ARM-NMS: SHAPE BASED NON-MAXIMUM SUPPRESSION FOR INSTANCE SEGMENTATION IN LARGE SCALE IMAGERY

Andreas Michel[1,2]*, Wolfgang Gross[1], Stefan Hinz[2], Wolfgang Middelmann[1]

[1] Fraunhofer IOSB – (andreas.michel, wolfgang.gross, wolfgang.middelmann)@iosb.fraunhofer.de
[2] Karlsruhe Institute of Technology – stefan.hinz@kit.edu

Commission II, WG II/6

KEY WORDS: Non-Maximum Suppression, shape-based NMS, Instance Segmentation, Object Detection, Deep Learning.

ABSTRACT:

Detecting objects in aerial scenes is a fundamental and critical task in remote sensing. However, state-of-the-art object detectors are susceptible to producing correlated scores in neighboring detections resulting in increased false positives. In addition, detection on large-scale images requires a tiling scheme with usually overlapping windows, consequently creating more double detections. Therefore, a non-maximum suppression (NMS) approach can be exploited as integral to the detection pipeline. NMS suppresses overlapping detections in regards to their scores. Current NMS algorithms filter detections by utilizing their corresponding bounding boxes. However, one can assume that comparing bounding boxes to determine the overlap of non-rectangular objects involves a certain degree of inaccuracy. Therefore, we propose Area Rescoring Mask-NMS (ARM-NMS), which uses object shapes for filtering. ARM-NMS exploits instance masks instead of the conventional boxes to eliminate detections and does not require retraining for instance segmentation pipelines. To exhibit the effectiveness of our approach, we evaluate our method on the large-scale aerial instance segmentation dataset iSaid. Our approach leads to considerable improvements for the COCO-style mAP metric of 3.3 points for segmentations and 3.5 points for boxes.

1. INTRODUCTION

Continuous advancements in camera, aircraft, and satellite technologies enable the collection of large amounts of electro-optical data of vast areas on the earth's surface. Nevertheless, the immense data quantity prevents a comprehensive manual analysis. However, computer vision methods provide a reasonable solution to approach this task. Object detection methods, for instance, are an efficient way to extract valuable position and classification details of visual data. State-of-the-art object detection methods are deep learning-based and can be grouped into one- and two-stage detectors. The main difference between one- and two-stage detectors is that two-stage detectors localize and classify objects in separate steps, while one-stage detectors perform both tasks simultaneously. One-stage detectors (Redmon et al., 2016) operate more time-efficient than their counterpart by exploiting only a dense prediction head. In contrast, the two-stage detector from (Ren et al., 2015) utilizes the dense prediction head only to predict proposals. These proposals serve as a basis for the following sparse prediction head. Overall, two-stage detectors are more accurate, but the extensive number of proposals usually require a filter strategy to reduce the computational complexity to an acceptable level. In addition, further filtering is integrated into the postprocessing in both object detector variants. This process is handled by non-maximum suppression (NMS), which is a crucial part of the object detection pipeline. It is an efficient method to eliminate overlapping detections. It usually utilizes the detected bounding boxes to determine overlapping detections and eliminates unnecessary ones. The elimination process is based on the confidence scores of the detections and the degree of overlap between two detections.In this work, we propose an improved approach to using bounding boxes by utilizing object shapes. Furthermore, the expected input of object detectors is usually limited by resolution. Input images with a higher resolution than the maximum resolution defined by detector architectures are usually resized. Resizing can lead to a loss of information or result in scaling issues. Object detectors based on convolutional neural networks are not scale-aware. Therefore, huge divergences in the size of objects between training and interference samples lead to a decrease in the detection performance. Techniques like the feature pyramid

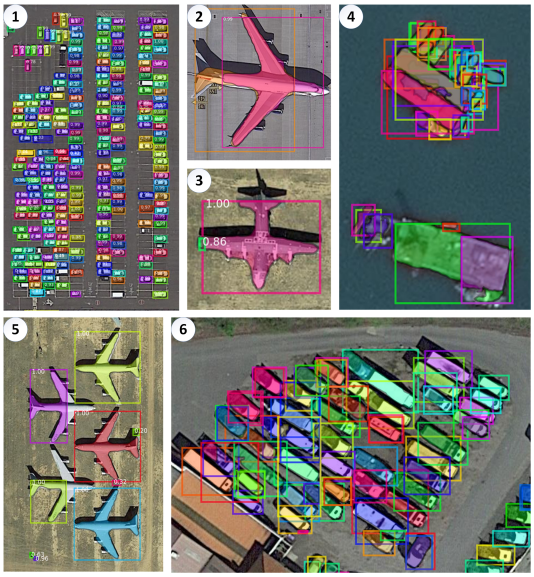

Figure 1. **Challenges for non-maximum suppression** (1) Ideal case for box-based NMS. (2) Double detections. (3) Partial detections. (4) Cluster of objects. (5) Overlapping concave objects. (6) Diagonally-aligned objects.

