Neural Attention-driven Non-Maximum Suppression for Person Detection

Charalampos Symeonidis, Ioannis Mademlis, Ioannis Pitas and Nikos Nikolaidis
Department of Informatics, Aristotle University of Thessaloniki, Thessaloniki, Greece

***Abstract*—Non-maximum suppression (NMS) is a post-processing step in almost every visual object detector. NMS aims to prune the number of overlapping detected candidate regions-of-interest (RoIs) on an image, in order to assign a single and spatially accurate detection to each object. The default NMS algorithm (GreedyNMS) is fairly simple and suffers from severe drawbacks, due to its need for manual tuning. A typical case of failure with high application relevance is pedestrian/person detection in the presence of occlusions, where GreedyNMS doesn't provide accurate results. This paper proposes an efficient deep neural architecture for NMS in the person detection scenario, by capturing relations of neighboring RoIs and aiming to ideally assign precisely one detection per person. The presented Seq2Seq-NMS architecture assumes a sequence-to-sequence formulation of the NMS problem, exploits the Multihead Scale-Dot Product Attention mechanism and jointly processes both geometric and visual properties of the input candidate RoIs. Thorough experimental evaluation on three public person detection datasets shows favourable results against competing methods, with acceptable inference runtime requirements.**

***Index Terms*—Non-Maximum Suppression, Object Detection, Scaled-Dot Product Attention, Sequence-to-Sequence Learning, Person Detection, Deep Neural Networks**

I. Introduction

Non-Maximum Suppression (NMS) is a final refinement step incorporated to almost every visual object detection framework, assigned the duty of merging/filtering any spatially overlapping detected Regions-of-Interest (RoIs), i.e., bounding boxes, which correspond to the same visible object on an image. The problem it attempts to solve arises from the tendency of many detectors to output multiple, neighbouring candidate object RoIs for a single visible object, due to their implicit sliding-window nature. Thus, an NMS algorithm processes the raw object detector outputs identified on an input image and attempts to filter out the duplicate RoIs.

The de facto dominant NMS method for object detection is GreedyNMS. It selects high-scoring detections and deletes less confident neighbours, since they most likely cover the same object. Its simplicity, speed and unexpectedly good behaviour in most cases make it competitive against proposed alternatives, since rapid execution is very important for NMS. An Intersection-over-Union (IoU) threshold determines which less-confident neighbors are suppressed by a detection. This fixed IoU threshold leads GreedyNMS to failure in certain cases. Too powerful a suppression, using a low threshold, may remove detections that cover different spatially overlapped objects, while a too high threshold may be unable to suppress duplicate detections.

Due to these limitations of traditional algorithms, modern Deep Neural Network (DNN)-based methods for performing NMS have emerged during the past few years. While some DNNs are assigned with auxiliary tasks complementing the original NMS scheme (e.g., estimate target density maps in order to apply dynamic suppression thresholding [1]), others provide a more straightforward solution (e.g., outputting a score for each candidate detection, thus indicating whether it corresponds to a "duplicate" detection or not [2]). The latter type of methods relies on building representations for each candidate detection, typically based on their corresponding geometric/spatial relations [2], while ignoring RoI visual appearance. This is either because CNN-based features can blur the boundaries between highly overlapping true positives and duplicates, or due to the difficulties DNNs are faced with when trying to extract accurate representations for highly occluded objects. However, evidence has recently surfaced indicating that appearance-based input may improve the performance of DNN-based NMS methods [3] [4], if that information is properly fused with the geometry-based input.

An additional issue stems from the fact that the NMS problem for object detection purposes is essentially sequential in nature. The output RoIs are sequentially processed by the common object detection evaluation protocols [5] [6], ordered according to the scalar confidence scores assigned to them by the NMS method. Similarly, the input candidate RoIs, i.e., the raw output of the object detector which is fed as input to the NMS algorithm, must also be ordered according to the initial confidence scores assigned to them by the detector. Thus, essentially, an NMS method actually decides whether a candidate RoI is duplicate, or not, based on the decisions it has previously taken for the preceding, higher-scoring candidate RoIs along the input sequence. However, to the best of our knowledge, NMS has not been previously explicitly formulated as a problem of processing sequences, thus related algorithms have not been applied to solving it.

Motivated by such issues of existing neural NMS approaches, this paper offers the following contributions:

- a novel reformulation of the NMS task for object detection as a sequence-to-sequence problem.
- a novel deep neural architecture for NMS, relying on the Scaled Dot-Product Attention mechanism, called *Seq2Seq-NMS*.

The source code is publicly available at: https://github.com/opendr-eu/opendr/tree/master/src/opendr/perception/object_detection_2d/nms/seq2seq_nms.

- a new, fast, efficient and GPU-based neural implementation of the low-level Frame Moments Descriptor (FMoD) [7], which is employed for feeding the proposed DNN with appearance-based representations of detected candidate RoIs.

The proposed method is highly applicable to the person/pedestrian detection task, where most NMS algorithms face difficulties in identifying individuals in the presence of occlusions. The majority of existing NMS methods target fast execution, but person detection requires a high degree of accuracy; this is critical for ensuring human safety in domains such as autonomous systems [8] [9] [10] [11] [12] [13]. Moreover, the visual appearance representation approach adopted by Seq2Seq-NMS, i.e., FMoD descriptors computed on edge maps of cropped candidate RoIs, is most accurate in cases where the visible silhouette of the target object class remains approximately identical in shape across the training and test images. This is true in the person detection case, bar abnormally extensive viewpoint variations across the employed dataset. Adopting FMoD, which has already proven its worth in NMS for person detection from aerial viewpoints [3], renders the applicability of the proposed method focused to similar scenarios.

Extensive quantitative evaluation using well-known metrics and public person detection datasets indicates favourable results in comparison to several competing NMS methods, both neural and non-neural, leading to state-of-the-art results. The source code is publicly available at: https://github.com/opendr-eu/opendr/tree/master/src/opendr/perception/object_detection_2d/nms/seq2seq_nms.

II. Related Work

NMS is the final step of typical object detection pipelines, thus this Section first briefly reviews state-of-the-art detectors. Subsequently, NMS algorithms and related loss functions are presented. Finally, the motivation behind the proposed method is discussed in the context of the existing approaches to NMS.

A. Object Detection

Object detection is a long-standing, fundamental problem in computer vision. Its task is to generate bounding boxes (in 2D pixel coordinates) for objects detected on an image that belong to prespecified object classes and to assign classification scores to them. Most of the early object detection algorithms [14] [15] relied mainly on local handcrafted descriptors and discriminative classifiers. The Deformable Part-based Model (DPM) [16] is a special case, where an object is represented by its component parts arranged in a deformable configuration. In [17], the authors designed a joint person detector, based on the DPM architecture, which overcomes the limitations imposed by frequent occlusions in real-world street scenes.

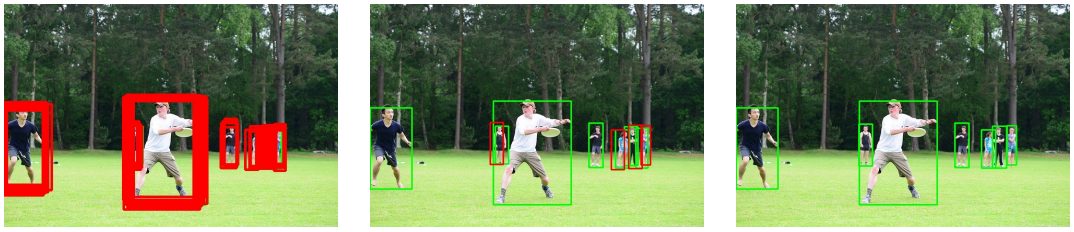
Object detection has been tremendously improved thanks to Deep Neural Networks (DNNs), with Convolutional Neural Networks (CNNs) being the most relevant architectures. DNN-based object detectors are usually grouped into two categories: two-stage and one-stage object detectors. Typically, the former ones (e.g., [18]) first create object proposals from input images, using a method such as selective search or a separate DNN, and then extract features from these proposals using CNNs. These features are then fed to a classifier that determines the existence and the class of any object in each proposal. Although two-stage detectors achieve state-of-the-art performance, their running speed is typically slow. One-stage object detectors, such as [19] [20] and [21] perform region proposal and object classification in a single, unified DNN. Initial regions are predefined bounding boxes with various scales and ratios placed densely on the image, which are generally referenced as anchors. From the initial anchors, the detectors find those that likely contain objects. Compared to two-stage detectors, their one-stage competitors are usually much faster, but less accurate.

B. Non-Maximum Suppression

The de facto standard in NMS for object detection is GreedyNMS [22]. It selects high-scoring detections and deletes less confident neighbours, since they most likely cover the same object. An Intersection-over-Union (IOU) threshold determines which less-confident neighboring detections are suppressed. It is a simple, well-known, but limited method, leading to several attempts for replacing it with much improved alternatives.

In Soft-NMS [23], a rescoring function decreases the score of neighboring less-confident detections, instead of completely eliminating them, achieving better precision and recall rates compared to GreedyNMS. The authors experiment with Gaussian and linear weighting functions, which both require a hyper-parameter tuning similar to GreedyNMS. In [24], the final coordinates of a detection are being reformulated as the weighted-average of the coordinates of all neighboring detections, given an IoU threshold. GossipNet [2] is a DNN designed to perform NMS, by processing the coordinates and scores of the detections. Overall, it jointly analyzes all detections in the image, so as not to directly prune them, but to rescore them. In [25], the authors replace the classification scores of candidate detections, used in GreedyNMS, with learned localization confidences to guide NMS towards preserving more accurately localized bounding boxes. In [4], an attention module is applied with the task to exploit relations between the input detections, in order to classify them as duplicate or not. [1] proposes Adaptive-NMS, a dynamic thresholding version of GreedyNMS. A relatively shallow neural network predicts a density map and sets adaptive IoU thresholds in NMS for different detections according to the predicted density. An accelerated NMS method has been proposed in [26], allowing higher inference times in exchange for a small performance drop, due to the large number of boxes that are likely to be over-suppressed.

