Neural Attention-driven Non-Maximum Suppression for Person Detection

This paper was downloaded from TechRxiv (<https://www.techrxiv.org>).

LICENSE

CC BY-NC-SA 4.0

SUBMISSION DATE / POSTED DATE

05-11-2021 / 06-11-2021

CITATION

Symeonidis, Charalampos; Mademlis, Ioannis; Pitas, Ioannis; Nikolaidis, Nikos (2021): Neural Attention-driven Non-Maximum Suppression for Person Detection. TechRxiv. Preprint.
<https://doi.org/10.36227/techrxiv.16940275.v1>

DOI

[10.36227/techrxiv.16940275.v1](https://doi.org/10.36227/techrxiv.16940275.v1)

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 dense human crowds, 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 neighbouring 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 and good behaviour for large numbers of raw candidate ROIs per image.

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

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 these 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*.
- 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 within a crowd, due to various levels of occlusions. The majority of existing NMS methods are oriented towards fast execution, but person detection in human crowds 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] [14] [15]. 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, but 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.

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 [16] [17] relied mainly on local handcrafted descriptors and discriminative classifiers. The Deformable Part-based Model (DPM) [18] is a special case, where an object is represented by its component parts arranged in a deformable configuration. In [19], 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., [20] [21]) 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 [22] [23] [24] and [25] 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 [26]. 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 [27], a rescore 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 [28], 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 [29], 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 [30], 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 spatial distribution of edges or interest-points 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.

(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.

More recently, [31] 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 [32], 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., [21] [22] [24]). However, [33] 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, [33] proposes the *Generalized-IoU* (GIoU) loss term, which handles similar scenarios but suffers from slow convergence and inaccurate regression. Thus, in [31], 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. [31] 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. Motivation

State-of-the-art object detectors continue to require NMS as a final step [25], 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 to images depicting dense human crowds, with high levels of occlusions [1] [34]; 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] [32] [31] [30], 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 [30] [31], 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 [35], 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.

Based on the above considerations, this paper presents: a) a reformulation of the NMS task as a sequence-to-sequence

problem, and b) a novel DNN architecture for solving it, called *Seq2Seq-NMS*. 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 based on their scalar confidence scores assigned by the detector:

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

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 spatial 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:

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

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, in order to solve the NMS task. 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 were used, which were computed on the spatial distribution of edges or interest-points 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. 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.

In [3], a CPU implementation of the low-level FMoD visual descriptor was employed for representing each candidate ROI. FMoD was originally devised in a global [7] and in a local [36] variant (LMoD), respectively applied to movie [37] and activity video [38] [39] [40] 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. In both [3] and in this paper, only the luminance channel is employed for the special use-case of describing a ROI interest-point map instead of a typical image/video frame. The intent is to compactly capture the spatial distribution of the interest-points within the ROI interest-point map in a single numerical description vector. However, in [3] ROIs were processed sequentially and not simultaneously, thus achieving very long inference times, nowhere close to real-time. To tackle this limitation in this paper, FMoD was re-implemented neurally so that it can run in parallel on modern GPUs and, given as input a set of candidate ROIs of different shape and scale, simultaneously extract several feature maps before computing in parallel their FMoD descriptors/representations. This fast neural implementation was rendered feasible thanks to the recently proposed operations [20] and [41].

The appearance-based ROI representations extraction process can be divided into three separate operations. The first one involves the computation of edges/interest-point maps of the input RGB image, which is a relatively fast and efficient process. The second step is using the *ROIAlign* [41] 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.