* Corresponding author

network (Lin et al., 2017) can be used to mitigate the scaling issue but cannot completely resolve it. Even transformer-based backbonesinstead of convolutional-based ones like the swin-transformer (Liu et al., 2021) are not able to completely overcome the scaling issues.

An alternative to resizing is to partition large-scale images into patches on which the detectors operate separately. Fusing the discrete detections into the original scenes leads to additional challenges. Utilizing non-overlapping patches lead to missing detections on the border. In contrast, overlapping patches result in duplicate detections, which the NMS can address. However, filtering overlapping detections comes with many difficulties. Figure 1 illustrates some of the main challenges for NMS:

- **The ideal case** for box-based NMS would be rectangular non-overlapping objects parallel aligned to the border of the image. The proposed bounding boxes can precisely cover the present objects.
- **Overlapping detections** are the default-case addressed by NMS. Ideally, the better-aligned detection exhibits a higher confidence score and suppresses neighboring scores with lower confidence. However, if both detections have a similar score or the worse-aligned detection scores higher, the elimination process can lead to errors in the final detections. Also, most applied NMS algorithms filter each class separately, which leads to double detections with different labels being ignored by NMS.
- **Partial detections** are particularly hard for NMS. NMS utilizes intersection over union (IoU) as an overlap metric. IoU is size-independent, which is generally considered a positive characteristic. However, in the case of detections that differ strongly in size, the IoU between both is low, and thus partial detections are often overlooked by NMS.
- **Cluster of objects** are another issue for NMS algorithms. Many overlapping objects can lead to false detections due to inaccuracies in the alignment of the detection boxes.
- **Overlapping concave objects** are a significant issue of box-based NMS. The detection boxes are usually not well-aligned to the underlying objects, and the detection of neighboring objects can be suppressed.
- **Diagonally-aligned objects** in high object densities exhibit a substantial overlap of detected bounding boxes without actual overlapping objects. This can easily lead to wrongfully eliminated detections.

In order to address these cases and inspired by Mask-NMS (Wang et al., 2020a), we propose the shape-based non-maximum suppression algorithm, Area Rescoring Mask-NMS (ARM-NMS). ARM-NMS cannot be directly used in object detection algorithms, but object detection can be easily expanded to instance segmentation by predicting the shapes of the detected objects. Typical representatives of instance segmentation are Mask R-CNN (He et al., 2017) and DetectorRS (Qiao et al., 2021). Our proposed method does not require any retraining and can thus easily be implemented into existing instance segmentation pipelines. We provide a solution to keep the computational complexity at an acceptable level, even with the more elaborate task of comparing shapes instead of boxes. Furthermore, we examine an additional overlap metric and further modifications to NMS.
Contributions. (1) We propose an improved shaped based NMS approach for instance segmentation, ARM-NMS. Our approach does not require any retraining and can be easily integrated into instance segmentation pipelines. (2) We propose an area-rescoring strategy to consider the size of an object in order to reduce partial detections. Area-rescoring can be easily integrated into ARM-NMS. (3) In order to display the effectiveness of ARM-NMS, we compare our method with box-based greedy-NMS and soft-NMS on the instance segmentation dataset iSaid.

2. RELATED WORK

In this section, we survey the most relevant works for NMS. NMS is an integral part of many detection computer vision algorithms. It is rooted in edge detection techniques (Rosenfeld and Thurston, 1971), and further developments lead to the following algorithms.

2.1 Greedy-NMS

Greedy-NMS (Dalal and Triggs, 2005) is still to this day a widely used approach in state-of-the-art object detectors to filter unnecessary objects proposals. The basic idea behind greedy-NMS is that bounding boxes with a high detection score suppress their overlapping neighbors with lower scores. The algorithm starts with sorting the bounding box detections $\mathbf{b} \in \mathcal{B}$ regarding their corresponding scores $s \in \mathcal{S}$ in descending order. In an iterative procedure, the bounding box $\mathbf{h}_b$ with the highest score s_{max} is transferred to the list of the final detections $\mathcal{F}$. To determine whether a box is eligible for the elimination process, the intersection over union (IoU) between the detected bounding box $\mathbf{h}_b$ with the maximum score and the remaining boxes are calculated. The new score $s_i \in \mathcal{S}$ of the i-th remaining detection $\mathbf{b}_i \in \mathcal{B}$ can be described as follows

$$s_i \leftarrow \begin{cases} s_i, & iou(\mathbf{h}_b, \mathbf{b}_i) < n_t \\ 0, & iou(\mathbf{h}_b, \mathbf{b}_i) \geq n_t \end{cases}, \tag{1}$$

where $n_t \in [0, 1]$ is the desired threshold value. This procedure is repeated until the list of initial detections $\mathcal{B}$ is empty. Although that greedy-NMS is an efficient and popular method, the characteristics of the hard threshold n_t can lead to errors. Applying a high n_t may lead to keeping many false-positive bounding boxes. In contrast, a low n_t can prevent false-positives from being deleted.