GossipNet was modified in [3], for the specific case of person detection from aerial views, so as to jointly process visual appearance and geometric properties of candidate RoIs. The method exploited handcrafted descriptors encoding statistical RoI appearance characteristics, which were computed on the



(a) Raw RoIs/detections. (b) RoIs/detections after applying GreedyNMS at 0.5 IOU. (c) RoIs/detections after applying the proposed Seq2Seq-NMS method.

Fig. 1: Candidate RoIs/detections from Faster-RCNN in an image from the COCO dataset. Detections matched successfully to humans are colored green, while "incorrect" detections are colored red.

spatial distribution of edges detected within each RoI. These distributions acted as a discriminant factor for identifying complete vs partial object silhouettes, since the silhouette of any person seen from an aerial view is rather similar in shape.

More recently, [27] proposed *Distance-IoU* (DIoU), a new metric which can replace the typical IoU metric in GreedyNMS. This work suggested that the suppression procedure should take into account not only the overlap of two neighboring detections, but also the distances between their centers. Alternatively, Cluster-NMS was proposed in [28], i.e., a technique where NMS is performed by implicitly clustering candidate detections. Cluster-NMS can incorporate geometric factors to improve both precision and recall rates and can efficiently run on a GPU, achieving very fast inference runtimes.

C. Loss Functions for Bounding Box Regression

In DNN-based methods for visual object detection, prediction of spatially accurate RoIs/bounding boxes is enforced by an additional loss term during model training. The regressed RoI parameters are position, shape and scale, in terms of 2D pixel coordinates. These parameters are predicted either directly, or as offsets relative to "anchor boxes", in the case of anchor-based detectors. It is common to use the $\mathcal{L}_n$-norm for calculating the corresponding loss term (e.g., [18] [19] [20]). However, [29] indicates that the correlation between training with such $\mathcal{L}_n$-norm loss terms and improving test accuracy, as measured by the Intersection-over-Union (IoU) metric, is not strong at all. On the other hand, directly incorporating the IoU metric in a loss function would implicitly force the detector itself to also perform a rudimentary degree of NMS, but this is unsuitable for cases where two bounding boxes are non-overlapping, due to their zero loss gradient. Thus, [29] proposes the *Generalized-IoU* (GIoU) loss term, which handles similar scenarios but suffers from slow convergence and inaccurate regression. Thus, in [27], a loss term relying on the DIoU metric was formulated, by adding to the IoU loss a penalty based on the 2D center point coordinates of two bounding boxes. This was shown to converge faster than GIoU. [27] also proposed the *Complete-IoU* (CIoU) loss, an extension of DIoU with an additional term which can be tuned so as to impose aspect ratio consistency between two bounding boxes, thus leading to further increases in test accuracy.

D. Limitations of Existing Methods

State-of-the-art object detectors continue to require NMS as a final step [21], even when they use sophisticated loss functions for bounding box regression during training. A typical scenario showcasing the indispensability of a reliable NMS method is when object detection is performed on images with high levels of occlusions [1] [30]; ironically, this constitutes a challenge even to state-of-the-art NMS algorithms.

Although the geometric properties of candidate RoIs have been considerably exploited by various NMS approaches [2] [27] [28] [26], only a couple of methods [1] [4] [3] have attempted to take advantage of both visual appearance and geometric/spatial RoI information. Therefore, joint exploitation of appearance and geometry for NMS in object detection is underexplored. In addition, despite a vast amount of effort expended towards achieving short inference times [26] [27], since fast execution is an important aspect of NMS, one can easily identify real-world scenarios where a potential improvement in accuracy may equally matter (e.g., pedestrian/person detection in human safety-centric applications).

Despite the sequential nature of the NMS task in object detection, since at least the input candidate RoIs are always ordered according to their confidence score, no previous method has relied on formulating the problem as a sequence-to-sequence task. Thus, the recent rise of self-attention neural modules [31], capable of efficiently capturing interrelations within a sequence, has not yet significantly affected NMS algorithms. To the best of our knowledge, the only relevant method employing self-attention mechanisms is [4], tailored for the duplicate removal task and not for pure NMS. Thus, it does not perform free rescoring: an input candidate RoI which was assigned a low confidence score by the detector (e.g., due to occlusion) cannot be rescored higher by the duplicate removal DNN; only lower. An unconstrained NMS method exploiting the powerful self-attention neural mechanism has yet to emerge.

Out of the existing literature, the proposed method is most related to [2] [3] and [4]. Like GossipNet in [2], Seq2Seq-NMS approaches NMS as a rescoring problem. However, an optimized geometric representation for each candidate RoI is proposed here, slightly similar, but different and enriched compared to the GossipNet input descriptor. Like [3], Seq2Seq-NMS jointly processes visual and geometric representations

of the input candidate RoIs, using the FMoD descriptor [7] computed on edge maps of cropped detections. However, in this paper, the FMoD descriptor has been re-implemented neurally, leading to significant runtime gains thanks to GP-GPU-based parallel processing, while a novel deep neural architecture is proposed here, so as to exploit the sequence-to-sequence formulation, instead of relying on GossipNet. Finally, similarly to [4], the Seq2Seq-NMS architecture employs the powerful self-attention neural mechanism, but since the proposed method is a complete, free rescoring NMS DNN it is able to search for and fully exploit interrelations between the candidate RoI representations, without being constrained by the original confidence score assigned by the object detector.

III. Attention-driven Non-Maximum Suppression

In this paper, NMS for object detection is first reformulated as a sequence-to-sequence task. This approach is highly related to the evaluation criteria established in object detection [5] [6], where the candidate RoIs identified on an input image are assumed to indirectly form a sequence, based on the scalar confidence score assigned to each of them by the detector (in descending order). Traditionally, evaluating a detector's accuracy on a known dataset involves an analysis of this sequence. At each step, a candidate RoI is processed and matched to a ground-truth object, if and only if: (a) their IoU is higher than a predefined threshold, and (b) that ground-truth object hasn't been previously matched to a higher-scoring candidate detection. In the case where both (a) and (b) are fulfilled, the candidate RoI is marked as "correct", otherwise it is marked as "false". In the special case where only (a) is fulfilled, the candidate detection is marked as "false", due to it being a "duplicate" detection. Thus, the position of a candidate RoI in the sequence can be a significant factor when taking the decision to classify it as a "duplicate" or not.

This emphasis in the ordering is shared with problems traditionally viewed as sequence-to-sequence ones. For instance, in machine translation, a sequence of words from one language must be transformed into a sequence of words in another language. The order of each word (*token*) in the sentence is crucial and can modify its meaning (*context*). Similarly, in object detection evaluation, although a candidate RoI (token) can be successfully matched to a ground-truth object, it can be classified as "duplicate" and therefore as "false", instead of being classified as "correct", due to the fact that a higher-scoring candidate detection, which has been positioned earlier in the sequence, has already been matched with the same ground-truth object.

Motivated by these notions, this paper explicitly formulates the NMS task as a mapping from an input sequence of candidate RoIs to a corresponding output sequence with identical length. Let $\mathbf{R}^{in}$ be the input sequence of candidate RoIs, in descending order with respect to detector confidence scores:

$$\mathbf{R}^{in} = [\mathbf{r}_1^{in}, ..., \mathbf{r}_N^{in} | r_i^{score_{det}} \geq r_{i+1}^{score_{det}}] \quad (1)$$

where $\mathbf{r}_i^{in} = [r_i^{x_{min}}, r_i^{y_{min}}, r_i^{x_{max}}, r_i^{y_{max}}, r_i^{score_{det}}]$ is an input candidate RoI expressed through its 2D image coordinates, along with its corresponding score assigned by the detector, and N is the number of candidate detections. Let $\mathbf{R}^{out}$ be the output sequence of candidate RoIs, in descending order based on the scores assigned by the NMS method:

$$\mathbf{R}^{out} = [\mathbf{r}_1^{out}, ..., \mathbf{r}_N^{out} | r_i^{score_{NMS}} \geq r_{i+1}^{score_{NMS}}] \quad (2)$$

where $\mathbf{r}_i^{out} = [r_i^{x_{min}}, r_i^{y_{min}}, r_i^{x_{max}}, r_i^{y_{max}}, r_i^{score_{NMS}}]$ is an NMS-rescored candidate RoI. The proposed formulation of the NMS task can be expressed as:

$$\mathbf{R}^{out} = NMS(\mathbf{R}^{in}) \quad (3)$$

Building upon this novel view of the NMS task, the method proposed in this paper, which we call *Seq2Seq-NMS*, receives as input a sequence of candidate RoIs, generated by an object detector, and extracts rich representations regarding their appearance and geometry. Subsequently, these representations are fed to a DNN which processes them in parallel, while mainly paying attention to spatially neighboring, higher-scoring candidates when analyzing each RoI. Finally, it outputs a sequence of scalar scores, each one defining the context of a candidate detection. This is essentially information that determines the final decision of whether the respective RoI should be classified as "correct" or as "potentially suppressed", after the NMS task has been completed. In the proposed formulation, the context of the i^{th} candidate detection is expressed through the corresponding output score, which is a classification probability $p_i : \{p_i \in \mathbb{R} | 0 \leq p_i \leq 1\}$ (1/0 means "correct"/"potentially suppressed", respectively). After the inference stage, simple thresholding can be applied on these output probabilities/scores, in order to decide which candidate detections should be retained. This formulation avoids hard discarding/pruning of RoIs at the inference phase itself, thus allowing us to find a balance in the trade-off between False Positive Rate (FPR) and True Negative Rate (TNR), depending on the application (e.g., using a low threshold in human safety-centric applications such as pedestrian detection).

Seq2Seq-NMS relies on building rich representations for each candidate detection, based on their visual appearance, their geometry and their interrelations. Abstractly, it consists of the following three steps:

- Appearance-based RoI representations extraction.
- Geometry-based RoI representations extraction.
- Detections rescoring through the attention-driven NMS DNN.

These steps are detailed below.

A. Appearance-based RoI Representations Extraction

This step can be considered optional, since RoI representations that have been already computed at the intermediate feature extraction layers of the DNN-based object detector itself can be used instead. However, the use of RoI representations computed solely for the NMS procedure makes the NMS DNN less detector-specific and more robust against variations in the effectiveness and the performance of the deployed detector. In [3], where the goal was person detection

from aerial views, representations consisting of statistical RoI appearance properties, computed on the spatial distribution of edges detected within each RoI, were used. These distributions acted as a discriminant factor for identifying complete vs partial object silhouettes, since the aerial view of persons silhouettes are similar in shape. However, the same argument can be made for people seen from a ground perspective (e.g., pedestrians perceived by an autonomous car), therefore this is a solution applicable to most person detection scenarios.