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 $\mathbf{B} = [\mathbf{b}_0, \mathbf{b}_1, \dots, \mathbf{b}_N] \in \mathbb{R}^{N \times 4}$
 (c) FMoD pyramid levels $L, L \geq 1$
Output: Appearance-based representations
 $\mathbf{A} \in \mathbb{R}^{N \times 5(4^L - 1)}$

```

1 begin
2   Resize image  $\mathbf{I}$  to a fixed size of  $W_f \times H_f$ .
3    $E(\mathbf{I}) \leftarrow$  Compute the edge/interest-point map of image  $\mathbf{I}$ .
4   Extract in parallel the  $0^{th}$ -level ROI maps
      $\mathbf{M}^0 = [\mathbf{M}_0^0, \mathbf{M}_1^0, \dots, \mathbf{M}_N^0]$ , where  $\mathbf{M}_i^0 \in \mathbb{R}^{1 \times W_0 \times H_0}$ ,
     through the ROIAlign operator on  $E(\mathbf{I})$ .
5   Compute in parallel the  $0^{th}$ -level FMoD
     representations  $\mathbf{A}^0 = [\mathbf{A}_0^0, \mathbf{A}_1^0, \dots, \mathbf{A}_N^0]$  of  $\mathbf{M}^0$ ,
     where  $\mathbf{A}_i^0 \in \mathbb{R}^{15}$ .
6   for  $j \leftarrow 1$  to  $(L - 1)$  do
7     Extract in parallel the  $j^{th}$ -level ROI maps
        $\mathbf{M}^j = [\mathbf{M}_0^j, \mathbf{M}_1^j, \dots, \mathbf{M}_N^j]$ , 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.
8     Compute in parallel the  $j^{th}$ -level FMoD
       representations  $\mathbf{A}^j = [\mathbf{A}_0^j, \mathbf{A}_1^j, \dots, \mathbf{A}_N^j]$  of  $\mathbf{M}^j$ ,
       where  $\mathbf{A}_i^j \in \mathbb{R}^{15 \times 4^j}$ .
9   end
10  Concatenate FMoD representations across all
     pyramid levels  $\mathbf{A} \in \mathbb{R}^{N \times 5(4^L - 1)}$ , where
      $\mathbf{A}_i = [\mathbf{A}_i^0, \dots, \mathbf{A}_i^L]$ .
11 end
```

- (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.

The corresponding procedure is described on 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/interest-point 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 [42]. At the pyramid base, the 0^{th} -level ROI maps $\mathbf{M}^0 = [\mathbf{M}_0^0, \mathbf{M}_1^0, \dots, \mathbf{M}_N^0]$, $\mathbf{M}_i^0 \in \mathbb{R}^{1 \times W_0 \times H_0}$ are extracted in parallel by applying the *ROIAlign* operator on $E(\mathbf{I})$, assuming that N candidate ROIs (expressed as rectangles in 2D pixel coordinates) have been identified by the object detector for

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.

input $\mathbf{I}$. Using $\mathbf{M}^0$, the 0^{th} -level FMoD representations $\mathbf{A}^0 = [\mathbf{A}_0^0, \mathbf{A}_1^0, \dots, \mathbf{A}_N^0]$, $\mathbf{A}_i^0 \in \mathbb{R}^{15}$ 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 L pyramid levels, where L is predefined and fixed, they are concatenated along them. For example, in an image with $N = 5$ candidate ROIs and $L = 2$ pyramid levels, $\mathbf{A} \in \mathbb{R}^{5 \times 75}$. This example is illustrated in Figure 2.

B. Geometry-based ROI Representations Extraction

The spatial/geometric interrelations between the various candidate ROIs detected on an input image, 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 the GossipNet neural architecture [2]. Thus, in this paper, a slightly similar, but enriched set of attributes has been devised, serving as an additional representation for each candidate ROI.

Given a set of N candidate ROIs expressed in 2D pixel coordinates, 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 distances¹ defined 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 width/height/area/aspect-ratio differences (e.g., $\frac{w_j}{w_i}$) between the j^{th} and the i^{th} ROI.

¹Horizontal and vertical distances are signed distances.

- (12) the detector's confidence score for the j^{th} ROI.
- (13) the detector's confidence score differences between the j^{th} (e.g., $s_j - s_i$) and the i^{th} ROI.
- (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 them, instead of directly pruning them. For a given set of N candidate ROIs detected on an input image, the DNN receives as input a sequence of corresponding representations ($\mathbf{A}$ and $\mathbf{G}$, encoding the appearance and the 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 provided set. The Scaled Dot-Product Attention mechanism [35], originally proposed for machine translation tasks, is employed to this end, since it has been proven effective in various applications, such as image classification [43], or image generation [44]. It is briefly described in the next Subsection. 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.

