Multi-view Vision Transformers for Object Detection

Brian K. S. Isaac-Medina*, Chris G. Willcocks*, Toby P. Breckon*†
Department of {*Computer Science, †Engineering}, Durham University, Durham, UK

***Abstract*—Object detection has been thoroughly investigated during the last decade using deep neural networks. However, the inclusion of additional information given by multiple concurrent views of the same scene has not received much attention. In scenarios where objects may appear in obscure poses from certain view points, the use of differing simultaneous views can improve object detection. Therefore, we propose a multi-view fusion network to enrich the backbone features of standard object detection architectures across multiple source and target view points. Our method consists of a transformer decoder for the target view that combines the remaining source views feature maps. In this way, the feature representation of the target view can aggregate feature information from the source view through attention. Our architecture is detector-agnostic, meaning it can be applied across any existing detection backbone. We evaluate performance using YOLOX, Deformable DETR and Swin Transformer baseline detectors, comparing standard single view performance against the addition of our multi-view transformer architecture. Our method achieves a 3% increase of the COCO AP over a four view X-ray security dataset and a slight 0.7% increase on a seven view pedestrian dataset. We demonstrate that the integration of different views using attention-based networks improves the detection performance of multi-view datasets.**[1]

I. Introduction

Object detection is a fundamental task in computer vision, comprising the localisation of objects of interest within an image. Recent advances in deep neural networks have made it possible to achieve high-accuracy, real-time automatic object detection. Multi-view object detection refers to localising objects of interest given multiple images of the same scene where the view points of each image may be either fully or partially overlapping. In this context, these views can be used in conjunction to improve detection performance but the investigation of deep neural network architectures that specifically exploit this condition remains limited.

Modern detectors consist of three subnetworks: backbone, neck and head. The backbone is responsible of extracting the feature maps and are usually taken from high accuracy image classification networks, such as VGG [1], ResNet [2] and Darknet [3]. Some detectors include a subnetwork, sometimes called the neck, that is used to aggregate features from different layers of the backbone. The head of the detector localises the objects based on the feature maps from the backbone (or neck). A trend that is arising in the computer vision context is the implementation of the Transformer architecture proposed by Vaswani *et al.* [4]. The Transformer is an attention based network that has been the dominant approach in sequence to sequence tasks [5]–[7]. It consists of an encoder that obtains a representation of the input, and a decoder that takes the output of the encoder and generates the target sequence in an autoregressive fashion. The basic building block of the transformer encoder is a self-attention layer followed by a feed forward layer. Similarly, the Transformer decoder has a self-attention layer and a feed forward layer, but it also includes an additional attention layer where the source sequence is the encoder output. Carion *et al.* [8] proposed the Detection Transformer (DETR), the first architecture that implements a Transformer for object detection, using it as the head of the detector. Zhu *et al.* [9] further improved DETR by using deformable attention. A recent successful implementation of the Transformer for image classification is the Vision Transformer (ViT) by Dosovitskiy *et al.* [10] and its subsequent improvements [11], [12].

In certain circumstances, object detection using multiple concurrent views of the same scene is possible. In this context, detection accuracy is evaluated on each view independently, although objects can be predicted using the views jointly. This is of interest in scenarios where objects can be highly occluded in one view, but are clearer in another view, such as in multi-camera visual surveillance, autonomous vehicle sensing solutions and multi-view X-ray security screening. Although some works have addressed multi-view object detection [13]–[16], detailed consideration of this task remains fairly limited. Furthermore, the use of modern architectures based on attention has not been investigated thoroughly, leading us to propose a novel architecture based on a Transformer decoder that uses the feature representations across multiple concurrent views to improve detection accuracy.

In this work we address multi-view object detection by using such a Transformer based architecture to combine the intermediate features from multiple concurrent views using the backbone of a standard object detection architecture. The fusion of these features is carried out by a Transformer decoder using the target view feature vectors as queries attending to the features from a source view. In this sense, the feature representation is aggregated with the information from other views, making it aware of the 3D scene geometry. We apply the Transformer decoder to each view, so all views are target and source at the same time. For scenarios with more than two views, we propose to account for the feature maps of each of the source views via concatenation. We call our method Multi-view Vision Transformers (MVViT).