Algorithm 1: Appearance-based RoI representations extraction using FMoD

Input: (a) an RGB image $\mathbf{I}$
(b) a set of N RoIs expressed in 2D pixel coordinates $\mathbf{B} = [\mathbf{b}_0, \mathbf{b}_1, .., \mathbf{b}_N] \in \mathbb{R}^{N \times 4}$
(c) FMoD pyramid levels L, $L \geq 1$

Output: Appearance-based representations $\mathbf{A} \in \mathbb{R}^{N \times 5(4^L - 1)}$

begin
  Resize image $\mathbf{I}$ to a fixed size of $W_f \times H_f$.
  $E(\mathbf{I}) \leftarrow$ Compute the edge map of image $\mathbf{I}$.
  Extract in parallel the 0^{th}-level RoI maps $\mathbf{M}^0 = [\mathbf{M}_0^0, \mathbf{M}_1^0, .., \mathbf{M}_N^0]$, where $\mathbf{M}_i^0 \in \mathbb{R}^{1 \times W_0 \times H_0}$, through the ROIAlign operator on $E(\mathbf{I})$.
  Compute in parallel the 0^{th}-level FMoD representations $\mathbf{A}^0 = [\mathbf{A}_0^0, \mathbf{A}_1^0, .., \mathbf{A}_N^0]$ of $\mathbf{M}^0$, where $\mathbf{A}_i^0 \in \mathbb{R}^{15 \times 1}$.
  for $j \leftarrow 1$ *to* $(L-1)$ **do**
    Extract in parallel the j^{th}-level RoI maps $\mathbf{M}^j = [\mathbf{M}_0^j, \mathbf{M}_1^j, .., \mathbf{M}_N^j]$, where $\mathbf{M}_i^j \in \mathbb{R}^{4^j \times \frac{W_0}{2^j} \times \frac{H_0}{2^j}}$, through subdivision of $\mathbf{M}^0$ RoI maps into four quadrants for j times, using the ROIAlign operator.
    Compute in parallel the j^{th}-level FMoD representations $\mathbf{A}^j = [\mathbf{A}_0^j, \mathbf{A}_1^j, .., \mathbf{A}_N^j]$ of $\mathbf{M}^j$, where $\mathbf{A}_i^j \in \mathbb{R}^{15 \times 4^j}$.
  end
  Concatenate FMoD representations across all pyramid levels $\mathbf{A} \in \mathbb{R}^{N \times 5(4^L - 1)}$, where $\mathbf{A}_i = [\mathbf{A}_i^0, .., \mathbf{A}_i^L]$.
end

In [3], a CPU implementation of the low-level FMoD visual descriptor was employed for representing candidate RoIs. FMoD was originally devised in a global [7] and in a local [32] variant (LMoD), respectively applied to movie [33] and activity video [34] [35] [36] summarization via key-frame extraction. Typically, FMoD and LMoD capture informative image statistics from various available image channels (e.g., luminance, color/hue, optical flow magnitude, edge map, and/or stereoscopic disparity), both in a global and in various local scales, under a spatial pyramid video frame partitioning scheme. Following [3], only the edge map of an image's luminance channel is used here as input channel for the FMoD algorithm, with the latter one applied separately at each candidate RoI. The intent is to compactly capture the spatial distribution of the edges within each RoI in a single description vector. However, in [3] RoIs were processed sequentially and not simultaneously, thus demanding very long inference times. To tackle this limitation, in this paper FMoD was re-implemented neurally so that it runs very fast and in parallel on modern GPUs. Given as input an image and a set of candidate RoIs (in pixel coordinates) of different shape and scale, it extracts all corresponding regions of the luminance edge map by cropping it along the boundaries of the respective RoIs. This is done separately for each candidate RoI, but in parallel for all of them (at a single step). Subsequently, the FMoD descriptors/representations of all these cropped edge maps/RoIS are also computed separately but in parallel.

The appearance-based RoI representations extraction process can be divided into three operations. The first one involves the computation of the edge map of the input image, which is a relatively fast and efficient process. The second step is the use of the *ROIAlign* [37] operator to extract, in parallel, fixed-size regions across one or multiple maps. Finally, deriving the FMoD representations of these fixed-size maps involves in-parallel computation of the following 15 scalar statistical attributes:

- (1-3) horizontal/vertical/vectorized-block mean values.
- (4-6) horizontal/vertical/vectorized-block standard deviation values.
- (7-9) horizontal/vertical/vectorized-block skew values.
- (10-12) horizontal/vertical/vectorized-block kurtosis values.
- (13-15) horizontal/vertical/vectorized-block signal power values.

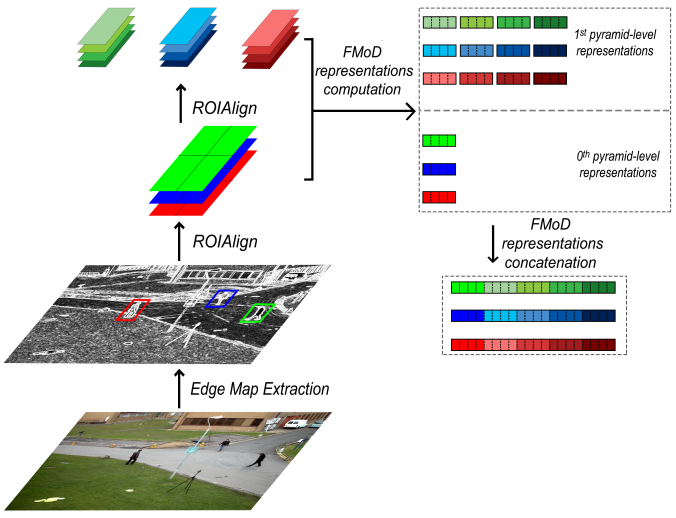


Fig. 2: Computation of the visual appearance-based candidate RoI representations, by applying the fast FMoD implementation to an image with 3 RoIs and using 2 pyramid levels.

The corresponding procedure is described in Algorithm 1. Initially, the RGB input image $\mathbf{I}$, of an arbitrary resolution, is resized to a fixed resolution of $W_f \times H_f$ and its corresponding

edge map $E(\mathbf{I})$ is computed. To make actual inference times even shorter, this operation is carried out here in parallel with the corresponding detector's inference phase. Similarly to [3], the FMoD representations of all RoIs are computed under a spatial pyramid partitioning scheme [38]. At the pyramid base, the 0^{th}-level RoI maps $\mathbf{M}^0 = [\mathbf{M}^0_0, \mathbf{M}^0_1, .., \mathbf{M}^0_N]$, $\mathbf{M}^0_i \in \mathbb{R}^{1\times W_0\times H_0}$ are extracted in parallel by applying the ROIAlign operator on $E(\mathbf{I})$, assuming that N candidate RoIs have been identified by the object detector for input $\mathbf{I}$. Using $\mathbf{M}^0$, the 0^{th}-level FMoD representations $\mathbf{A}^0 = [\mathbf{A}^0_0, \mathbf{A}^0_1, .., \mathbf{A}^0_N]$, $\mathbf{A}^0_i \in \mathbb{R}^{15\times 1}$ are computed in parallel. Subsequently, the representations at the remaining spatial pyramid levels are computed iteratively, by the in-parallel computation first of $\mathbf{M}^j$ and then of the corresponding partial FMoD descriptors $\mathbf{A}^j$. Once the latter ones have been computed for all (predefined and fixed) L pyramid levels, they are concatenated along them. For example, in an image with $N = 3$ candidate RoIs and $L = 2$ pyramid levels, $\mathbf{A} \in \mathbb{R}^{3\times 75}$. This example is illustrated in Figure 2.

B. Geometry-based RoI Representations Extraction

The spatial/geometric interrelations between the various candidate RoIs, based only on their 2D pixel coordinates and not on their visual appearance, is crucial for solving the NMS problem. Such a set of purely geometric attributes has previously proven effective as an input descriptor, in the context of GossipNet [2]. Thus, in this paper, a slightly similar, but enriched set of attributes has been devised, serving as an additional representation for each RoI.

Given a set of N candidate RoIs, along with their corresponding detection scores, the tensor $\mathbf{G} \in \mathbb{R}^{N\times N\times 14}$ is computed, where each entry $\mathbf{G}^{ij} \in \mathbb{R}^{14}$ contains the following attributes:

- (1-3) the normalized horizontal/vertical/euclidean distances[1] between the centers of the j^{th} and the i^{th} RoI.
- (4-7) the normalized width/height/area/aspect-ratio of the j^{th} RoI.
- (8-11) the ratios between the j^{th} and the i^{th} RoIs width/height/area/aspect-ratio (e.g., $\frac{w_j}{w_i}$).
- (12) the detector's confidence score for the j^{th} RoI.
- (13) the detector's confidence score differences between the j^{th} and the i^{th} RoI (e.g., $s_j - s_i$).
- (14) the IoU between the j^{th} and the i^{th} RoI.

Therefore, each diagonal entry $\mathbf{G}^{ii} \in \mathbb{R}^{14}$ contains the geometric representation of the i-th input candidate RoI/detection.

C. Detections rescoring through the attention-driven NMS DNN

The goal of the proposed DNN architecture is to perform one-class Non-Maximum Suppression on a set of candidate RoIs/detections through rescoring rather than pruning them. For a given set of N such RoIs, the DNN receives as input a sequence of corresponding representations ($\mathbf{A}$ and $\mathbf{G}$, encoding the appearance and geometry of all RoIs in the sequence), sorted in a descending order based on the respective scalar detection confidence score.

During inference, these two types of information are fused and each candidate RoI refines its representation by attending to the representations of all detections in the set. The Scaled Dot-Product Attention mechanism [31], originally proposed for machine translation tasks, is employed, since it has been proven effective in various applications, such as image classification [39], or generation [40]. The mechanism is briefly described below. In the context of the proposed DNN, the candidate detections used as keys are represented in a relative-to-each-query manner within this attention mechanism. Although this choice leads to slightly increased computational and memory costs, it allows the DNN to more effectively capture the interrelations between the candidate detections.

