDM* removes incorrect objects and reduces the confidence of the *laptop*. More importantly, in the following image, *TEDM* outperforms Faster RCNN predictions. It corrects the *bed*, which is incorrectly detected, to a *couch* as correct detection. This clearly shows how the proposed model does not only contribute to removing false detections but also how it rescores and relabels them.

TABLE II
MAP AND F1 SCORES IN PERCENTAGES [%] FOR FASTER RCNN AND TRANSFORMER-ENCODER DETECTOR MODULE (TEDM).

Model	mAP$_{0.5}$	mAP	F1
Faster RCNN	62.82	33.48	57.34
Relabelling Model [1]	65.50	34.07	58.95
Transformer-Encoder Detector Module	**69.07**	**38.19**	**63.27**

Summary. Statistical and visual results suggest that *TEDM* achieves a better performance than the baseline detector benefiting from the self-attention mechanism. It also does a great job in detecting either single-object or multiple objects, which are better in comparison with Faster RCNN and the *Relabelling Model* [2].

B. *Experiment Two: Adversarial Images*

Adversarial images are used to examine the impact of *TEDM*, i.e. the use of contextual information, against adversarial attacks. Such images may have different visual features due to the addition of Adversarial Perturbations leading to an effect on contextual information especially when some objects are misdetected.

Adversarial Perturbations are noises carefully computed such that, when added to images, they are not visible to the naked human eye but they do fool the Deep Neural Networks (DNNs) into mispredictions.

Due to the impact that both methods (i.e. UAP and FFF) cause to the CNN models, they are used in this paper to examine their effect on *TEDM* and Faster RCNN detector. UAP perturbation used in this experiment is trained on MS COCO 2017, as it is data-dependent. However, FFF is used as proposed (i.e. already trained on ImageNet), but will be tested on MS COCO 2017. Figure 5 shows the perturbations, which are added on images. It can be seen they are different, even though they both are trained on the same model (i.e. VGG-16), but on different training datasets.

The two types of perturbations are applied and added to images in two different approaches: (1) added to the entire image as experimented in Section IV-B1, (2) added to each detected region as experimented in Section IV-B2.

1) Perturbation on the Entire Image: In this section, perturbations are added to the entire image, where statistical and visual results are illustrated, below, to show the performance of *TEDM* in comparison with Faster RCNN.

In terms of the statistical results, as shown in Table III. The average AUC scores for *TEDM* is noticeably higher than Faster RCNN in both types of attacks. Considerably, *TEDM* obtains higher scores for some objects such as the *train*, *stop sign* and *Pizza*, we can clearly see a huge difference, as *TEDM* benefits from the use of visual and contextual features in contrast to Faster RCNN, which depends only on the visual features. In comparison with the *Relabelling Model*, which encodes contextual information in an explicitly manner, the

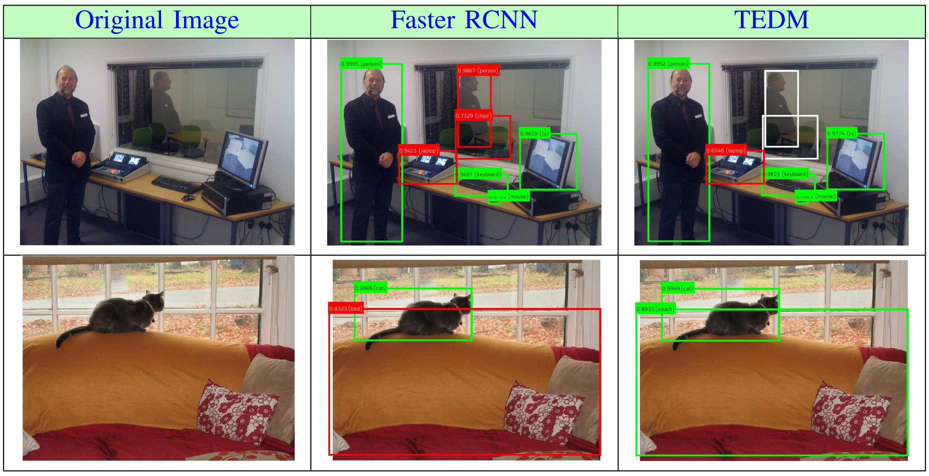

Fig. 4. Results: Faster RCNN vs. TEDM outputs: Green, red and white boxes represent correct detection, incorrect detection, and objects removed and relabelled as background, respectively