[1]Code available at https://github.com/KostadinovShalon/MVViT

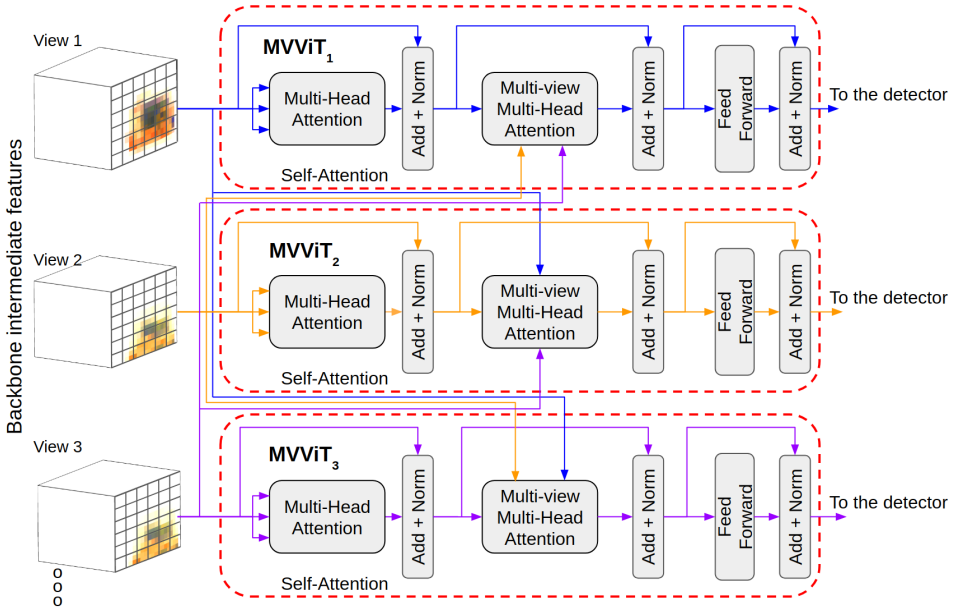


Fig. 1. MVViT for Object Detection: architectural design and overview.

Our key contributions are as follows:

- A novel Transformed-based architecture for multi-view object detection. Our method aggregates the feature representation of each view hence constructing a joint feature representation with awareness of the underlying 3D scene geometry.
- Consideration of three object detection architectures, YOLOX [17], Deformable DETR [9] and Swin Transformers [18], where our MVViT is integrated. It is shown that MVViT could improve multi-view object detection in both cases, demonstrating that it is detector agnostic.
- Improved multi-view object detection performance compared to a single view baseline, for both a multi-camera surveillance dataset (+0.7% COCO AP, +0.7% COCO $AP_{0.5}$) and an X-ray security imagery dataset (+3.0% COCO AP, +1.9% COCO $AP_{0.5}$).

II. Related Work

We review recent work on general multi-view object detection (Section II-A) and specifically Transformer architectures for object detection (Section II-B).

A. Multi-view Object Detection

Recent work explicitly addressing multi-object detection using contemporary detection architectures is limited [13]–[16], [19], [20]. Nassar *et al.* [19] apply a convolutional neural network that takes multi-view images and corresponding geolocation information as inputs and uses a joint loss function considering all views, resulting in an increase of the detection mAP by up to 27.8%. A different approach by Isaac-Medina *et al.* [14] apply a post-processing algorithm to eliminate detections that do not lie in the epipolar line between two views using the probability distribution of objects centroids, improving the overall detection mAP by 2.8%. With a similar application context to one of our evaluation cases, Steitz *et al.* [13] investigate merging features from multi-view X-ray baggage imagery using a 3D pooling layer and rely on geometric constraints that result from multiple 2D detection projections processed through a 3D region proposal network and a 3D region-based alignment layer to achieve improved average precision (+6.73% single class, firearms).

B. Transformers for Object Detection

The seminal Transformer based detector, DETR [8], comprises a ResNet as backbone and a Transformer head. Features are flattened and given a positional encoding that is fed into the encoder whilst the decoder, on the other hand, is fed with a fixed number N of learnable object queries that uses the encoded feature maps to predict the object instances. This process is done in parallel, contrary to the autoregressive nature of the Transformer architecture used in transduction models. In contrast to earlier detection architectures, DETR eliminates the need of anchor boxes by using a set-based loss, comparing the N predicted boxes to the ground truth (where the number of ground truth boxes is always lower than N). Boxes that are not paired with a ground truth object are assigned to a special background class. Vision Transformers (ViT) can similarly be used as feature extractors [10]. ViT divides the image into patches that are treated as an input sequence. The addition of a class token prepended to the input patches is used to learn the class by attending to every patch element. Beal *et al.* [21] show that ViT can be used as a backbone for object detection. Furthermore, Liu *et al.* [18] improve the detection accuracy with the Swin Transformer, adding multi-scale feature maps and reducing the ViT complexity from $\mathcal{O}(n^2)$ to $\mathcal{O}(n)$ by implementing

a shifted-window self-attention pattern. A recent work from Hou and Zheng [16] addresses multi-view pedestrian detection by using a DETR architecture with multi-view attention. In order to account for spatial consistency, they use a projective transform to the common ground plane.

III. Multi-view Vision Transformers

This work implements the Transformer decoder architecture to leverage from multiple view points feature maps to create a feature representation as awareness of the underlying 3D scene geometry. Our method, the Multi-view Vision Transformer (MVViT), acts as an extra layer within the backbone of the existing baseline detection architecture and it is depicted in Figure 1.