Finally, the model predicts a new scalar score for each RoI, indicating whether it should be suppressed or not. The output sequence is formed by sorting the candidate RoIs, based on their new scores in descending order.

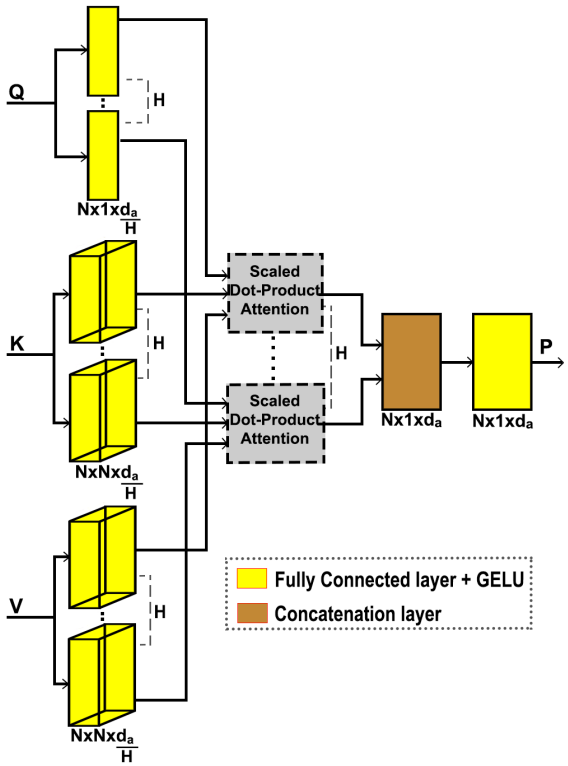


Fig. 3: Illustration of the Multihead Self-Attention Module.

Multihead Self-Attention Module: The Scaled Dot-Product Attention, also known as self-attention, was presented in [31] and formulated as follows:

$$Attention(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = softmax(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}})\mathbf{V}, \qquad (4)$$

where $\mathbf{Q} \in \mathbb{R}^{N_q\times d_k}$ are the queries, $\mathbf{K} \in \mathbb{R}^{N_k\times d_k}$ are the keys and $\mathbf{V} \in \mathbb{R}^{N_k\times d_v}$ are the values. Each query and each key has

[1]Horizontal and vertical distances are signed distances.

a dimension of d_k, while each value has a dimension of d_v. Multihead Attention was also proposed in [31], as a module which allows various attention mechanisms, including self-attention, to run in parallel. This module can be formulated as:

$$Multihead(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = [\mathbf{h}_1, ..., \mathbf{h}_H]\mathbf{W}^O, \tag{5}$$

where

$$\mathbf{h}_i = Attention(\mathbf{Q}\mathbf{W}_i^Q, \mathbf{K}\mathbf{W}_i^K, \mathbf{V}\mathbf{W}_i^V). \tag{6}$$

In this formulation, $\mathbf{W}_i^Q \in \mathbb{R}^{d_a \times d_k}$, $\mathbf{W}_i^K \in \mathbb{R}^{d_a \times d_k}$, $\mathbf{W}_i^V \in \mathbb{R}^{d_a \times d_v}$, $\mathbf{W}_i^O \in \mathbb{R}^{Hd_v \times d_a}$ are projection parameter matrices, H is the number of heads, $d_k = d_v = \frac{d_a}{H}$, and the operator $[...]$ implies concatenation.

The proposed DNN architecture relies on these mechanisms in order to identify relations between candidate detections, based both on their visual appearance and their geometric properties. Such relations can help the model in determining whether a detection should be suppressed or not. For example, the DNN can decide that a higher-scoring candidate RoI should possibly suppress other less-scoring ones having similar appearance and geometric representations.

In [31] the authors introduced *positional encoding* for Natual Language Processing (NLP) tasks, which uses a combination of sines and cosines at multiple frequencies, in order to encode the position of a word in a sequence. In theory, this approach could also be adopted for encoding RoI geometry (e.g., the position of RoI centers along a certain axis). However, this may fail to capture the interrelations of candidate RoIs in a relative manner, as the encoded information in the NMS task is far more complex compared to [31]. As an alternative, we approached the task by encoding all the representations of the input candidate detections in a relative-to-each-RoI manner. Thus, the keys and values of the Scale Dot-Product Attention are represented in a relative-to-each-query representation scheme. For example, the j^{th} key may be represented differently for the i^{th} query, compared to its representation for the $(i+1)^{th}$ query. Although this increases the method's memory complexity, each query is allowed to represent the keys and the values relatively to itself. Thus, for N detections, $\mathbf{Q} \in \mathbb{R}^{N \times 1 \times d_a}$, $\mathbf{K} \in \mathbb{R}^{N \times N \times d_a}$ and $\mathbf{V} \in \mathbb{R}^{N \times N \times d_a}$, the output is $\mathbf{P} \in \mathbb{R}^{N \times 1 \times d_a}$.

Due to the increased number of dimensions of $\mathbf{Q}$, $\mathbf{K}$ and $\mathbf{V}$, batch matrix multiplication is employed in Eq. (4) to speed up the process. The architecture of this module is illustrated in Figure 3.

Joint Processing Module (JPM): In this module, the representations of the detections are jointly and simultaneously refined, mainly through the Multihead Self-Attention mechanism. The JPM receives as its input $\mathbf{F}_t^Q \in \mathbb{R}^{N \times 1 \times d_m}$, which holds the current representations of all candidate detections, as well as $\mathbf{F}_t^K \in \mathbb{R}^{N \times N \times d_a}$, which holds the current relative-to-each-detection representations, for all N candidate detection.

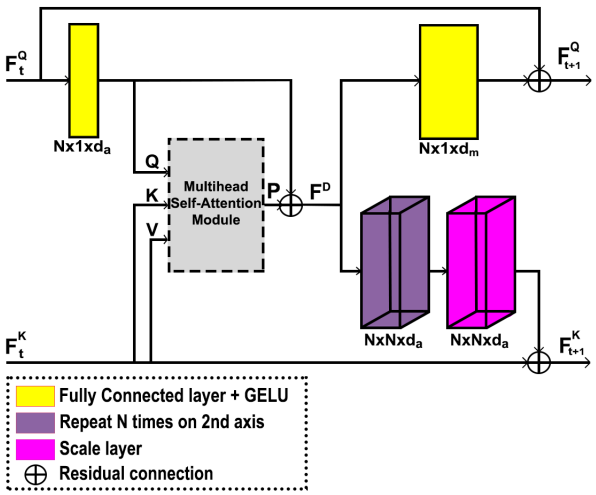


Fig. 4: Illustration of the Joint Processing Module (JPM).

The architecture of the JPM is shown in Fig. 4. The queries and keys are formed as:

$$\begin{aligned} \mathbf{Q} &= \mathbf{F}_t^Q \mathbf{C}^Q, \\ \mathbf{K} &= \mathbf{F}_t^K, \\ \mathbf{V} &= \mathbf{K}, \end{aligned} \tag{7}$$

where $\mathbf{C}^Q \in \mathbb{R}^{d_m \times d_a}$ stands for the weights of a fully connected layer. The new representations of the candidate detections, which is the output of this module, are formed as:

$$\begin{aligned} \mathbf{F}_{t+1}^Q &= \mathbf{F}^D \mathbf{C}^D + \mathbf{F}_t^Q, \\ \mathbf{F}^D &= \mathbf{P} + \mathbf{Q}, \end{aligned} \tag{8}$$

where $\mathbf{C}^D \in \mathbb{R}^{d_a \times d_m}$ also denotes the weights of a fully connected layer. In addition, residual connections [41] are applied between $\mathbf{Q}$ and $\mathbf{P}$ as well as between $\mathbf{F}_{t+1}^Q$ and $\mathbf{F}_t^Q$.

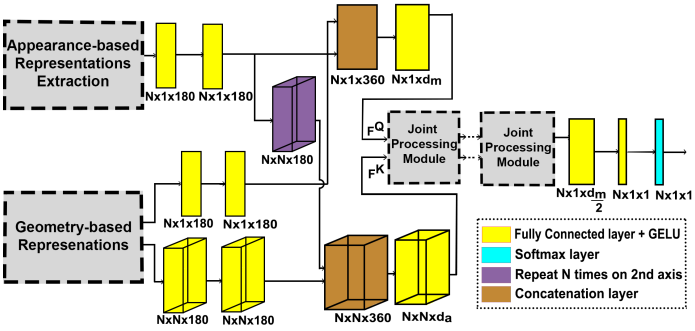


Fig. 5: Seq2Seq-NMS architecture. N is the number of input candidate RoIs/detections.

Finally, the relative-to-each-candidate-detection representations $\mathbf{F}^K$ are refined as:

$$\mathbf{F}_{t+1}^K = \mathbf{F}_t^K + \mathbf{F}^S \otimes \mathbf{C}^K, \tag{9}$$

where $\mathbf{F}^S$ is derived from $\mathbf{F}^D$, by repeating it N times along its second dimension, and $\mathbf{C}^K$ are learned weights of

a *Scale Layer* that we introduce, performing an element-wise multiplication between its weights and an input representation. Its purpose is to select the degree of information which will flow from $\mathbf{F}^S$ to $\mathbf{F}^K_{t+1}$ in each JPM.

Masking: A masking approach has been integrated into the self-attention mechanism of the proposed architecture. For N sorted candidate detections, we mask the values of the input of the softmax function in Eq. (4). Without loss of generality, masking is detailed below for the simplest case, where $H = 1$.

Given a candidate RoI $\mathbf{r}^{in}_i$, an its associate RoI $\mathbf{r}^{in}_j$ and $\mathbf{S} = \frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}}$, masking is defined as:

$$S_{ij} = \begin{cases} -\infty, & \text{if } IoU(\mathbf{r}^{in}_i, \mathbf{r}^{in}_j) < 0.2 \\ 0.1 \cdot S_{ij}, & \text{if } IoU(\mathbf{r}^{in}_i, \mathbf{r}^{in}_j) \geq 0.2 \text{ and } j > i \\ S_{ij}, & \text{otherwise} \end{cases} \quad (10)$$

Masking is employed for two reasons. First, each RoI must be prevented from attending to spatially distant detections. The overlap of RoIs is used to determine whether S_{ij} should be set to $-\infty$, before applying the softmax function. If yes, the attention weight linking $\mathbf{r}^{in}_i$ to $\mathbf{r}^{in}_j$ (after applying softmax) will be zeroed out. Second, we attempt to replicate the behaviour of Greedy NMS, where a detection is characterized as duplicate, thus marked for suppression, when another, higher-scoring detection spatially covers the same object. In the proposed architecture this can be accomplished by forcing (through masking) the internal representation of a candidate detection to be modified by attending mainly to representations that correspond to RoIs higher-scoring than itself.