Multihead Self-Attention Module: The Scaled Dot-Product Attention, also known as self-attention, was presented in [35] and formulated as follows:

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

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 $\mathbf{V} \in \mathbb{R}^{N_v \times d_v}$ are the values. Each query and each key has a dimension of d_k , while each value has a dimension of d_v . Multihead Attention was also proposed in [35], as a module which allows various attention mechanisms, including self-attention, to run in parallel. This module can be formulated as:

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

where

$$\mathbf{h}_i = Attention(\mathbf{QW}_i^Q, \mathbf{KW}_i^K, \mathbf{VW}_i^V). \quad (6)$$

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

In this formulation, $\mathbf{W}_i^Q \in \mathbb{R}^{d_a \times d_k}$, $\mathbf{W}_i^K \in \mathbb{R}^{d_a \times d_k}$, $\mathbf{W}_i^V \in \mathbb{R}^{d_a \times d_v}$, $\mathbf{W}_i^O \in \mathbb{R}^{H d_v \times d_a}$ 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 on their geometric properties. Such relations can help the model in determining whether a candidate detection should be suppressed or not. For example, the DNN can decide that a higher-scoring candidate ROI should possibly suppress other less-scoring detections having similar appearance and geometric representations.

In [35] the authors introduced *positional encoding* for Natural 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 in the image 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 [35]. As an alternative, in this paper we approached the task by encoding all the representations of the N 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 presented method's memory complexity, each query is allowed to represent the keys and the values relatively to itself. Thus, for N candidate 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 $\mathbf{F}^D \in \mathbb{R}^{N \times 1 \times d_a}$. A residual connection [45] is applied between $\mathbf{Q}$ and $\mathbf{F}^D$. 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.

Fig. 4: Illustration of the Joint Processing Module.

Joint Processing Module: In this neural module, the representations of the candidate detections are jointly and simultaneously refined, mainly through the Multihead Self-Attention mechanism. The Joint Processing Module 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-candidate-detection representations, for all N candidate detection. The architecture of the Joint Processing Module is shown in Fig. 4. The queries and keys are formed as:

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

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

$$\begin{aligned} \mathbf{F}_{t+1}^Q &= \mathbf{F}^D \mathbf{C}^D + \mathbf{F}_t^Q, \\ \mathbf{F}^D &= \text{Multihead}(\mathbf{Q}, \mathbf{K}, \mathbf{V}), \end{aligned} \quad (8)$$

where $\mathbf{C}^D \in \mathbb{R}^{d_a \times d_m}$ also denotes the weights of a fully connected layer.

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

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

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 $\mathbf{F}_{t+1}^K$ in each Joint Processing Module.

Masking: A masking approach has been integrated into the self-attention mechanism of the proposed architecture. For N candidate detections, sorted in descending order based on the detector’s score values, 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 the number of heads $H = 1$.

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

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

This masking operation is employed for two reasons. First, each ROI must be prevented from attending to spatially distant candidate 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 $\mathbf{r}_i^{in}$ to $\mathbf{r}_j^{in}$ (after the softmax has been applied) will be zeroed out. Second, an additional motivation is our attempt to replicate the behaviour of Greedy NMS, where a candidate detection is characterized as duplicate, thus marked for suppression, when another, higher-scoring detection spatially covers the same object. In the proposed neural 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 detections, sorted in descending order based on the detector’s confidence score values, the proposed DNN uses as input their corresponding appearance-based representations $\mathbf{A}$ and their 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, which is $\mathbf{G} \in \mathbb{R}^{N \times N \times 14}$, 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 Joint Processing Module.

A stack of Joint Processing Modules, 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 described architecture of the model is depicted in Fig. 5. The Gaussian Error Linear Unit (GELU) is used as activation function by the network. Layer normalization [46] is applied on the output of residual connections and dropout [47] is used for regularization, similarly to [35].

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 threshold was set to 0.5. The class weights, are defined as:

$$w_j = (1-j)(1-c_j) \frac{\sum_{m=1}^M (\text{count}(\mathbf{G}_m) + \max(0, \text{count}(\mathbf{R}_m) - \text{count}(\mathbf{G}_m)))}{\sum_{m=1}^M (\max(0, \text{count}(\mathbf{R}_m)) - \text{count}(\mathbf{G}_m))} + j c_j \frac{\sum_{m=1}^M (\text{count}(\mathbf{G}_m) + \max(0, \text{count}(\mathbf{R}_m) - \text{count}(\mathbf{G}_m)))}{\sum_{m=1}^M \text{count}(\mathbf{G}_m)}, \quad (12)$$

where, M is the number of images in a dataset, $\mathbf{R}_m$ are the candidate ROIs in the m^{th} image, $\mathbf{G}_m$ are the ground-truth ROIs in the m^{th} image, and c_1 is a predefined scalar (e.g., 0.1) to balance the class weights.