For each view $i = 1, ..., v$, MVViT applies a Transformer decoder taking the intermediate feature map $z_i \in \mathbb{R}^{W' \times H' \times C}$ as input and the remaining views feature maps z_j, $j \neq i$ as source views for the attention layer (Figure 1a). Following the ViT architecture, each decoder that comprises MVViT is composed of a multi-head self-attention layer, a multi-head attention module and a feed forward network consisting on two linear layers with internal dimension d_f. All of the sub-modules use residual connection followed by layer normalisation [22].

The attention mechanism, which is the basic building block of Transformers, can be described as a weighted sum based on a similarity function. Given N query d_k dimensional vectors embedded in the matrix $Q \in \mathbb{R}^{N \times d_k}$ and M pairs of key and value matrices $K \in \mathbb{R}^{M \times d_k}$ and $V \in \mathbb{R}^{M \times d_v}$ of d_k and d_v dimensional vectors, the attention mechanism is described as:

$$Attention(Q, K, V) = sim\left(Q, K\right) V\,, \tag{1}$$

where $sim(\cdot, \cdot)$ is a similarity function. A popular choice for the similarity function is the scaled dot product followed by a $softmax$ operation, that is:

$$sim(Q, K) = softmax\left(\frac{QK^{\intercal}}{\sqrt{d_k}}\right)\,. \tag{2}$$

Transformers define a multi-head attention (MHA) mechanism, where the attention inputs are linearly projected h times and attention is applied on each projection. This can be written as:

$$\begin{aligned} MHA(Q, K, V) &= concat\left(head_1, ..., head_h\right) \mathbf{W}^O\,, \\ head_i &= Attention\left(Q\mathbf{W}_i^Q, K\mathbf{W}_i^K, V\mathbf{W}_i^V\right),\ i = 1, ..., h\,, \end{aligned} \tag{3}$$

where $W_i^Q \in \mathbb{R}^{d_k \times d_m}$, $W_i^K \in \mathbb{R}^{d_k \times d_m}$, $W_i^V \in \mathbb{R}^{d_v \times d_m}$ and $W^O \in \mathbb{R}^{hd_m \times d_k}$ are learnable linear projections.

The feature map z_i can be seen as a $W' \times H'$ grid of feature vectors that serve as a sequence input for the decoder. These feature vectors are the object queries in our attention functions and an attention map of the source view is obtained for each of them. To achieve this, we modify the original implementation of the Transformer to use batched matrix multiplications in Equation (3) instead of being flattened to a 1D sequence.

In order to account for cases with more than one source view, we concatenate the source views in the feature dimension $\mathcal{V}_i = concat(\{z_j\}_{j \neq i}) \in \mathbb{R}^{W' \times H' \times (v-1)C}$ and apply MHA, that is:

$$MVMHA(z_i, \{z_j\}_{j \neq i}) = MHA\left(z_i, \mathcal{V}_i, \mathcal{V}_i\right)\,. \tag{4}$$

In this context, the target view attends to the source views at the same time, making it possible to vanish the attention from views where object instances do not appear in a source view overlapping field of view.

IV. Evaluation

Our evaluation is based on two different multi-view datasets (Section IV-A), with implementation details presented for repeatability (Section IV-B) and measured using the MS-COCO detection metrics [23].