Network Architecture: For a set of N candidate sorted detections, the proposed DNN uses as input their corresponding appearance-based $\mathbf{A}$ and geometry-based representations $\mathbf{G}$. FMoD representations of 3 pyramid levels are employed as $\mathbf{A} \in \mathbb{R}^{N\times 1\times 315}$. The extracted geometry-based RoI representations, namely $\mathbf{G} \in \mathbb{R}^{N\times N\times 14}$, are assigned to $\mathbf{G}^K$ as it contains the relative-to-each-candidate-detection representations. Its diagonal, derived from the first two dimensions, forms $\mathbf{G}^Q \in \mathbb{R}^{N\times 1\times 14}$. The representations derived from a fusion between $\mathbf{A}$ and $\mathbf{G}^Q$ form $\mathbf{F}^Q \in \mathbb{R}^{N\times 1\times d_m}$. This fusion is mainly accomplished by concatenating and applying fully-connected layers between the two types of representations. In addition, the representations derived from a fusion between $\mathbf{A}$ and $\mathbf{G}^K$ form $\mathbf{F}^K \in \mathbb{R}^{N\times N\times d_a}$. Both $\mathbf{F}^Q$ and $\mathbf{F}^K$ are used as input to the first JPM.

A stack of JPMs, sequentially connected, are in charge of refining representations $\mathbf{F}_Q$ and $\mathbf{F}_K$. Finally, after applying two fully connected layers on $\mathbf{F}^Q$, the DNN uses a softmax function to output the final NMS scores. The model architecture is depicted in Fig. 5. The Gaussian Error Linear Unit (GELU) is used as activation function. Layer normalization [42] is applied on the output of residual connections and dropout [43] is used for regularization, similarly to [31].

Training: The weighted binary cross entropy was selected as the training objective of the proposed neural architecture. In particular, the loss function is defined as:

$$L = -\sum_{i=1}^{N}(w_1 y_i \log(r_i^{scores_{NMS}}) + w_0(1-y_i)\log(1-r_i^{scores_{NMS}})), \quad (11)$$

where N is the number of candidate detections, $\mathbf{r}^{scores_{NMS}}$ are the output NMS scores, $\mathbf{w}$ are class weights and $\mathbf{y}$ are the labels derived from a matching function, given a specific IoU value. In particular, $y_i \in \{1, 0\}$ indicates whether the i^{th} detection was successfully matched to an object or not. A detection is matched successfully to an object, when the IoU between its RoI and an object's 2D bounding box is higher or equal to a matching threshold, and that specific object hasn't been matched to any higher scoring detection. In this paper, this matching IoU threshold was set to 0.5. A strategy similar to the one in [2], is used for the class weights computation.

IV. Experimental Evaluation

The performance of Seq2Seq-NMS was evaluated on three separate datasets for the person detection task. In all datasets, candidate RoIs from the *Single Shot Detector* (SSD) [19] were provided as input to the proposed NMS method. In the implemented version of the detector, VGG16 with atrous convolutions was selected as the backbone CNN. The input images were resized to a resolution of 512×512 pixels, while the detector was trained from scratch for each dataset[2]. Independently from this set of experiments, a complementary evaluation scheme was also conducted by employing a different detector per dataset. These detectors were selected in order: a) to facilitate direct comparisons with previously published NMS methods, and b) to compare the proposed NMS approach against competing ones in conjunction with different detectors with different behaviour. In this complementary set of experiments, the following detectors were employed: (a) a non-neural detector [17], (b) a two-stage DNN-based detector [18] and (c) a one-stage DNN-based detector [21].

The employed Seq2Seq-NMS architecture consists of 4 Joint Processing Modules. We set $d_m = 256$, and $d_a = \frac{d_m}{2} = 128$. The Multihead Self-Attention module uses $H = 2$ attention heads and thus $d_q = d_k = d_v = \frac{128}{H} = 64$. Appearance-based RoI representations computed from 3-level FMoD were used, with 0^{th} level RoI maps extracted at resolution $W_0 \times H_0 = 160 \times 160$ pixels. In each evaluation setup, the proposed method was trained using the ADAM [44] optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.99$ and $\epsilon = 10^{-9}$. Given that the number of RoIs may be extremely large, we first applied TorchVision NMS with the relaxed 0.8 IoU threshold as a preprocessing step (common in NMS literature [2]). To achieve a fair comparison, this preprocessing step is applied in all deployed methods. Finally, Seq2Seq-NMS is trained using only the 720 highest-scoring candidate detections as an input sequence, due to practical memory limitations.

In all cases, Seq2Seq-NMS was compared against both neural and non-neural NMS algorithms. The first competing method is a baseline Greedy NMS approach running on GPU.

[2]The employed SSD implementation was adopted from https://github.com/opendr-eu/opendr/tree/master/src/opendr/perception/object_detection_2d/ssd

TABLE I: COMPARISON OF DIFFERENT NMS METHODS ON THE PETS DATASET, USING DETECTIONS FROM [17]. THE BOTTOM LINE REPORTS THE GAINS ACHIEVED WITH THE PROPOSED METHOD.

Method	**Device**	**Pre-NMS max dets. = 600**			**Pre-NMS max dets. = 1200**			**Pre-NMS max dets. = 1500**		
		$AP_{0.5}$	**$AP_{0.5}^{0.95}$**	**Average Inference Time (ms)**	**$AP_{0.5}$**	**$AP_{0.5}^{0.95}$**	**Average Inference Time (ms)**	**$AP_{0.5}$**	**$AP_{0.5}^{0.95}$**	**Average Inference Time (ms)**
Original NMS IoU>0.4	GPU	76.7%	32.2%	2.1	77.2%	32.1%	3.5	77.3%	32.0%	5.0
Original NMS IoU>0.5	GPU	74.2%	31.7%	2.8	74.7%	31.7%	6.4	74.8%	31.7%	8.1
Original NMS IoU>0.6	GPU	66.9%	29.6%	4.2	67.2%	29.7%	10.1	67.3%	29.7%	13.6
TorchVision NMS IoU>0.4	GPU	76.8%	32.2%	0.4	77.3%	32.1%	0.6	77.3%	32.1%	0.6
TorchVision NMS IoU>0.5	GPU	73.9%	31.7%	0.4	74.4%	31.6%	0.6	74.4%	31.6%	0.5
TorchVision NMS IoU>0.6	GPU	66.4%	29.5%	0.4	66.6%	29.6%	0.5	66.7%	29.6%	0.6
Soft-NMS_L	CPU	77.6%	32.5%	50.3	77.6%	32.3%	98.5	77.6%	32.1%	143.3
Soft-NMS_G	CPU	78.2%	33.4%	39.2	77.6%	32.9%	89.5	77.2%	32.6%	154.7
Fast-NMS	GPU	75.3%	31.9%	1.4	75.2%	31.6%	2.2	75.2%	31.5%	3.2
Cluster-NMS	GPU	76.8%	32.2%	3.2	77.2%	32.1%	5.1	77.3%	32.1%	7.5
Cluster-NMS_S	GPU	75.7%	32.3%	2.7	74.0%	31.3%	4.2	74.7%	31.6%	6.6
Cluster-NMS_D	GPU	77.0%	32.3%	3.8	77.6%	32.1%	7.3	77.6%	32.1%	9.1
Cluster-NMS_{S+D}	GPU	77.2%	32.6%	4.0	76.5%	32.0%	8.0	76.5%	32.0%	11.2
Cluster-NMS_{S+D+W}	GPU	77.2%	32.6%	47.6	76.5%	32.0%	154.8	76.5%	32.0%	276.1
GossipNet	GPU	81.9%	36.3%	27.2	84.3%	37.2%	64.2	84.6%	37.2%	95.8
Seq2Seq-NMS	GPU	83.6%	37.8%	11.0	85.4%	38.4%	13.8	**85.5%**	**38.4%**	15.4
Seq2Seq-NMS Gains (The best performance of each method is used for comparison)		**$AP_{0.5}$**			**$AP_{0.5}^{0.95}$**					
		+0.9%			+1.2%					

The second is TorchVision's[3] GreedyNMS implemented to run very fast on GPUs. Soft-NMS [23], i.e., a non-neural NMS method widely used as a more accurate replacement for Greedy NMS, was also tested. Evaluation was conducted using both the linear and the Gaussian weighting functions (referred to as Soft-NMS_L and Soft-NMS_G, respectively), with on-CPU execution. Another competing algorithm is Fast-NMS [26]: a generally faster, non-neural replacement for standard NMS, executed on GPU but suffering a marginal penalty regarding accuracy. Additionally, several variants of Cluster-NMS [28], a more recent non-neural method, were also used for comparisons. Below, the term Cluster-NMS_S is used to imply the use of the score penalty mechanism, while Cluster-NMS_D implies the addition of the normalized central point distance. In the latter case, the method is equivalent to DIoU-NMS [27]. The term Cluster-NMS_{S+D} is used when both of these mechanisms are utilized. Finally, Cluster-NMS_{S+D+W} indicates a weighted strategy similar to [24]. More details regarding these variations can be found in [28]. The last approach selected for comparison purposes is GossipNet [2], a neural NMS method achieving state-of-the-art accuracy.

The hyperparameters of all non-neural methods were tuned so as to report the best achieved results on 0.5 IoU matching threshold. Evaluation was performed on a PC using an Intel Core i7-7700 CPU and an NVIDIA GeForce RTX 2080 GPU with 11GB of memory, both for training and inference. The employed evaluation metrics are $AP_{0.5}$, $AP_{0.5}^{0.95}$ and inference times. $AP_{0.5}$ corresponds to the average precision for 0.5 IoU, while $AP_{0.5}^{0.95}$ to the mean average precision for IoU ranging from 0.5 to 0.95 with a step size of 0.05.