IV. EXPERIMENTAL EVALUATION

The performance of Seq2Seq-NMS was evaluated on three separate datasets for the person detection task. In order to assess its accuracy regardless of the selected object detector, candidate ROIs from three different detectors were employed. These differ significantly in the way they approach the object detection task in general.

The neural architecture used for evaluation 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 dataset, Seq2Seq-NMS was compared against both neural and non-neural NMS algorithms. While some are focused more on achieving state-of-the-art results, others succeed in attaining ultra-fast inference times. The first competing method is a baseline Greedy NMS approach running on GPU. The second is TorchVision’s² GreedyNMS implemented to run very fast on GPUs. Additionally, Soft-NMS [27] was tested, i.e., a non-neural NMS method widely used as a more accurate replacement for Greedy NMS. Evaluation was conducted

²<https://pytorch.org/vision/stable/ops.html#torchvision.ops.nms>

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

using both the linear and the Gaussian weighting functions (referred to as Soft-NMS_L and Soft-NMS_G , respectively). The method was executed on CPU. Another competing algorithm we employed is Fast-NMS [30], a generally faster, non-neural replacement for standard NMS, that suffers a marginal penalty regarding accuracy. Fast-NMS is executed on GPU. Additionally, several variants of Cluster-NMS [32], i.e., a more recent non-neural method, was also selected for comparison purposes. Below, the term Cluster-NMS_S is used to imply the case where the score penalty mechanism is used, and Cluster-NMS_D the scenario where the normalized central point distance is added. In the latter case, the method is equivalent to DIOU-NMS [31]. Moreover, the term Cluster-NMS_{S+D} is used when both of these mechanisms are utilized. Finally, $\text{Cluster-NMS}_{S+D+W}$ indicates the case where a weighted strategy similar to [28] is followed. More details regarding these variations can be found in [32]. 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. Both for GossipNet and for the proposed method, the class weights are computed through Eq. (12), using a common, dataset-specific c_1 value. Evaluation was performed on a PC using an Intel Core i7-7700 CPU and an NVIDIA GeForce GTX 1070 Ti GPU with 8GB of memory, both for training and inference. The employed evaluation metrics are $\text{AP}_{0.5}$, $\text{AP}_{0.5}^{0.95}$ and inference times. $\text{AP}_{0.5}$ corresponds to the average precision for 0.5 IoU, while $\text{AP}_{0.5}^{0.95}$ to the mean average precision for IoU ranging from 0.5 to 0.95 with a step size of 0.05. Finally, for all methods, all candidate detections were used during the evaluation process, without any thresholding.

A. PETS

PETS [48] 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 about 14. We use candidate detections from [19], a non-neural person detection method, designed specifically to handle occlusions. Due to the fact that the number of ROIs is extremely large in some images, we first apply TorchVision NMS with the relaxed 0.8 IoU threshold on all deployed methods as a typical preprocessing step, commonly utilized in NMS literature. Since the number of candidate detections for the NMS step remains huge even after applying TorchVision NMS, we report results for various ROI retention thresholds, with only a maximum number of higher-scoring detections kept in each case. Typically, candidate ROIs with near-zero confidence scores, as assigned by the detector, are not true positive samples and, thus, do not significantly affect the accuracy of traditional non-neural NMS methods.

The proposed method was trained using the ADAM [49] optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.99$ and $\epsilon = 10^{-9}$ for 8 epochs. The learning rate was set to 10^{-4} for the first 4 epochs, then to 10^{-5} for the next 3 epochs and finally it as to 10^{-6} for the final epoch. At most the 600 highest-scoring candidate detections were only used as the input sequence for each image when training the network, due to memory limitations. Regarding GossipNet, architecture and training process was based on [2].

Table I reports the results of the proposed method, as well as of competing NMS approaches. As mentioned above, the employed object detector [19] 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., < 0.1) which are mostly false positive samples.

TABLE I: COMPARISON OF DIFFERENT NMS METHODS ON THE PETS DATASET, USING DETECTIONS FROM [19]. BOTTOM LINE REPORTS ON THE PROPOSED METHOD.