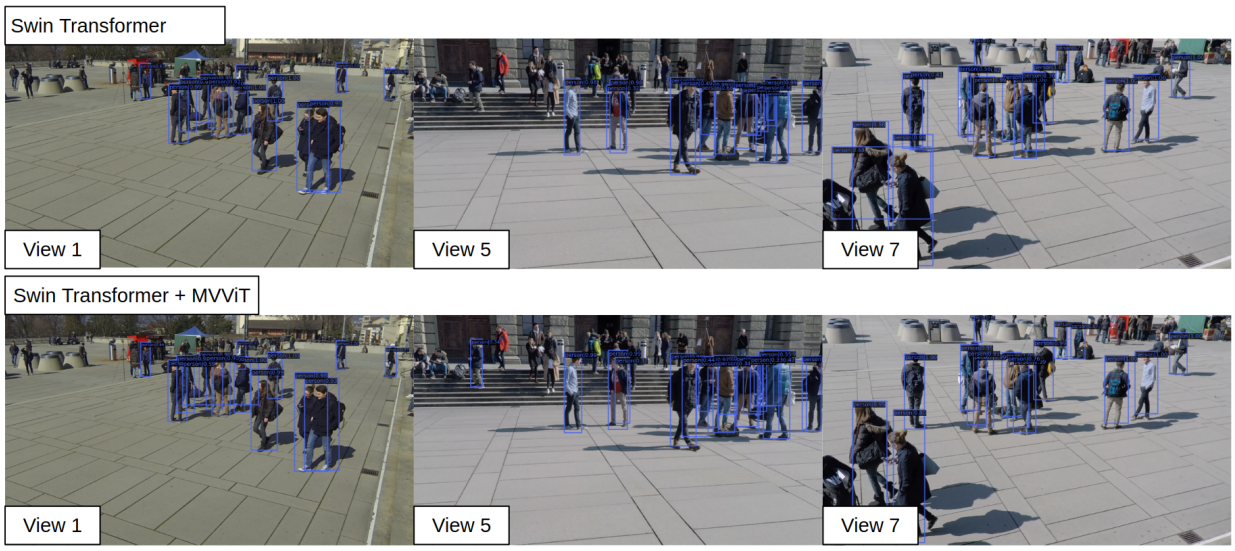
A. Dataset

Wildtrack: the Wildtrack seven-camera HD dataset [24] comprises a set of 7 outdoors concurrent videos from different points of view with only one class. This dataset includes scenarios where instances may appear in one view but not in the other. A total of 2,240 images and 33,962 object instances accounting for all views were used for training and 560 images and 8,571 object instances for validation.

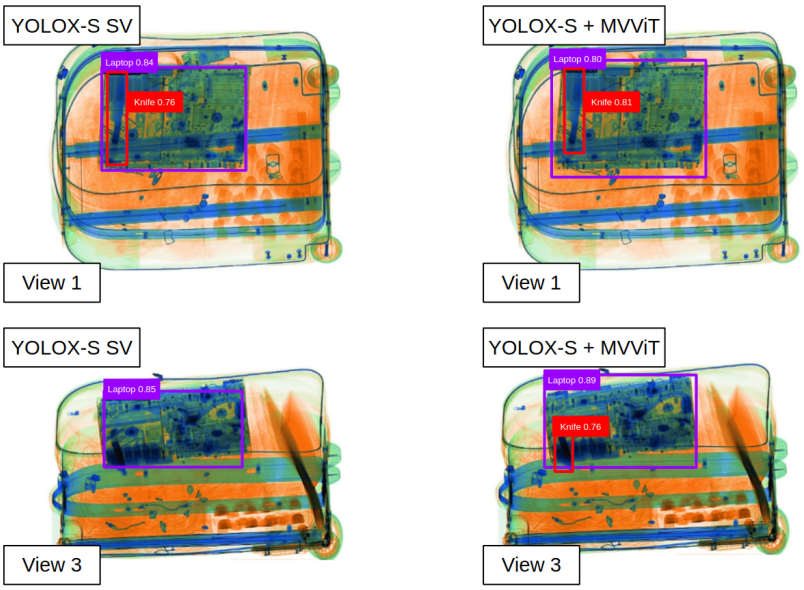
X-ray-Quad: we use false-coloured X-ray cabin baggage security imagery from a Smith Detection security scanner with four views. A total of 10,112 images were scanned and four object categories were identified (4,260 firearms, 2,376 laptops, 4,736 knives and 664 cameras). We used a split of 80% for training and 20% for testing. To assess the impact of the number of viewpoints, a partition X-ray-Dual is also assessed, with only two perpendicular views.

B. Implementation details

In order to assess the performance of MVViT in different detectors, YOLOX-S [17], Deformable DETR [9] and Swin Transformer [18] architectures are used as baselines. The Swin Transformer backbone is used in conjunction with a Faster-RCNN architecture [25], similarly to the original work. MixUp, Mosaic and Random Affine augmentations were removed in the YOLOX-S implementation, since they are not multi-view consistent. In order to avoid an increased performance due to having larger datasets in the implementation of the MVViT, the same datasets were used when comparing to the single-view (sv) baselines, with the difference that different views from the same when are used to create the 3D aware features in MVViT layers. Input images for YOLOX-S are square padded (with a white background for X-ray datasets and a grey background for the Wildtrack dataset) and resized to 640×640, while the input images for Deformable DETR and Swin Transformer are kept to a maximum size of 1333 for the X-ray-Dual dataset and 800 for X-ray-Quad and Wildtrack datasets. MVViT is applied before the fourth CSP block of the YOLOX-S backbone (Modified CSPNet v5 [26]), after the conv4 block of the Deformable DETR



(a)



(b)

Fig. 2. Exemplar multi-view object detections contrasting single view (SV) and multi-view vision transformers (MVViT) performance for the Wildtrack dataset (a) and the X-ray-Quad dataset (b).

backbone (ResNet-50) and before the fourth stage swin block of the Swin Transformer. We use 8 heads for the MHA modules, internal decoder dimension $d_k = 512$ and feed forward dimension d_f = 2048. ReLU activations are used and a dropout with a rate of 0.1 is applied after each MVViT layer. The model is trained using Stochastic Gradient Descent for YOLOX-S and AdamW optimization [27] for Deformable DETR and Swin Transformer. A batch size of 6 images per view is used to train YOLOX for both X-ray datasets and 2 images per view for the Wildtrack dataset. On the other hand, a batch size of 2 images per view is used to train Deformable DETR for the X-ray-Dual dataset and 1 image per view for both X-ray-Quad and Wildtrack datasets. Finally, a batch size of 4 is used for both X-ray datasets and 3 for the Wildtrack dataset. MMDetection [28] framework was used with the original training and optimisation settings for the three detectors. Models were trained using an NVIDIA Tesla V100.