In the evaluation of all methods, the number of maximum candidate detections prior to the NMS procedure was set to 1500. All RoIs outputted by the NMS algorithms were utilized for evaluation, without any thresholding.

[3]https://pytorch.org/vision/stable/ops.html#torchvision.ops.nms

A. PETS

PETS [45] is a relatively small dataset, whose images were collected from static surveillance cameras and provide diverse levels of occlusion. The average number of people depicted in an image is approximately 14. Apart from [19], [17], a non-neural person detection method designed to handle occlusions, was selected as the corresponding detector for providing raw candidate RoIs as input to the NMS methods.

The proposed NMS architecture was trained for 8 epochs. The learning rate was set to 10^{-4}/10^{-5}/10^{-6} for epochs 1-4/5-7/8, respectively. GossipNet's architecture and training followed [2]. Final parameters of all methods were selected according to the best achieved accuracy in the validation set.

Table I reports the results of the proposed and the competing NMS methods, using candidate detections from [17] as input. This object detector outputs a large number of candidate RoIs, thus leading to increased GPU memory consumption for both the proposed method and GossipNet. Typically, most candidate detections that can be successfully matched to ground-truth objects are assigned higher confidence scores by the detector, compared to RoIs with lower scores (e.g., < 0.05) which are mostly false positive samples. Thus, in this experiment, we attempt to evaluate whether the lowest scoring detections have an impact on the performance of the proposed and the competing NMS methods. Table I reports the results of each NMS approach using N candidate detections as input, for different values of N. As it can be seen, the performance of several non-neural methods, such as Soft-NMS_L and Cluster-NMS_D, does not improve when the lowest-scoring detections

(e.g., $>$ 1200) are used. In contrast, both neural methods achieve more accurate results for longer input sequences (more candidate RoIs per image). In this setup, the proposed method achieved both the best $AP_{0.5}$ and the best $AP^{0.95}_{0.5}$, against all competing approaches, even in the case where only the highest 1200 candidate input detections were used. The obtained $AP_{0.5}$ was 85.5%, which is a $+7.3\%$ improvement against Soft-NMS_L and Cluster-NMS_D, the non-neural method with the best $AP_{0.5}$, and a $+0.9\%$ improvement against GossipNet. In addition, the obtained $AP^{0.95}_{0.5}$ was 38.4%, which is an $+1.2\%$ gain over the competing methods. Notably, when using only a small number of the highest-scoring candidate detections (e.g, N = 600), the proposed method still achieves better results compared to all non-neural NMS algorithms. Regarding inference runtimes, it needs 15.4 ms to run per image when $N = 1500$, since the required edge maps are computed in parallel with the object detector's inference. Thus, it is faster than GossipNet, as well as far less affected (with respect to runtime) by the number of candidate detections used as input. Indeed the GossipNet inference runtime drastically increases with N but this is not the case for the proposed approach. However, Seq2Seq-NMS is slower than most non-neural methods running on GPU.

TABLE II: COMPARISON OF DIFFERENT NMS METHODS ON THE TEST SET OF THE PETS DATASET, USING DETECTIONS FROM [19]. THE BOTTOM LINE REPORTS THE GAINS ACHIEVED WITH THE PROPOSED METHOD.

Method	Device	$AP_{0.5}$	$AP^{0.95}_{0.5}$	Average Inference Time (ms)
Original NMS IoU>0.4	GPU	87.6%	35.0%	12.7
Original NMS IoU>0.5	GPU	89.9%	36.3%	13.1
Original NMS IoU>0.6	GPU	89.8%	37.1%	13.4
TorchVision NMS IoU>0.4	GPU	88.0%	35.1%	0.3
TorchVision NMS IoU>0.5	GPU	90.0%	36.4%	0.2
TorchVision NMS IoU>0.6	GPU	89.8%	37.2%	0.3
Soft-NMS_L	CPU	90.0%	38.2%	134.4
Soft-NMS_G	CPU	89.6%	38.6%	108.1
Fast-NMS	GPU	87.6%	36.8%	6.0
Cluster-NMS	GPU	90.2%	36.9%	13.4
Cluster-NMS_S	GPU	90.1%	38.0%	13.8
Cluster-NMS_D	GPU	90.2%	36.6%	17.9
Cluster-NMS_{S+D}	GPU	90.6%	38.3%	22.4
Cluster-NMS_{S+D+W}	GPU	90.6%	38.3%	38.2
GossipNet	GPU	90.7%	**38.8%**	24.5
Seq2Seq-NMS	GPU	**90.9%**	38.6%	19.7
Seq2Seq-NMS Gains		+0.2%	-0.2%	-

Table II reports the results using cadidate detections from [19]. The proposed method achieved an $AP_{0.5}$ of 90.9%, thus attaining a gain of $+0.2\%$ over GossipNet. In terms of $AP^{0.95}_{0.5}$, the proposed method was outperformed only by GossipNet (-0.2%) and was on par with Soft-NMS_G. Regarding inference runtimes, Seq2Seq-NMS needed on average 19.7 ms to run per image, since the required edge maps are computed in parallel with the object detector's inference stage. Though this is faster than GossipNet, it is again slower than non-neural methods running on GPU.

B. COCO Person

COCO 2014 is a large dataset consisting of 82,783 images for training and 40,504 images for validation/testing. Although it contains 80 labeled classes, only the "person" class was used for evaluating the proposed method. Its images depict people in various viewing angles, scales and poses. The average ground-truth number of persons depicted in an image is 2.17. When considering only the images that actually contain visible people, this number increases to 4.01. Candidate detections were extracted from SSD and Faster R-CNN [18], in separate experiments, while the validation set splits were adopted from [2]. The first data subset, referred to as "minival", contains 5000 images, while the second subset, referred to as "minitest", contains 35000 images.

The proposed method was trained for 12 epochs. The learning rate was set to $10^{-4}/10^{-5}/10^{-6}$ for epochs 1-8/9-11/12, respectively. GossipNet's architecture and training again followed [2]. The final hyperparameters of all methods were selected according to the best achieved accuracy in the minival (validation) set. Table III reports the results of all competing NMS approaches.

TABLE III: COMPARISON OF DIFFERENT NMS METHODS ON THE MINITEST SET OF THE COCO DATASET, USING DETECTIONS FROM [18] AND [19]. THE BOTTOM LINE REPORTS THE GAINS ACHIEVED WITH THE PROPOSED METHOD.

Method	Device	Input dets. from [18]		Input dets. from [19]		Average Inference Time (ms)
		$AP_{0.5}$	$AP^{0.95}_{0.5}$	$AP_{0.5}$	$AP^{0.95}_{0.5}$	
Original NMS IoU>0.4	GPU	65.4%	35.6%	56.3%	31.6%	4.3
Original NMS IoU>0.5	GPU	65.3%	35.8%	56.1%	31.6%	5.4
Original NMS IoU>0.6	GPU	63.3%	35.6%	55.5%	31.7%	6.9
TorchVision NMS IoU>0.4	GPU	65.4%	35.5%	56.3%	31.6%	0.3
TorchVision NMS IoU>0.5	GPU	65.3%	35.8%	56.1%	31.7%	0.3
TorchVision NMS IoU>0.6	GPU	63.1%	35.5%	55.5%	31.7%	0.4
Soft-NMS_L	CPU	66.6%	37.0%	57.0%	32.1%	11.6
Soft-NMS_G	CPU	66.3%	36.7%	57.2%	32.5%	11.7
Fast-NMS	GPU	64.3%	35.4%	55.8%	31.5%	1.6
Cluster-NMS	GPU	65.4%	35.5%	56.3%	31.6%	3.1
Cluster-NMS_S	GPU	65.3%	36.1%	57.1%	31.9%	3.7
Cluster-NMS_D	GPU	65.5%	35.6%	56.3%	31.6%	5.1
Cluster-NMS_{S+D}	GPU	65.9%	36.6%	57.3%	32.1%	5.3
Cluster-NMS_{S+D+W}	GPU	66.0%	**37.7%**	57.3%	32.1%	7.3
GossipNet	GPU	66.9%	36.1%	67.7%	36.7%	5.1
Seq2Seq-NMS	GPU	**67.4%**	37.0%	**68.7%**	**37.8%**	7.2
Seq2Seq-NMS Gains		+0.5%	-0.7%	+1.0%	+1.1%	-

When candidate detections from [18] were used as input, the proposed method achieves the best $AP_{0.5}$, equal to 67.4%, which is an improvement of $+0.8\%$ against Soft-NMS_L and $+0.5\%$ against GossipNet. In terms of $AP^{0.95}_{0.5}$, Seq2Seq-NMS is outperformed by Cluster-NMS_{S+D+W} and is on par with Soft-NMS_L, achieving a value of 37.0%.

Using candidate detections from [19], the proposed Seq2Seq-NMS architecture achieved more significant gains: an $AP_{0.5}$ of 68.7% and an $AP^{0.95}_{0.5}$ of 37.8%, thus reaching

gains of $+1.0\%$ and $+1.1\%$ respectively over the second best approach.

Regarding inference time, Seq2Seq-NMS is close to that of Cluster-NMS$_{\text{S+D+W}}$, but somewhat slower than GossipNet. The reported values are obtained by averaging the inference times of each method over the two separate cases (different employed detectors). Notably, the joint processing of input candidate RoIs by the neural NMS methods, compared to the non-neural ones, accomplishes more significant improvements when given inputs from the one-stage detector [19] than those from the two-stage detector [18]. In a sense, the neural NMS approaches seem to compensate for the inferior accuracy of one-stage detectors compared to the two-stage ones.

C. CrowdHuman

The CrowdHuman dataset has been recently released to specifically target human detection in crowded areas. Crowded scenes are particularly challenging for person detectors, due to heavy visual occlusion of individual humans. The dataset contains 15000 images for training, 4370 images for validation and 5000 images for testing. The average number of persons in an image is 22.64, with various types of occlusions. Candidate detections were extracted from SSD [19] and YOLOv4 [21]. The images fed to the latter were rescaled to a resolution of 608×608 pixels.