Method	Device	Max dets. = 400			Max dets. = 600			Max dets.= 800			Max dets. = 1200			Max dets. = All		
		AP _{0.5}	AP _{0.95} _{0.5}	Average Inference Time (ms)	AP _{0.5}	AP _{0.95} _{0.5}	Average Inference Time (ms)	AP _{0.5}	AP _{0.95} _{0.5}	Average Inference Time (ms)	AP _{0.5}	AP _{0.95} _{0.5}	Average Inference Time (ms)	AP _{0.5}	AP _{0.95} _{0.5}	Average Inference Time (ms)
Original NMS IoU>0.4	GPU	75.4%	31.7%	3.8	76.2%	31.6%	2.5	76.5%	31.3%	4.5	76.3%	31.2%	5.0	76.3%	31.2%	6.3
Original NMS IoU>0.5	GPU	73.1%	31.0%	4.1	73.6%	30.8%	3.6	73.6%	30.8%	5.7	73.6%	30.8%	7.5	73.6%	30.8%	9.8
Original NMS IoU>0.6	GPU	65.7%	28.6%	6.8	65.7%	28.6%	5.5	65.7%	28.6%	10.0	65.7%	28.6%	12.8	65.7%	28.6%	16.0
Original NMS IoU>0.4 (TorchVision)	GPU	75.6%	31.7%	0.4	76.1%	31.6%	0.5	76.4%	31.3%	0.5	76.4%	31.3%	0.8	76.4%	31.3%	0.8
Original NMS IoU>0.5 (TorchVision)	GPU	72.7%	30.8%	0.6	73.2%	30.8%	0.5	73.2%	30.8%	0.5	73.2%	30.8%	0.7	73.2%	30.8%	0.8
Original NMS IoU>0.6 (TorchVision)	GPU	65.1%	28.5%	0.5	65.1%	28.5%	0.5	65.1%	28.5%	0.5	65.1%	28.5%	0.7	65.1%	28.5%	0.8
Soft-NMS _L	CPU	76.6%	31.8%	27.1	76.6%	31.5%	50.3	76.5%	31.3%	64.0	76.3%	31.2%	98.5	76.3%	31.2%	143.3
Soft-NMS _G	CPU	77.4%	32.7%	23.8	77.2%	32.3	39.2	76.7%	32.0%	54.4	76.0%	31.8%	89.5	75.8%	31.7%	154.7
Fast-NMS	GPU	75.3%	31.5%	3.2	75.2%	31.1%	1.6	75.0%	31.0%	1.5	74.5%	30.8%	2.8	74.4%	30.7%	3.5
Cluster-NMS	GPU	75.9%	31.4%	2.9	76.4%	31.3%	3.6	76.4%	31.3%	3.7	76.4%	31.3%	5.5	76.4%	31.3%	8.0
Cluster-NMS _S	GPU	75.4%	31.6%	4.8	74.6%	31.2%	2.9	74.0%	30.8%	3.8	73.6%	30.7%	5.0	73.0%	30.6%	7.3
Cluster-NMS _D	GPU	76.2%	31.6%	5.4	76.6%	31.3%	4.1	76.5%	31.2%	5.1	76.5%	31.1%	7.7	76.5%	31.1%	10.0
Cluster-NMS _{S+D}	GPU	76.4%	31.9%	3.7	76.3%	31.6%	4.2	75.9%	31.3%	5.4	75.2%	31.0%	8.4	74.9%	30.9%	12.1
Cluster-NMS _{S+D+W}	GPU	76.4%	31.9%	28.3	76.3%	31.6%	55.7	75.9%	31.3%	88.2	75.2%	31.0%	171.9	74.9%	30.9%	292.3
GossipNet	GPU	79.6%	34.4%	19.1	81.7%	35.1%	32.5	82.8%	35.6%	48.0	83.6%	35.9%	73.9	83.8%	36.1%	107.2
Seq2Seq-NMS	GPU	80.9%	36.5%	11.1	82.9%	37.1%	11.3	83.9%	37.3%	11.9	84.7%	37.3%	14.7	-	-	-

Thus, in this paper, we attempt to evaluate whether the lowest scoring detections have an impact on the performance of the proposed and of the competing NMS methods. In Table I, the results of each method are reported using N candidate detections as input, for different values of N . As it can be seen, the performance of the non-neural methods is not improved when the lowest-scoring detections ($N > 800$) are used. In contrast, both GossipNet and the proposed method achieve more accurate results for longer input sequences (more candidate ROIs per image).

This advantage of the neural NMS methods most likely stems from the inability of non-neural algorithms to jointly process the candidate detections in order to rescore/suppress them. Regarding the proposed method, its performance improves as more relations between a corresponding candidate detection and other neighboring ROIs are built. Additionally, it is able to handle “hard” True Positive (TP) candidate detections and classify them as correct ones, even if the employed detector originally assigned them very low confidence scores.