V. Results

The statistical performance of MVViT compared with single view detection is presented in Table I. For the X-ray-Dual and X-ray-Quad datasets, results for each class, as well as for all classes are presented. MVViT outperforms single view detection in the Wildtrack dataset only for the Swin Transformer architecture, with a slight increment of 0.7% in the COCO AP, while small decrements are seen in the

TABLE I
SINGLE VIEW VS MVVIT DETECTION - STATISTICAL PERFORMANCE

Dataset	Architecture	Category	Method	AP	$AP_{0.5}$	$AP_{0.75}$	AP_S	AP_M	AP_L	AR	AR_S	AR_M	AR_L
Wildtrack	YOLOX-S	Person	SV	**0.383**	**0.773**	**0.334**	-	**0.299**	**0.412**	**0.492**	-	**0.453**	**0.514**
			MVViT	0.370	0.764	0.301	-	0.274	0.409	0.471	-	0.368	0.511
	Deformable DETR	Person	SV	**0.417**	**0.772**	**0.388**	-	**0.335**	**0.450**	**0.587**	-	**0.515**	**0.613**
			MVViT	0.401	0.761	0.368	-	0.318	0.432	0.577	-	0.513	0.601
	Swin Transformer	Person	SV	0.367	0.780	0.267	-	0.266	0.408	0.489	-	0.449	0.508
			MVViT	**0.374**	**0.784**	**0.300**	-	**0.274**	**0.419**	**0.503**	-	**0.463**	**0.522**
X-ray-Dual	YOLOX-S	Firearm	SV	0.624	0.939	0.730	-	0.633	0.709	0.674	-	0.666	0.796
			MVViT	**0.695**	**0.972**	**0.830**	-	**0.704**	**0.747**	**0.735**	-	**0.729**	**0.832**
		Knife	SV	0.242	0.540	0.169	0.093	0.280	**0.048**	0.349	0.118	0.366	**0.425**
			MVViT	**0.285**	**0.619**	**0.229**	**0.098**	**0.325**	0.033	**0.383**	**0.147**	**0.402**	0.350
		Laptop	SV	0.710	0.981	**0.869**	-	-	0.710	0.762	-	-	0.762
			MVViT	**0.723**	**0.990**	0.868	-	-	**0.723**	**0.771**	-	-	**0.771**
		Camera	SV	0.566	0.867	0.672	-	**0.800**	0.562	0.624	-	**0.800**	0.622
			MVViT	**0.632**	**0.896**	**0.811**	-	**0.800**	**0.630**	**0.688**	-	**0.800**	**0.686**
		All	SV	0.535	0.832	0.610	0.093	0.571	0.507	0.602	0.118	0.611	0.651
			MVViT	**0.583**	**0.869**	**0.685**	**0.098**	**0.610**	**0.533**	**0.644**	**0.147**	**0.644**	**0.660**
	Deformable DETR	Firearm	SV	0.674	**0.968**	0.816	-	0.685	0.667	0.741	-	0.735	**0.832**
			MVViT	**0.680**	0.960	**0.817**	-	**0.689**	**0.679**	**0.743**	-	**0.738**	0.818
		Knife	SV	**0.251**	**0.626**	**0.142**	**0.139**	**0.285**	0.033	**0.428**	**0.162**	**0.450**	0.325
			MVViT	0.237	0.598	0.116	0.112	0.269	**0.159**	0.423	0.135	0.444	**0.425**
		Laptop	SV	0.803	0.990	0.947	-	-	0.803	0.855	-	-	0.855
			MVViT	**0.839**	**0.995**	**0.953**	-	-	**0.839**	**0.879**	-	-	**0.879**
		Camera	SV	**0.646**	**0.918**	**0.837**	-	0.700	**0.647**	**0.738**	-	0.700	**0.738**
			MVViT	0.601	0.847	0.739	-	**0.800**	0.600	0.723	-	**0.800**	0.722
		All	SV	**0.593**	**0.876**	**0.686**	**0.139**	0.557	0.537	0.691	**0.162**	0.628	0.688
			MVViT	0.589	0.850	0.656	0.112	**0.586**	**0.569**	**0.692**	0.135	**0.661**	**0.711**
	Swin Transformer	Firearm	SV	0.698	**0.989**	0.873	-	0.705	**0.747**	0.741	-	0.735	**0.821**
			MVViT	**0.702**	**0.989**	**0.898**	-	**0.711**	0.718	**0.746**	-	**0.741**	0.818
		Knife	SV	0.419	0.821	0.370	0.189	0.449	0.311	0.493	0.279	0.507	0.675