TABLE IV: COMPARISON OF DIFFERENT NMS METHODS ON THE CROWDHUMAN DATASET, USING DETECTIONS FROM [21] AND [19]. THE BOTTOM LINE REPORTS THE GAINS ACHIEVED WITH THE PROPOSED METHOD.

Method	Device	Input dets. from [21]		Input dets. from [19]		Average Inference Time (ms)
		$AP_{0.5}$	$AP_{0.5}^{0.95}$	$AP_{0.5}$	$AP_{0.5}^{0.95}$	
Original NMS IoU>0.4	GPU	78.8%	45.6%	62.6%	29.9%	8.3
Original NMS IoU>0.5	GPU	83.3%	48.2%	66.3%	31.5%	8.6
Original NMS IoU>0.6	GPU	85.3%	49.8%	67.0%	32.4%	9.8
TorchVision NMS IoU>0.4	GPU	79.1%	45.7%	62.8%	30.0%	0.3
TorchVision NMS IoU>0.5	GPU	83.5%	48.3%	66.4%	31.6%	0.3
TorchVision NMS IoU>0.6	GPU	85.3%	49.9%	66.9%	32.4%	0.4
Soft-NMS$_{\text{L}}$	CPU	85.8%	51.1%	66.5%	32.3%	54.2
Soft-NMS$_{\text{G}}$	CPU	84.9%	50.4%	67.1%	33.0%	58.1
Fast-NMS	GPU	84.3%	49.7%	64.8%	31.4%	2.2
Cluster-NMS	GPU	85.3%	49.9%	67.1%	32.1%	5.0
Cluster-NMS$_{\text{S}}$	GPU	83.6%	49.2%	64.0%	31.0%	5.2
Cluster-NMS$_{\text{D}}$	GPU	85.5%	50.4%	67.1%	32.2%	6.5
Cluster-NMS$_{\text{S+D}}$	GPU	84.7%	50.1%	65.7%	31.8%	8.0
Cluster-NMS$_{\text{S+D+W}}$	GPU	84.7%	50.1%	65.7%	31.9%	32.3
GossipNet	GPU	87.2%	51.0%	72.4%	35.0%	10.0
Seq2Seq-NMS	GPU	**87.3%**	**51.2%**	**73.9%**	**35.9%**	9.4
Seq2Seq-NMS Gains		+0.1%	+0.1%	+1.5%	+0.9%	-

The proposed NMS method was trained for 14 epochs. The learning rate was set to $10^{-4}/10^{-5}/10^{-6}$ for epochs 1-8/9-12/13-14, respectively. GossipNet was trained for 10^6 iterations, with a learning rate set to 10^{-4} and decreased by 0.1 at the 6×10^5-th and the 8×10^5-th iterations.

Table IV shows that the proposed method achieves minimal gains, in terms of $AP_{0.5}$ and $AP_{0.5}^{0.95}$, when input candidate detections are provided by [21]. Indeed, Seq2Seq-NMS achieves an $AP_{0.5}$ of 87.3%, which is a $+1.5\%$ improvement against Soft-NMS$_{\text{L}}$ but corresponds to a minor $+0.1\%$ improvement over GossipNet. Similarly, the proposed method achieved $AP_{0.5}^{0.95} = 51.2\%$ which corresponds to only a minor $+0.1\%$ improvement against the best competitor. However, when candidate detections are provided by [19] the proposed method achieves an $AP_{0.5}$ of 73.9% and $AP_{0.5}^{0.95} = 35.9\%$. The gains in both metrics are quite significant compared to the second-best GossipNet, achieving improvements of $+1.5\%$ and of $+0.9\%$ respectively.

Regarding inference runtime, the proposed method requires on average 9.4 ms; thus, it is faster than all non-GPU approaches and slightly faster than GossipNet. The reported values are obtained by averaging the inference times of each method over the two separate cases (different employed detectors).

D. FMoD Ablation Study

This Subsection examines the effect of the appearance-based features extracted by FMoD on the performance of Seq2Seq-NMS. Moreover, alternative appearance-based descriptors which could replace FMoD in the overall pipeline are investigated. Experiments were performed on the CrowdHuman dataset, using [19] for providing the input raw candidate detections.

TABLE V: PERFORMANCE EVALUATION OF THE PROPOSED METHOD USING APPEARANCE-BASED ROI REPRESENTATIONS OBTAINED BY DIFFERENT FMOD VARIANTS.

Resolution of RoIs (in pixels)	Num. of Pyramid Layers	$AP_{0.5}$	$AP_{0.5}^{0.95}$	Average Inference Time (ms)
20×20	1	73.2%	35.5%	7.4
160×160	1	73.3%	35.5%	7.9
20×20	2	73.3%	35.5%	8.3
160×160	2	73.4%	35.7%	8.5
20×20	3	73.7%	35.8%	8.9
160×160	3	**73.9%**	**35.9%**	9.0

The following aspects of FMoD were examined:

- the scale of RoIs used for computing the FMoD descriptors. To do so, the edge map RoIs obtained by the ROIAlign operator were extracted in a fixed resolution of either: a) 20×20 pixels, or b) 160×160 pixels, before computing the respective FMoD descriptors on them.
- the optimal number of FMoD spatial pyramid levels. Experiments were carried out for pyramid levels L equal to 1, 2 and 3.

As shown in Table V similar performance is attained for 1 or 2 FMoD pyramid levels, but the accuracy of Seq2Seq-NMS is improved with 3 FMoD pyramid levels. The scale of RoIs extracted by the RoIAlign operator seems to have a minimal impact on the accuracy. The reported inference

times amount to the overall time needed for computing the corresponding edge maps and extracting their appearance-based RoI representations using FMoD.

Moreover, candidate detections from [19] in the CrowdHuman dataset were also utilized in order to compare the following three variants of Seq2Seq-NMS:

- Seq2Seq-NMS that utilizes only geometry-based RoI representations. To achieve this, the DNN was fed with dummy zero-vectors as appearance-based representations.
- An extension of Seq2Seq-NMS where learnt convolutional features are employed as appearance-based RoI representations, instead of FMoD descriptors: in practice, already computed feature maps from the corresponding detector's backbone CNN are exploited. Two variants were examined by employing the feature maps from the initial layers of VGG16 during inference. Early layers were preferred in order to retain as much spatial information as possible. The size of the selected maps, defined as tensors, were $64 \times 64 \times 512$, with the last dimension being the depth of the corresponding convolutional layer. In the first variant, the maps were properly resized and RoI maps were extracted using the ROIAlign operator in a 20×20 resolution. In the second variant, a convolutional layer, with window=1×1, stride=1×1, and 32 filters followed by ReLU as activation function was employed before the ROIAlign operator. In this variant, the memory requirements induced by the ROIAlign operator were heavily reduced compared to the first variant. It must be highlighted that the ROIAlign operator is fully differentiable. A simple deep neural module, depicted in Table VI was implemented in order to compute the final appearance-based RoI representations. Seq2Seq-NMS was trained jointly with this module.
- The default Seq2Seq-NMS which uses FMoD descriptors as appearance-based RoI representations.

TABLE VI: IMPLEMENTED DEEP NEURAL MODULE IN SEQ2SEQ-NMS, TASKED WITH EXTRACTING APPEARANCE-BASED ROI DESCRIPTIONS.

Conv2D + ReLU, window=3×3, stride=1×1, filters=20
Conv2D + ReLU, window=3×3, stride=1×1, filters=4
Max-Pooling, window=2×2, stride=2×2
Flatten
Fully-Connected Layer + ReLU

The relevant evaluation results are reported in Table VII. Default Seq2Seq-NMS with FMoD descriptors as appearance-based RoI representations improves $AP_{0.5}$ by $+0.8\%$ and $AP_{0.5}^{0.95}$ by $+0.3\%$, compared to geometry-only RoI representations. A more notable improvement is demonstrated with convolutional RoI representations derived by the deep neural module: in the base case, this variant improved $AP_{0.5}$ by $+2.2\%$ and $AP_{0.5}^{0.95}$ by $+1.3\%$, compared to the geometry-only Seq2Seq-NMS. The more memory-efficient variant achieved $+1.0$ and $+0.5\%$ in the respective metrics. Regarding inference times, FMoD requires 2.7 ms in order to extract the corresponding appearance-based RoI representations from edge maps. If one includes the edge map computation, the corresponding inference time rises to 9.0 ms since, in our implementation, edge maps were computed in CPU; GPU alternatives may be much less time-demanding, thus significantly reducing overall inference requirements. In addition, the first variant of deep neural appearance-based RoI representations extraction requires 0.8 ms, while the more time- and memory-efficient variant requires 0.5 ms. The time needed by VGG16, in order to compute the raw feature maps is not included in the reported times.

TABLE VII: PERFORMANCE OF SEQ2SEQ-NMS ON THE CROWDHUMAN DATASET, USING DIFFERENT APPROACHES TO APPEARANCE-BASED ROI REPRESENTATION.

Type of the Appearance-based RoI Representations	**$AP_{0.5}$**	**$AP_{0.5}^{0.95}$**	**Average Inference Time (ms)**
Geometry-based RoI representations only (Using zero vectors as dummy appearance representations)	73.1%	35.6%	0.0
Deep neural RoI representations extracted from raw VGG16 feature maps at size $64 \times 64 \times 512$	**75.3%**	**36.9%**	0.8
Deep neural RoI representations extracted from VGG16 feature maps at size $64 \times 64 \times 32$	74.1%	36.1%	0.5
FMoD RoI representations	73.9%	35.9%	2.7 (9.0)

E. Discussion

Overall, the proposed Seq2Seq-NMS DNN achieves top accuracy on the $AP_{0.5}$ metric in all three datasets. The results show that Seq2Seq-NMS can successfully capture interrelations between candidate detections for the person detection task, based both on their visual appearance and their geometry. The three datasets used for evaluation contain images with a great variety of visible persons density, ranging from images of individual people to photographs of large crowds, indicating that Seq2Seq is suitable for generic person detection.

Regarding the $AP_{0.5}^{0.95}$ metric, Seq2Seq-NMS achieves top accuracy in most cases. The main exception is COCO dataset, when using candidate detections from [19]. This behaviour can be explained by the fact that our method was specifically enforced during training to match candidate RoIs to ground-truth RoIs, in case their in-between IoU is more than 0.5, instead of doing so for various IoU thresholds in the $[0.5, 0.95]$ range. More details about the matching strategy procedure, adopted in training, can be found in Section III.