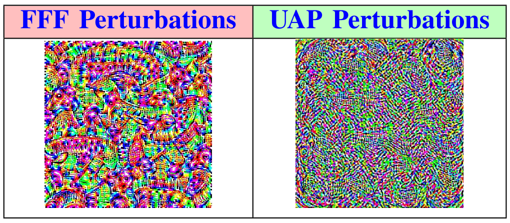

Fig. 5. FFF vs. UAP perturbations added to images

TABLE IV
AP AND F1 SCORES IN PERCENTAGES [%] FOR FASTER RCNN AND TEDM ON PERTURBED IMAGES

Attacks	Models	$\text{mAP}_{0.5}$	F1 Score
FFF	Faster RCNN TEDM	45.34% **50.59%**	39.13% **43.16%**
UAP	Faster RCNN TEDM	46.40% **51.67%**	40.05% **44.00%**

TABLE III
A COMPARISON BETWEEN SOME MSCOCO OBJECT CLASSES AUC SCORES BETWEEN FASTER RCNN AND TEDM ON PERTURBED IMAGES

	FFF		UAP	
Class Label	Detector	TEDM	Detector	TEDM
Person	0.62389	**0.69576**	0.62881	**0.73342**
Train	0.42961	**0.62051**	0.41062	**0.56773**
Stop Sign	0.76242	**1**	0.69155	**0.78571**
Surfboard	0.39266	**0.52464**	0.44750	**0.59365**
Pizza	0.67393	**0.87767**	0.63185	**0.81783**
Mean	0.34714	**0.47683**	0.36050	**0.48713**

average AUC scores are *0.4203*, and *0.4333* for FFF and UAP attacks respectively.

In addition, Table IV illustrates the mAP and F1 scores for *TEDM* and Faster RCNN, where FFF and UAP perturbations are applied. Clearly, *TEDM* scores a higher performance than Faster RCNN in both cases. However, in the case of UAP attack, we can see that both models are less impacted compared with FFF. In comparison with the *Relabelling Model* which obtains mAP and F1 of 49.39%, 41.91% and 50.72%, 43.12% for FFF and UAP respectively.

Figure 6 presents two images illustrating the impact of FFF and UAP perturbations. Firstly, in the top row results, FFF perturbation is added to the entire image. Faster RCNN detects two objects correctly, which are two *persons*, whereas incorrectly detects an *umbrella*, which is actually a *wall*. The same perturbed image is passed into the *TEDM* predicting the presence of only the correct objects, which are the two *persons*, and relabels the *umbrella* as a background. However, in compare with Faster RCNN, *TEDM* scores the correctly detected objects with lower confidences, but still higher than *0.81*.

In terms of UAP perturbations, as illustrated in the second row. We can see that both models perform very well. However, Faster RCNN fails to detect one of the *frisbees*, which labels as a *sports ball*. *TEDM* correctly labels it as *frisbee* with a confidence of *0.9089*. Both models score all other objects with considerably high confidences, even though perturbations are added.

Summary. We can clearly see that *TEDM* performs better scoring higher average AUC scores in the majority of cases, and higher mAP and F1 scores compared with Faster RCNN and the *Relabelling Model*. From the reported results, we can say that *TEDM* is an excellent tool to overcome some of the errors that Faster RCNN attempts. Below, both models are examined when perturbations are added on regions rather than on the entire image.