			MVViT	**0.428**	**0.847**	**0.381**	**0.219**	**0.458**	**0.317**	**0.499**	**0.315**	**0.512**	**0.700**
		Laptop	SV	**0.833**	**0.991**	**0.976**	-	-	**0.833**	**0.876**	-	-	**0.876**
			MVViT	0.820	0.987	**0.976**	-	-	0.820	0.864	-	-	0.864
		Camera	SV	**0.680**	0.967	**0.836**	-	**0.700**	**0.681**	0.721	-	**0.700**	0.722
			MVViT	0.668	**0.976**	0.806	-	**0.700**	0.669	**0.723**	-	**0.700**	**0.723**
		All	SV	**0.657**	0.942	0.764	0.189	0.618	**0.643**	**0.708**	0.279	0.648	0.773
			MVViT	0.655	**0.950**	**0.765**	**0.219**	**0.623**	0.631	**0.708**	**0.315**	**0.651**	**0.776**
X-ray-Quad	YOLOX-S	Firearm	SV	0.734	0.973	0.884	-	0.742	0.787	0.767	-	0.759	**0.845**
			MVViT	**0.748**	**0.979**	**0.907**	-	**0.760**	**0.790**	**0.779**	-	**0.774**	0.838
		Knife	SV	0.353	0.693	0.331	0.150	0.392	0.022	0.447	**0.188**	0.459	**0.325**
			MVViT	**0.379**	**0.732**	**0.346**	**0.152**	**0.414**	**0.051**	**0.461**	0.178	**0.475**	**0.325**
		Laptop	SV	0.765	0.987	0.909	-	-	0.765	0.812	-	-	0.812
			MVViT	**0.806**	**0.992**	**0.946**	-	-	**0.806**	**0.844**	-	-	**0.844**
		Camera	SV	0.639	0.899	0.778	-	**0.800**	0.639	0.688	-	**0.800**	0.687
			MVViT	**0.678**	**0.926**	**0.840**	-	0.700	**0.678**	**0.726**	-	0.700	**0.726**
		All	SV	0.623	0.888	0.726	0.150	**0.644**	0.553	0.678	**0.188**	**0.673**	0.667
			MVViT	**0.653**	**0.907**	**0.760**	**0.152**	0.625	**0.581**	**0.703**	0.178	0.650	**0.683**
	Deformable DETR	Firearm	SV	**0.726**	0.978	**0.885**	-	**0.740**	0.711	0.784	-	0.779	0.832
			MVViT	0.724	**0.997**	0.884	-	0.735	**0.720**	**0.788**	-	**0.782**	**0.854**
		Knife	SV	**0.352**	0.751	**0.286**	0.123	**0.390**	**0.143**	0.501	**0.163**	0.517	0.375
			MVViT	0.347	**0.760**	0.261	**0.140**	0.386	0.085	**0.506**	0.161	**0.521**	**0.438**
		Laptop	SV	0.847	0.984	0.970	-	-	0.847	0.896	-	-	0.896
			MVViT	**0.859**	**0.993**	**0.977**	-	-	**0.859**	**0.912**	-	-	**0.912**
		Camera	SV	0.646	0.896	0.772	-	**0.800**	0.647	**0.773**	-	**0.800**	**0.773**
			MVViT	**0.674**	**0.909**	**0.836**	-	0.700	**0.674**	0.772	-	0.700	**0.773**
		All	SV	0.643	0.902	0.728	0.123	**0.643**	**0.587**	0.738	**0.163**	**0.699**	0.719
			MVViT	**0.651**	**0.910**	**0.739**	**0.140**	0.607	0.585	**0.745**	0.161	0.668	**0.744**
	Swin Transformer	Firearm	SV	**0.742**	**0.990**	0.932	-	**0.755**	**0.770**	**0.780**	-	**0.774**	**0.846**
			MVViT	0.738	**0.990**	**0.934**	-	0.751	0.758	0.779	-	**0.774**	0.836
		Knife	SV	0.503	**0.904**	0.515	**0.288**	0.537	0.290	0.566	**0.359**	0.574	**0.738**
			MVViT	**0.508**	0.901	**0.531**	0.254	**0.539**	**0.305**	**0.569**	0.302	**0.580**	0.713
		Laptop	SV	0.863	0.990	0.977	-	-	0.863	0.903	-	-	0.903
			MVViT	**0.873**	**0.992**	**0.982**	-	-	**0.873**	**0.908**	-	-	**0.908**
		Camera	SV	0.669	0.918	**0.854**	-	**0.800**	0.669	**0.720**	-	**0.800**	**0.720**
			MVViT	**0.671**	**0.927**	0.814	-	**0.800**	**0.670**	0.709	-	**0.800**	0.708
		All	SV	0.694	0.950	**0.819**	**0.288**	**0.697**	0.648	**0.742**	**0.359**	0.716	**0.802**
			MVViT	**0.698**	**0.952**	0.815	0.254	**0.697**	**0.651**	0.741	0.302	**0.718**	0.791