Moving on to inference running time, the proposed method is relatively slower than non-neural, mostly less accurate, GPU-executed algorithms. However, when compared against DNN architectures for NMS, such as GossipNet, Seq2Seq-NMS achieves faster inference, with the exception of COCO (Table III). In addition, the inference runtime of Seq2Seq-NMS seems less affected by the input sequence length (number of candidate detections N), thus achieving faster inference when processing longer sequences, as shown in, e.g., Table I.

Another observation stemming from the presented experimental results is that Seq2Seq-NMS fits well with person detectors of various types: it achieves improved $AP_{0.5}$ performance against several competing NMS methods when combined with detectors of any nature (non-neural, one- and two-stage DNN-based). In the default Seq2Seq-NMS architecture, the use of FMoD for describing the visual appearance of the cropped candidate RoIs reinforces such a behaviour, since FMoD descriptors are independent of the employed person detector.

In addition, as shown in the ablation study presented in Section IV-D, the use of appearance-based RoI representations from FMoD indeed improves the performance of Seq2Seq-NMS, compared to the case where only geometry-based representations are used. The same study showed that the best accuracy is achieved when the appearance-based features are computed using three FMoD pyramid levels, whereas the scale of RoIs has minimal impact on accuracy. Finally, a simple variant of Seq2Seq-NMS that exploits deep neural appearance-based RoI representations from internal feature maps of the employed detector, instead of FMoD descriptors, further improves accuracy as shown in Table VI.

Besides the results depicted in Tables I, II III and IV, an ablation study was also performed regarding the proposed masking operation (described in Section III-C) of the self-attention mechanism. Omitting masking led to reduced accuracy rates, or even training convergence failures in cases with huge numbers of candidate RoIs per image. The importance of masking stems from the fact that it enforces an ordering constraint on how the internal representation of each candidate detection is shaped: thanks to masking, its form is finalized by attending mainly to representations that correspond to RoIs higher-scoring than itself, using the Scaled Dot-Product Attention mechanism. Thus, in our view, this finding supports the validity of the sequence-to-sequence formulation of the NMS task.

V. Conclusions

Detecting humans accurately is crucial for human safety-centric applications, but also extremely challenging. Large variations in human poses and high levels of occlusions negatively affect person detection accuracy. Non-Maximum Suppression (NMS) is the last step in a typical object detection system, which is also affected by such challenges. This paper presented Seq2Seq-NMS, a novel deep neural architecture for performing NMS in similar hard cases, relying on a reformulation of NMS as a sequence-to-sequence problem. The proposed method utilises the Multihead Scaled Dot-Product Attention mechanism, in order to efficiently capture interrelations across the sequence of candidate detections, while also jointly exploiting visual appearance and geometric properties of the input RoIs in order to better represent them. Quantitative evaluation on three public person detection datasets showed that Seq2Seq-NMS can provide state-of-the-art results at the IoU threshold used for annotating its training dataset, with acceptable inference runtime requirements. Future extensions may focus on a training strategy suitable for various IoU thresholds and on adapting the proposed method to multiclass object detection.

VI. Acknowledgments

This project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No. 871449 (OpenDR). This publication reflects the authors' views only. The European Commission is not responsible for any use that may be made of the information it contains.

References

[1] S. Liu, D. Huang, and Y. Wang, "Adaptive NMS: Refining pedestrian detection in a crowd," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[2] J. Hosang, R. Benenson, and B. Schiele, "Learning Non-Maximum Suppression," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[3] C. Symeonidis, I. Mademlis, N. Nikolaidis, and I. Pitas, "Improving neural Non-Maximum Suppression for object detection by exploiting interest-point detectors," in *Proceedings of the IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, 2019.

[4] H. Hu, J. Gu, Z. Zhang, J. Dai, and Y. Wei, "Relation networks for object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[5] T.-Y. Lin, M. Maire, S. J. Belongie, L. D. Bourdev, R. B. Girshick, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2014.

[6] M. Everingham, L. Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (VOC) challenge," *International Journal of Computer Vision*, vol. 88, pp. 303–338, 2009.

[7] I. Mademlis, N. Nikolaidis, and I. Pitas, "Stereoscopic video description for key-frame extraction in movie summarization," in *Proceedings of the EURASIP European Signal Processing Conference (EUSIPCO)*, 2015.

[8] I. Mademlis, N. Nikolaidis, A. Tefas, I. Pitas, T. Wagner, and A. Messina, "Autonomous unmanned aerial vehicles filming in dynamic unstructured outdoor environments," *IEEE Signal Processing Magazine*, vol. 36, pp. 147–153, 2018.

[9] C. Symeonidis, E. Kakaletsis, I. Mademlis, N. Nikolaidis, A. Tefas, and I. Pitas, "Vision-based UAV safe landing exploiting lightweight deep neural networks," in *Proceedings of the International Conference on Image and Graphics Processing (ICIGP)*, 2021.

[10] C. Papaioannidis, I. Mademlis, and I. Pitas, "Autonomous UAV safety by visual human crowd detection using multi-task deep neural networks," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2021.

[11] E. Kakaletsis, E. Symeonidis, M. Tzelepi, I. Mademlis, T. A., N. Nikolaidis, and I. Pitas, "Computer vision for autonomous UAV flight safety: An overview and a vision-based safe landing pipeline example," *ACM Computing Surveys*, 2021.

[12] I. Mademlis, V. Mygdalis, N. Nikolaidis, and I. Pitas, "Challenges in autonomous UAV cinematography: an overview," in *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)*, 2018.

[13] P. Nousi, I. Mademlis, I. Karakostas, A. Tefas, and I. Pitas, "Embedded UAV real-time visual object detection and tracking," in *Proceedings of the IEEE International Conference on Real-time Computing and Robotics (RCAR)*, 2019.

[14] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005.

[15] P. Viola and M. Jones, "Robust real-time face detection," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2001.

[16] P. Felzenszwalb, D. McAllester, and D. Ramanan, "A discriminatively trained, multiscale, deformable part model," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[17] S. Tang, M. Andriluka, A. Milan, K. Schindler, S. Roth, and B. Schiele, "Learning people detectors for tracking in crowded scenes," in *Proceedings of the International Conference on Computer Vision (ICCV)*, 2013, pp. 1049–1056.

[18] 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, 2017.

[19] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. E. Reed, C.-Y. Fu, and A. Berg, "SSD: Single shot multibox detector," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016.

[20] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," *ArXiv*, vol. abs/1804.02767, 2018.

[21] A. Bochkovskiy, C.-Y. Wang, and H. Liao, "YOLOv4: Optimal speed and accuracy of object detection," *ArXiv*, vol. abs/2004.10934, 2020.

[22] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1627–1645, 2010.

[23] N. Bodla, B. Singh, R. Chellappa, and L. S. Davis, "Soft-NMS: Improving object detection with one line of code," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017.

[24] H. Zhou, Z. Li, C. Ning, and J. Tang, "CAD: Scale invariant framework for real-time object detection," in *Proceedings of the IEEE International Conference on Computer Vision Workshops (ICCVW)*, 2017.

[25] B. Jiang, R. Luo, J. Mao, T. Xiao, and Y. Jiang, "Acquisition of localization confidence for accurate object detection," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

[26] D. Bolya, C. Zhou, F. Xiao, and Y. J. Lee, "YOLACT: Real-time instance segmentation," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2019.

[27] Z. Zheng, P. Wang, W. Liu, J. Li, R. Ye, and D. Ren, "Distance-IoU loss: Faster and better learning for bounding box regression," *Association for the Advancement of Artificial Intelligence (AAAI)*, vol. 34, no. 07, 2020.

[28] Z. Zheng, P. Wang, D. Ren, W. Liu, R. Ye, Q. Hu, and W. Zuo, "Enhancing geometric factors in model learning and inference for object detection and instance segmentation," *IEEE Transactions on Cybernetics*, pp. 1–13, 2021.

[29] S. H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese, "Generalized intersection over union: A metric and a loss for bounding box regression," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[30] X. Huang, Z. Ge, Z. Jie, and O. Yoshie, "NMS by representative region: Towards crowded pedestrian detection by proposal pairing," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[31] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proceedings of the International Conference on Neural Information Processing Systems (NIPS)*, 2017.

[32] I. Mademlis, A. Tefas, N. Nikolaidis, and I. Pitas, "Compact video description and representation for automated summarization of human activities," in *Proceedings of the INNS Conference on Big Data*, 2016.

[33] I. Mademlis, A. Tefas, N. Nikolaidis, and I. Pitas, "Multimodal stereoscopic movie summarization conforming to narrative characteristics," *IEEE Transactions on Image Processing*, vol. 25, no. 12, pp. 5828–5840, 2016.

[34] I. Mademlis, A. Tefas, and I. Pitas, "A salient dictionary learning framework for activity video summarization via key-frame extraction," *Information Sciences*, vol. 432, pp. 319 – 331, 2018.

[35] I. Mademlis, A. Tefas, and I. Pitas, "Regularized SVD-based video frame saliency for unsupervised activity video summarization," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2018.

[36] ——, "Greedy salient dictionary learning with optimal point reconstruction for activity video summarization," in *Proceedings of the IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, 2018.

[37] K. He, G. Gkioxari, P. Dollar, and R. Girshick, "Mask R-CNN," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017.

[38] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond Bags of Features: Spatial pyramid matching for recognizing natural scene categories," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006.

[39] C.-F. Chen, Q. Fan, and R. Panda, "CrossVit: Cross-attention multi-scale vision transformer for image classification," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 357–366.

[40] N. Parmar, A. Vaswani, J. Uszkoreit, L. Kaiser, N. S. A. Ku, and D. Tran, "Image transformer," in *Proceedings of the International Conference on Machine Learning (ICML)*, 2018.

[41] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[42] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," 2016.

[43] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, no. 56, pp. 1929–1958, 2014.

[44] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015.

[45] J. M. Ferryman and A. Ellis, "PETS2010: Dataset and challenge," in *Proceedings of the IEEE International Conference on Advanced Video and Signal-Based Surveillance (AVSS)*, 2010.