2) Perturbation On Regions: In this section, perturbation is added on detected regions, because perturbations are added to images, which are passed into single-object models, as presented and developed in [23], [24]. Therefore, perturbations will be added to each region of interest (RoI) that the detector detects during the detection process.

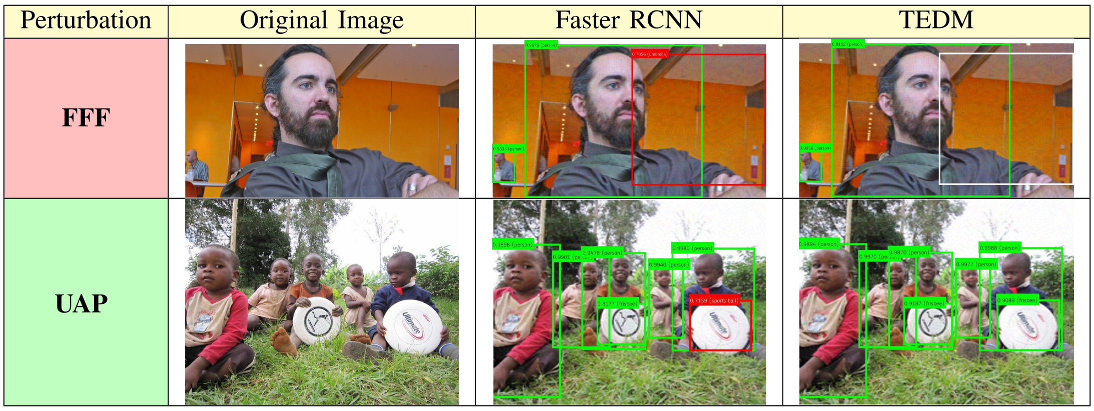

Fig. 6. Results: Faster RCNN and TEDM outputs for FFF and UAP perturbed images: Green, red and white boxes represent correct detection, incorrect detection, and objects removed and relabelled as background, respectively

TABLE V
AP AND F1 SCORES IN PERCENTAGES [%] FOR FASTER RCNN AND TEDM ON PERTURBED IMAGES

Attacks	Models	$mAP_{0.5}$	F1 Score
FFF	Faster RCNN	26.25%	20.78%
	TEDM	**35.28%**	**26.25%**
UAP	Faster RCNN	23.18%	18.17%
	TEDM	**31.39%**	**24.14%**

TABLE VI
A COMPARISON BETWEEN SOME MSCOCO OBJECT CLASSES AUC SCORES BETWEEN DETECTOR AND TEDM ON PERTURBED REGIONS

	FFF		UAP	
Class Label	Detector	TEDM	Detector	TEDM
Person	0.18573	**0.31982**	0.14653	**0.28385**
Train	0.05501	**0.08737**	**0.03883**	0.02912
Cat	0.06807	**0.10563**	0.03990	**0.84507**
Wine Glass	**0.27083**	0.19791	0.24791	**0.25625**
Microwave	**0.17241**	0	0.06896	**0.09143**
Mean	0.14310	**0.23058**	0.12043	**0.19991**

Looking at Table V we can see that *TEDM* outperforms Faster RCNN in both types of perturbation. FFF seems to impact both models lower than what UAP does, but in general, both attacks significantly drop the performance of both models. We believe *TEDM* can be offered, to some extent, yet as a solution to address perturbation issue in the field of object detection. It also outperforms the *Relabelling Model*, whose mAP and F1 scores for both FFF and UAP perturbations are respectively 29.05%, 23.14% and 26.36%, 20.53%.

AUC scores are computed as presented in Table VI. On average, *TEDM* performs better than Faster RCNN in the two cases of perturbation. As said, both models perform better when FFF attack is applied, unlike when UAP perturbation added. However, in some classes such as *microwave* Faster RCNN has higher AUC scores (as per class) comparing with the proposed method. In comparison with the *Relabelling Model*, which has an average AUC score for FFF and UAP as 0.19470 and 0.16994, where *TEDM* again achieves better performance.