YOLOX-S and Deformable DETR architectures. As seen in Figure 2a, this dataset has many occlusions across the different views, which imposes an additional challenge for MVViT. The results for the X-ray-Dual dataset show an improvement when training with YOLOX-S, with an increase of 4.8% on the AP metric and 6.7% on the $AP_{0.5}$ metric. The performance gets slightly worsen when training Deformable DETR and Swin Transformer with MVViT on the X-ray-Dual dataset. This effect may be caused by the fact that these architectures obtain high precision for almost all classes (except for knives). The remaining not-detected objects present a significant detection challenge against which further advancement may adversely impact overall network performance across other classes. On the other hand, the precision of the three detectors improves when using MVViT on the X-ray-Quad dataset, with an increase of 3% AP, 1.9% $AP_{0.5}$ with the YOLOX-S architecture, and small increments of 0.8% and 0.4% on the AP when using the Deformable DETR and Swin Transformer architectures. These results indicates that the performance can be increased if the model integrates features from different views, having a better performance when more views are used. However, as seen in the performance on the Wildtrack dataset, it is sensitive to highly occluded data.

Figure 2a shows an example of single view detection compared with our method for the Wildtrack dataset (views 1, 5 and 7). Some duplicates and false positives can be noted when doing single view detection (bottom left people in view 7). When using the MVViT module, the detector is able to remove these instances, along with detecting other missed people. However, some new duplicates can be seen in crowded areas such as the centre of view 1 and right of view 5. This indicates that although our network is 3D aware, it is difficult to cope with highly occluded scenarios. Figure 2b shows detection examples for the X-ray-Dual dataset. In this case, a missed knife is detected in view 3 when using MVViT. This missed knife in view 3 is highly occluded since it is behind a laptop. However, since it is better seen in view 1, the aggregated features in view 3 using MVViT allow for being detected. We hypothesise that the overlapping nature of transmission images, such as X-ray, puts an additional difficulty since feature vectors may contain information from more than one class, which can be alleviated with a MVViT layer.

Finally, we look qualitatively at the attention map in the source view given a feature vector in the target view. Figure 3 shows the attention mechanism in the source view given a feature vector from the target view which spatial location is represented by a red square. The right image of Figure 3 shows the attention mechanism for the X-ray dataset. When we look at the attention for a feature vector located at a firearm, it is noted that the attention in the source view is focused in different parts of the gun. This is true regarding the view that is used as the target view. However, it is also noted that MVViT is not applying attention in all the object instance but only in small localised regions. A model that captures the shape of the object instances in the source view through attention could improve object detection performance and remains as an area for future work.

Fig. 3. Attention mechanism in MVViT: the left image is the target view where the red square represents the location of the feature vector that gets the attention whilst the right image is the source view.

VI. Conclusions

In this work we present multi-view vision transformers (MVViT), a novel architecture that uses attention to aggregate the feature maps across multiple concurrent views within a standard detection architecture. MVViT takes as input the feature maps of a target view and applies attention on the feature maps of the other concurrent source views to create 3D scene geometry aware feature representations.

We investigate the performance of MVViT for a quad-view X-ray security scanner imagery dataset, obtaining an overall COCO AP increase of 4.8% for two views and 3% with four views using the YOLOX-S detector. Additionally, a slight increase in the performance is also observed with four views and using the Deformable DETR and Swin Transformer architectures. A decrease on the performance was observed when using a Deformable DETR and Swin Transformer detectors for the two views X-ray dataset, apparently caused by the detectors already reaching the best performance. It is also observed that our method increases the AP of a seven-view pedestrian dataset by 0.7% with the Swin Transformer architecture, but it fails with YOLOX-S and Deformable DETR. This indicates that the highly occluded nature of the Wildtrack dataset imposes a greater challenge for MVViT. Additionally, we look at the attention maps in the source views with respect a feature vector in the target view. It is further observed that the attention in the source view matches the corresponding feature vector from the target view, although it does not capture the whole region of interest. Future work will investigate the role of the depth in MVViT and its application to different datasets and detector models, as well as new detection models based on attention.

Acknowledgments

This work is partially supported by the Mexican Council of Science and Technology (CONACyT).