Overall, the proposed method achieved both the best AP_{0.5} and the best AP_{0.95}_{0.5}, against all competing approaches, when using an input sequence of 1200 detections. The obtained AP_{0.5} was 84.7%, which is a +7.3% improvement against Soft-NMS_G, the non-neural method with the best AP_{0.5}, and a +0.9% improvement against GossipNet. Notably, when using only a small number of the highest-scoring candidate detections (e.g. $N = 400$), the proposed method still achieves better results compared to all non-neural NMS algorithms. Regarding inference runtimes, it needs 14.7 ms to run per image when $N = 1200$, since the required edge maps are computed in parallel with the object detector’s inference. Thus it is faster than GossipNet, as well as less affected by the number of candidate detections used as input. However, it is

slower than the non-neural methods running on GPU.

B. COCO Person

COCO 2014 is a large dataset consisting 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. Candidate detections (extracted by Faster R-CNN [21]) and validation set splits from [2] were also employed here. The first data subset, referred to as “minival”, contains 5000 images, while the second subset, referred to as “minitest”, contains 35000 images. The average ground-truth number of people depicted in an image is about 2.17. However, when taking into account only the images that actually contain visible people, this average ground-truth number is 4.01.

The proposed method was trained using the ADAM optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.99$ and $\epsilon = 10^{-9}$ for 8 epochs. The learning rate was set to 10^{-4} for the first 4 epochs, then to 10^{-5} for the next 3 epochs and, finally, to 10^{-6} for the final epoch. At most the 600 highest-scoring candidate detections per image were employed as input at the training stage. GossipNet architecture and training again followed [2]. Final model parameters were selected according to achieved accuracy in the minival (validation) set during training, both for the proposed method and for GossipNet.

As it is shown in Table II, the proposed method achieves the best AP_{0.5} against all competing NMS methods, in both the minival and the minitest sets. It leads to an AP_{0.5} = 68.1% in the minival set, which is an +0.8% improvement against Soft-NMS_L, the best non-neural method, and +0.4% against GossipNet. In the minitest set, it achieved AP_{0.5} = 67.5%, which is an +0.9% improvement against Soft-NMS_L and +0.6% against GossipNet. As far as AP_{0.95}_{0.5} is concerned, the

proposed method is outperformed only by Cluster-NMS_{S+D+W}, achieving $AP_{0.5}^{0.95} = 37.2\%$ and $AP_{0.5}^{0.95} = 36.9\%$ in the minival and the minitest sets, respectively. These figures are on par with the results of Soft-NMS_L. Regarding inference times, the proposed method is close to that of Cluster-NMS_{S+D+W}.

TABLE II: COMPARISON OF DIFFERENT NMS METHODS ON THE COCO DATASET, USING DETECTIONS FROM FASTER-RCNN. BOTTOM LINE REPORTS ON THE PROPOSED METHOD.

Method	Device	Minival set		Minitest set		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	66.1%	35.7%	65.4%	35.5%	4.3
Original NMS IoU>0.5	GPU	66.0%	36.0%	65.3%	35.8%	4.5
Original NMS IoU>0.6	GPU	64.1%	35.9%	63.2%	35.5%	4.8
Original NMS IoU>0.4 (TorchVision)	GPU	66.2%	35.8%	65.4%	35.6%	0.59
Original NMS IoU>0.5 (TorchVision)	GPU	66.0%	36.0%	65.2%	35.8%	0.60
Original NMS IoU>0.6 (TorchVision)	GPU	64.0%	35.9%	63.1%	35.5%	0.60
Soft-NMS _L	CPU	67.3%	37.2%	66.6%	36.9%	5.8
Soft-NMS _G	CPU	67.0%	37.0%	66.2%	36.7%	6.6
Fast-NMS	GPU	65.0%	35.5%	64.1%	35.3%	2.5
Cluster-NMS	GPU	66.2%	35.6%	65.4%	35.5%	4.9
Cluster-NMS _S	GPU	66.0%	36.0%	65.3%	36.0%	4.8
Cluster-NMS _D	GPU	66.2%	35.8%	65.5%	35.6%	7.6
Cluster-NMS _{S+D}	GPU	66.5%	36.8%	65.8%	36.6%	7.6
Cluster-NMS _{S+D+W}	GPU	66.7%	37.8%	65.8%	37.6%	9.7
GossipNet	GPU	67.7%	36.1%	66.9%	36.0%	6.1
Seq2Seq-NMS	GPU	68.1%	37.2%	67.5%	36.9%	10.1

C. CrowdHuman

The CrowdHuman dataset has been recently released to specifically target human detection in crowded areas. It 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. All methods were trained on the CrowdHuman training set and evaluated on its validation set, since annotations weren't provided for the test set.