Table 7 presents some results obtained from the application of both attacks on Faster RCNN and *TEDM*. An image showing the detected regions before any perturbation added, followed by the predictions of both models are illustrated. As the aim, here, is not just to illustrate the differences between models performances before and after the attack, but rather to show the differences among both models after attacked per region is added. Therefore, results after the attack are reported to provide ease when comparing.

In the first image, where FFF perturbation is added to regions *TEDM* predicts no objects. All objects detected by Faster RCNN after attacked are false detection. *TEDM* helps to prevent false detections that Faster RCNN outputs. Noticeably, as shown in the detected regions before the attack, the *bed* is detected, but after the attack, it could not be predicted. This is a good example of how negatively the perturbation impacts the model. It is found that when the regions are large in size, they are likely to be impacted more by perturbation resulting in not being detected.

In terms of UAP perturbation, ten regions are detected by Faster RCNN before the attack, and only half of them are detected after. Faster RCNN detects four objects correctly but fails in detects the *cup*. The *TEDM* fails to detect the *person* that Faster RCNN already detects, which can be due to the perturbation and lighting conditions. We can see that the *person* in the large region is not detected, which can be in line with the findings stated earlier that larger regions are more likely to be impacted.

Summary. *TEDM* is performing better than Faster RCNN to tackle adversarial perturbations. This is because of the features encoding during the encoder process, as it learns the spatial and visual features.

V. Conclusion

This paper presented a new context module, called *Transformer-Encoder Detector Module (TEDM)*, which when combined with an object detection architecture to improve

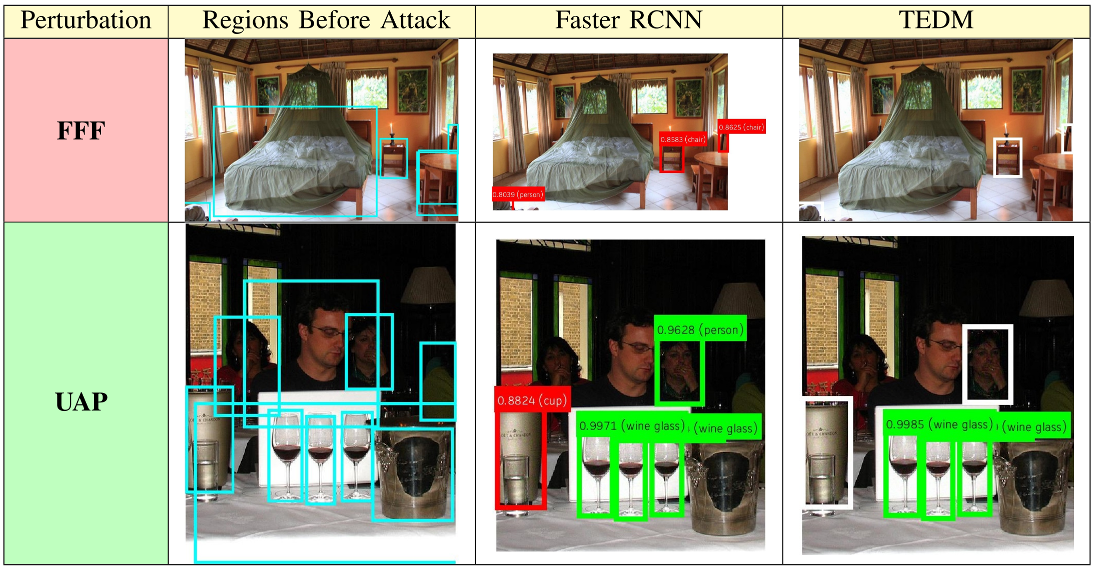

Fig. 7. Results: Faster RCNN and TEDM outputs for FFF and UAP perturbed regions: Light blue boxes represent regions detected before perturbation added. Green, red and white boxes represent correct detection, incorrect detection, and objects removed and relabelled as background, respectively.

both performance and robustness to adversarial attacks. The proposed model is based on the encoder part from the Transform model [35] to encode the contextual features implicitly for scenes, and visual features using Faster RCNN. *TEDM* was examined on natural images and perturbed images, where it outperforms Faster RCNN and a contextual model that explicitly encodes contextual features.