REFERENCES

[1] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *International Conference on Learning Representations*, 2015.

[2] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[3] J. Redmon and A. Farhadi, "Yolo9000: better, faster, stronger," in *Proc. Conference on Computer Vision and Pattern Recognition*, 2017, pp. 7263–7271.

[4] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[5] S. Takase and S. Kiyono, "Lessons on parameter sharing across layers in transformers," *arXiv preprint arXiv:2104.06022*, 2021.

[6] Q. Xu, A. Baevski, T. Likhomanenko, P. Tomasello, A. Conneau, R. Collobert, G. Synnaeve, and M. Auli, "Self-training and pre-training are complementary for speech recognition," in *International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2021, pp. 3030–3034.

[7] H. Akbari, L. Yuan, R. Qian, W.-H. Chuang, S.-F. Chang, Y. Cui, and B. Gong, "Vatt: Transformers for multimodal self-supervised learning from raw video, audio and text," *Advances in Neural Information Processing Systems*, vol. 34, 2021.

[8] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *Proc. European Conference on Computer Vision*. Springer, 2020, pp. 213–229.

[9] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, "Deformable detr: Deformable transformers for end-to-end object detection," in *International Conference on Learning Representations*, 2020.

[10] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," in *International Conference on Learning Representations*, 2020.

[11] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou, "Training data-efficient image transformers & distillation through attention," in *International Conference on Machine Learning*. PMLR, 2021, pp. 10 347–10 357.

[12] H. Touvron, M. Cord, A. Sablayrolles, G. Synnaeve, and H. Jégou, "Going deeper with image transformers," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 32–42.

[13] J.-M. O. Steitz, F. Saeedan, and S. Roth, "Multi-view X-ray R-CNN," in *Proc. German Conference on Pattern Recognition*. Springer, 2018, pp. 153–168.

[14] B. K. S. Isaac-Medina, C. G. Willcocks, and T. P. Breckon, "Multi-view object detection using epipolar constraints within cluttered x-ray security imagery," in *Proc. International Conference on Pattern Recognition*. IEEE, 2021, pp. 9889–9896.

[15] Y. Hou, L. Zheng, and S. Gould, "Multiview detection with feature perspective transformation," in *Proc. European Conference on Computer Vision*. Springer, 2020, pp. 1–18.

[16] Y. Hou and L. Zheng, "Multiview detection with shadow transformer (and view-coherent data augmentation)," in *Proceedings of the 29th ACM International Conference on Multimedia*, 2021, pp. 1673–1682.

[17] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, "Yolox: Exceeding yolo series in 2021," *arXiv preprint arXiv:2107.08430*, 2021.

[18] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proc. International Conference on Computer Vision*, 2021, pp. 10 012–10 022.

[19] A. S. Nassar, S. Lefèvre, and J. D. Wegner, "Simultaneous multi-view instance detection with learned geometric soft-constraints," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 6559–6568.

[20] J. Deng and K. Czarnecki, "Mlod: A multi-view 3d object detection based on robust feature fusion method," in *Intelligent Transportation Systems Conference*. IEEE, 2019, pp. 279–284.

[21] J. Beal, E. Kim, E. Tzeng, D. H. Park, A. Zhai, and D. Kislyuk, "Toward transformer-based object detection," *arXiv preprint arXiv:2012.09958*, 2020.

[22] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *arXiv preprint arXiv:1607.06450*, 2016.

[23] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *Proc. European conference on computer vision*. Springer, 2014, pp. 740–755.

[24] T. Chavdarova, P. Baqué, S. Bouquet, A. Maksai, C. Jose, T. Bagautdinov, L. Lettry, P. Fua, L. Van Gool, and F. Fleuret, "Wildtrack: A multi-camera hd dataset for dense unscripted pedestrian detection," in *Proc. Conference on Computer Vision and Pattern Recognition*, June 2018.

[25] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2016.

[26] C.-Y. Wang, H.-Y. M. Liao, Y.-H. Wu, P.-Y. Chen, J.-W. Hsieh, and I.-H. Yeh, "Cspnet: A new backbone that can enhance learning capability of cnn," in *Proc. Conference on Computer Vision and Pattern Recognition Workshops*, 2020, pp. 390–391.

[27] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *International Conference on Learning Representations*, 2018.

[28] K. Chen, J. Wang, J. Pang, Y. Cao, Y. Xiong, X. Li, S. Sun, W. Feng, Z. Liu, J. Xu, Z. Zhang, D. Cheng, C. Zhu, T. Cheng, Q. Zhao, B. Li, X. Lu, R. Zhu, Y. Wu, J. Dai, J. Wang, J. Shi, W. Ouyang, C. C. Loy, and D. Lin, "MMDetection: Open mmlab detection toolbox and benchmark," *arXiv preprint arXiv:1906.07155*, 2019.