Candidate detections were extracted using YOLOv4 [25] for this dataset. The detector was trained for 6000 steps, using the Nesterov Accelerated Gradient [50] (NAG) optimizer with a momentum equal to 0.95. Mini-batch size was set to 64 with 64 subdivisions, while images were re-scaled to a resolution of 608×608 pixels. The learning rate was set to 10^{-3} , decreasing by 0.1 at the 4800-th and the 5400-th step.

The proposed NMS DNN was trained using the ADAM optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-8}$ for 12 epochs. The learning rate was set to 10^{-4} for the first 5 epochs, then to 10^{-5} for the next 5 epochs and finally to 10^{-6} for the final 2 epochs. Once more, at most the 600 highest-scoring candidate detections per image were employed as input at the training stage. During inference, the maximum number of detections per image was set to 800, due to memory constraints. GossipNet was trained for 120000 iterations, with

a learning rate set to 10^{-4} and decreasing by 0.1 at the 60000-th and the 90000-th iteration. Due to the very large number of candidate ROIs on several images, TorchVision NMS with the relaxed 0.8 IoU threshold was again applied as a preprocessing step, before feeding the proposed DNN and the GossipNet model with input. This strategy was followed during both the training and the inference stage.

As shown in Table III, the proposed method achieves the best $AP_{0.5}$, along with GossipNet, against all competing NMS methods. In particular, it leads to an $AP_{0.5}$ of 83.7%, which is a +1.2% improvement against Soft-NMS_L. Moving on to the $AP_{0.5}^{0.95}$ metric, the proposed method was slightly outperformed only by Soft-NMS_L. Overall, it achieves a $AP_{0.5}^{0.95}$ of 49.1%. Regarding inference runtimes, the proposed method is faster than all non-GPU approaches and than GossipNet, since the latter's required inference time depends strongly on the number of candidate detections per image. The average inference time of the proposed method was 11.1 ms, given that the edge maps needed for the appearance-based ROI representations extraction are computed in parallel with the detector's inference stage.

TABLE III: COMPARISON OF DIFFERENT NMS METHODS ON THE CROWDHUMAN DATASET, USING DETECTIONS FROM YOLOv4. BOTTOM LINE REPORTS ON THE PROPOSED METHOD.

Method	Device	Validation set		Average Inference Time (ms)
		$AP_{0.5}$	$AP_{0.5}^{0.95}$	
Original NMS IoU>0.4	GPU	76.9%	44.8%	2.2
Original NMS IoU>0.5	GPU	80.6%	47.0%	2.9
Original NMS IoU>0.6	GPU	82.0%	48.2%	4.6
Original NMS IoU>0.4 (TorchVision)	GPU	77.1%	44.9%	0.4
Original NMS IoU>0.5 (TorchVision)	GPU	80.7%	47.0%	0.4
Original NMS IoU>0.6 (TorchVision)	GPU	82.0%	48.3%	0.4
Soft-NMS _L	CPU	82.5%	49.3%	30.8
Soft-NMS _G	CPU	81.7%	48.6%	39.8
Fast-NMS	GPU	81.0%	48.0%	1.6
Cluster-NMS	GPU	82.0%	48.2%	4.7
Cluster-NMS _S	GPU	80.3%	47.3%	3.4
Cluster-NMS _D	GPU	82.2%	48.6%	5.1
Cluster-NMS _{S+D}	GPU	81.4%	48.2%	5.5
Cluster-NMS _{S+D+W}	GPU	81.4%	48.2%	47.9
GossipNet	GPU	83.7%	49.0%	16.1
Seq2Seq-NMS ₈₀₀	GPU	83.7%	49.1%	11.1

D. Discussion

Overall, the proposed Seq2Seq-NMS DNN achieves top accuracy on the $AP_{0.5}$ metric in all three datasets, being on par with GossipNet in CrowdHuman and outperforming all

competing methods in the remaining two 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.