As experimented, the impact of adversarial attacks was reported to be higher when applied on regions, which we believe is due to the size of the regions: the larger the region is, the more it is impacted. Surprisingly, UAP perturbation affects the performance of the examined models when added to the entire image less than FFF does. However, when added to regions, it drops performances considerably, which can be due to the data-dependency.

Future work will involve developing an end-to-end model, to refine not only predictions but also boundary boxes from both contextual and visual features.

References

[1] Faisal Alamri and Nicolas Pugeault. Contextual relabelling of detected objects. In *2019 Joint IEEE 9th International Conference on Development and Learning and Epigenetic Robotics (ICDL-EpiRob)*, pages 313–319, 2019.

[2] Faisal Alamri and Nicolas Pugeault. Improving object detection performance using scene contextual constraints. *IEEE Transactions on Cognitive and Developmental Systems*, July 2020.

[3] Moshe Bar. Visual objects in context. *Nature Reviews Neuroscience*, 5(8):617–629, 2004.

[4] Ehud Barnea and Ohad Ben-Shahar. Contextual object detection with a few relevant neighbors. *CoRR*, abs/1711.05705, 2017.

[5] Sean Bell, C. Lawrence Zitnick, Kavita Bala, and Ross B. Girshick. Inside-outside net: Detecting objects in context with skip pooling and recurrent neural networks. *CoRR*, abs/1512.04143, 2015.

[6] Irving Biederman, Robert J. Mezzanotte, and Jan C. Rabinowitz. Scene perception: Detecting and judging objects undergoing relational violations. *Cognitive Psychology*, 14(2):143 – 177, 1982.

[7] Ilker Bozcan and Sinan Kalkan. Cosmo: contextualized scene modeling with boltzmann machines. *Robotics and Autonomous Systems*, 113:132–148, 2019.

[8] X. Cao, X. Wei, Y. Han, and X. Chen. An object-level high-order contextual descriptor based on semantic, spatial, and scale cues. *IEEE Transactions on Cybernetics*, 45(7):1327–1339, July 2015.

[9] Nicolas Carion, F. Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander M Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. *ArXiv*, abs/2005.12872, 2020.

[10] Xinlei Chen and Abhinav Gupta. Spatial memory for context reasoning in object detection. *CoRR*, abs/1704.04224, 2017.

[11] Myung Jin Choi, Antonio Torralba, and Alan S. Willsky. A tree-based context model for object recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(2):240–252, Feb 2012.

[12] Chaitanya Desai, Deva Ramanan, and Charless C. Fowlkes. Discriminative models for multi-class object layout. *International Journal of Computer Vision*, 95(1):1–12, Oct 2011.

[13] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018.

[14] Carolina Galleguillos and Serge Belongie. Context based object categorization: A critical survey. *Comput. Vis. Image Underst.*, 114(6):712–722, June 2010.

[15] Stephen Gould, Jim Rodgers, David Cohen, Gal Elidan, and Daphne Koller. Multi-class segmentation with relative location prior. *International Journal of Computer Vision*, 80(3):300–316, Dec 2008.

[16] Han Hu, Jiayuan Gu, Zheng Zhang, Jifeng Dai, and Yichen Wei. Relation networks for object detection. *CoRR*, abs/1711.11575, 2017.

[17] N. Inoue, R. Furuta, T. Yamasaki, and K. Aizawa. Object detection refinement using markov random field based pruning and learning based rescoring. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1652–1656, March 2017.

[18] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the 22Nd ACM International Conference on Multimedia*, MM '14, pages 675–678, New York, NY, USA, 2014. ACM.

[19] Martin Krasser. Image captioning transformer. https://github.com/krasserm/fairseq-image-captioning#image-captioning-transformer, July 2020. (Accessed on 07/15/2020).

[20] Li Liu, Wanli Ouyang, Xiaogang Wang, Paul W. Fieguth, Jie Chen, Xinwang Liu, and Matti Pietikäinen. Deep learning for generic object detection: A survey. *CoRR*, abs/1809.02165, 2018.

[21] Zhiwu Lu, Horace H. S. Ip, and Yuxin Peng. Contextual kernel

and spectral methods for learning the semantics of images. *IEEE Transactions on Image Processing*, 20(6):1739–1750, June 2011.

[22] MATLAB. *Deep Learning Toolbox R2017a*. The MathWorks Inc., Natick, Massachusetts, United States, 2017.

[23] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. Universal adversarial perturbations. *CoRR*, abs/1610.08401, 2016.

[24] Konda Reddy Mopuri, Utsav Garg, and R. Venkatesh Babu. Fast feature fool: A data independent approach to universal adversarial perturbations. *CoRR*, abs/1707.05572, 2017.

[25] Roozbeh Mottaghi, Xianjie Chen, Xiaobai Liu, Nam-Gyu Cho, Seong-Whan Lee, Sanja Fidler, Raquel Urtasun, and Alan Yuille. The role of context for object detection and semantic segmentation in the wild. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 891–898, June 2014.

[26] Martin Fodslette Møller. A scaled conjugate gradient algorithm for fast supervised learning. *Neural Networks*, 6(4):525 – 533, 1993.

[27] Dennis Park, Deva Ramanan, and Charless Fowlkes. Multiresolution models for object detection. In Kostas Daniilidis, Petros Maragos, and Nikos Paragios, editors, *Computer Vision – ECCV 2010*, pages 241–254, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.

[28] Andrew Rabinovich, Andrea Vedaldi, Carolina Galleguillos, Eric Wiewiora, and Serge Belongie. Objects in context. In *2007 IEEE 11th International Conference on Computer Vision*, pages 1–8. IEEE, 2007.

[29] Andrew Rabinovich, Andrea Vedaldi, Carolina Galleguillos, Eric Wiewiora, and Serge J. Belongie. Objects in context. *2007 IEEE 11th International Conference on Computer Vision*, pages 1–8, 2007.

[30] Alec Radford. Improving language understanding by generative pre-training. In *Technical report, OpenAI*, 2018.

[31] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster R-CNN: towards real-time object detection with region proposal networks. *CoRR*, abs/1506.01497, 2015.

[32] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015.

[33] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott E. Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. *CoRR*, abs/1409.4842, 2014.

[34] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *International Conference on Learning Representations*, 2014.

[35] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *CoRR*, abs/1706.03762, 2017.

[36] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. Xlnet: Generalized autoregressive pretraining for language understanding. *CoRR*, abs/1906.08237, 2019.

[37] T. Young, D. Hazarika, S. Poria, and E. Cambria. Recent trends in deep learning based natural language processing [review article]. *IEEE Computational Intelligence Magazine*, 13(3):55–75, 2018.

[38] Xingyu Zeng, Wanli Ouyang, Bin Yang, Junjie Yan, and Xiaogang Wang. Gated bi-directional cnn for object detection. In *ECCV*, 2016.

[39] Zhengxia Zou, Zhenwei Shi, Yuhong Guo, and Jieping Ye. Object detection in 20 years: A survey. *CoRR*, abs/1905.05055, 2019.

[40] Hande Çelikkanat, Güner Orhan, Nicolas Pugeault, Frank Guerin, Erol Şahin, and Sinan Kalkan. Learning context on a humanoid robot using incremental latent dirichlet allocation. *IEEE Transactions on Cognitive and Developmental Systems*, 8(1):42–59, March 2016.