Besides the measurements reported in Tables I, II and III, an ablation study was also performed regarding the proposed masking operation (as described in Section III-C) of the self-attention mechanism. Omitting masking led to reduced accuracy rates, or to training convergence failures in cases with huge numbers of candidate ROIs per image. Most likely, 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.

Regarding the $AP_{0.5}^{0.95}$ metric, Seq2Seq-NMS outperforms most competing methods but achieves top accuracy only on the PETS dataset. Most likely, this behaviour can be explained by the fact that the labels of the candidate detections for Seq2Seq-NMS training were created based on [5] using an IoU threshold of 0.5.

Moving on to required inference running times, 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, the inference runtime of Seq2Seq-NMS seems less affected by the input sequence length (number of candidate detections), thus achieving faster inference when processing longer sequences, as shown in Tables I and III.

Finally, evaluation in this paper was performed using candidate detections from: (a) a non-neural detector [19], (b) a two-stage DNN-based detector [21], and (c) a one-stage DNN-based detector [25]. Therefore, it can be claimed that the behaviour of Seq2Seq-NMS is rather invariant to the type of the deployed person detector, achieving an improved $AP_{0.5}$ performance against several competing NMS methods. This can be easily explained by the use of FMoD for visual appearance representation of the cropped candidate ROIs, instead of relying on learnt convolutional representations computed by the DNN-based person detector.

V. CONCLUSIONS

Non-Maximum Suppression (NMS) is the last step in a typical object detection system. Heavy occlusions when detecting humans in images of crowded areas imposes great challenges to most NMS methods, despite the importance of such a task for human safety-centric applications. This paper presented Seq2Seq-NMS, a novel deep neural architecture for performing NMS in similar hard cases, which relies on

reformulating NMS as a sequence-to-sequence problem. The proposed method relies on 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, each one using a different detector, 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 and good behaviour for large numbers of raw candidate ROIs per image. Future extensions may focus on a training strategy suitable for various IoU thresholds, as well as on assessing method performance when using a different modality for describing the visual appearance of the cropped input detections (instead of edge maps). This may enable an extension of the proposed method to multiclass object detection tasks.

VI. ACKNOWLEDGMENTS

This project has received funding from the European Unions 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)*. Springer, 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)*. IEEE, 2015.
- [8] L. Chen, S. Lin, X. Lu, D. Cao, H. Wu, C. Guo, C. Liu, and F.-Y. Wang, "Deep neural network-based vehicle and pedestrian detection for autonomous driving: A survey," *IEEE Transactions on Intelligent Transportation Systems*, 2021.
- [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, accepted.
- [12] P. Nousi, E. Patsiouras, A. Tefas, and I. Pitas, "Convolutional Neural Networks for visual information analysis with limited computing resources," in *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, 2018.
- [13] 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.
- [14] 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.
- [15] 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.
- [16] 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.
- [17] P. Viola and M. Jones, "Robust real-time face detection," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2001.
- [18] 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.
- [19] 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.
- [20] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.
- [21] 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.
- [22] 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)*. Springer, 2016.
- [23] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.
- [24] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," *ArXiv*, vol. abs/1804.02767, 2018.
- [25] A. Bochkovskiy, C.-Y. Wang, and H. Liao, "YOLOv4: Optimal speed and accuracy of object detection," *ArXiv*, vol. abs/2004.10934, 2020.
- [26] 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.
- [27] 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.
- [28] 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.
- [29] 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)*. Springer, 2018.
- [30] 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.
- [31] "Distance-IoU loss: Faster and better learning for bounding box regression," *Association for the Advancement of Artificial Intelligence (AAAI)*, vol. 34, no. 07, pp. 12 993–13 000, 2020.
- [32] 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," *ArXiv*, vol. abs/2005.03572, 2020.
- [33] 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.
- [34] 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.
- [35] 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.
- [36] 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*. Springer, 2016.
- [37] 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.
- [38] 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.
- [39] 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.
- [40] —, "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.
- [41] K. He, G. Gkioxari, P. Dollar, and R. Girshick, "Mask R-CNN," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017.
- [42] 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.
- [43] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021.
- [44] 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.
- [45] 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.
- [46] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," 2016.
- [47] 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.
- [48] 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.
- [49] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015.
- [50] Y. Nesterov, "A method for unconstrained convex minimization problem with the rate of convergence $O(1/k^2)$," *Doklady AN USSR*, vol. 269, pp. 543–547, 1983.