2.2 Soft-NMS

Soft-NMS was developed to address the difficulties of greedy-NMS. Instead of eliminating overlapping bounding boxes, the scores of the lower-scored bounding boxes overlapping the maximum score box $\mathbf{h}_b$ are reduced. This leads to a more continuous penalty function and increases the average precision of the detection.

On the downside, due to not removing the overlapping box from $\mathcal{B}$, the computational complexity of soft-NMS is slightly higher than greedy-NMS. The Gaussian soft-NMS penalty function can be written as follows

$$s_i \leftarrow s_i e^{-\frac{iou(\mathbf{h}_b, \mathbf{b}_i)^2}{\sigma}}, \forall \mathbf{b}_i \notin \mathcal{F}, \tag{2}$$

where σ is a less sensitive threshold parameter than n_t. Similar to greedy-NMS this update rule is applied until all detections from $\mathcal{B}$ fall below a certain threshold or are transferred to $\mathcal{F}$.

2.3 Further Developments

Recent approaches like softer-NMS (He et al., 2018), adaptive-NMS (Liu et al., 2019), or IoU-Net (Jiang et al., 2018) modify the object-detection model but need intensive retraining for minor performance increases. Methods like cluster-NMS (Zheng et al., 2021) try to accelerate NMS but perform similar to greedy-NMS. Another method worth mentioning is weighted boxes fusion (WBF) (Solovyev et al., 2021). Instead of eliminating or rescoring neighboring boxes, WBF merges overlapping boxes into new boxes. WBF excels in test-time-augmentation settings, but it is slightly worse than soft-NMS under conventional conditions. In consequence, greedy-NMS and soft-NMS are still among the most applied methods in object detection. Nevertheless, all mentioned methods utilize boxes instead of shapes. Recently, a few approaches utilize shape-based NMS, for instance, mask-based NMS (Wang et al., 2020a), (Tian et al., 2020), or Matrix-NMS (Wang et al., 2020b). The goal of this work is to improve shape-based NMS methods.

3. METHOD

The basic idea behind ARM-NMS is quite intuitive: Use shapes instead of boxes to filter unnecessary detections. Shapes can align better with the underlying objects than just bounding boxes and lead to a more precise filtering process. However, comparing masks is a more computationally complex task than matching boxes. Hence, adjustments to the traditional NMS methods for an efficient workflow are required. Furthermore, we apply additional improvements to increase the filter performance. The pseudocode of our proposed algorithm can be seen in Figure 2.

3.1 ARM-NMS

ARM-NMS is computed on a list of detected instance masks $\mathcal{M} = \{\mathbf{m}_1, ..., \mathbf{m}_N\}$ with corresponding scores $\mathcal{S} = \{s_1, ..., s_N\}$ for a given scene. The first step is to adjust the scores after their relative size $\mathcal{A} = \{a_1, ..., a_N\}$. We hypothesize the following: larger detections are probably more accurate than smaller ones with the same confidence score. One of the reasons for this could be that larger detections are based on more pixels and consequently on more information. Ideally, area-rescoring should lead to the suppression of partial detections and preserve complete detections. In the beginning, the areas of the instance masks are calculated separately. Then, we compute the mean area $\overline{\mathcal{A}}$ of the instance masks. The area-rescoring function can be described as follows

$$s_i \leftarrow \frac{a_i s_i \frac{1}{w_a}}{\overline{\mathcal{A}}}, \tag{3}$$

where the parameter w_a controls the impact of the corresponding area a_i. As a result, detections with a larger area are assigned higher corresponding scores and smaller areas result in a score reduction. After the area-rescoring, we iterate through the list of detections in a descending order regarding their complementary scores. The detection $\mathbf{h}_m$ with the highest score s_{max} is removed from the list $\mathcal{M}$ and added to the list of final detections $\mathcal{F}$. Likewise, we utilize $\mathbf{h}_m$ to remove further detections from $\mathcal{M}$. Similar to box-based NMS approaches, we are looking for detections $\mathbf{m}_i$ which overlap with $\mathbf{h}_m$. Yet comparing shapes is a more comlex task, and therefore, we compare $\mathbf{h}_m$ only with detections $\mathbf{m}_i$ within a specified radius d_t. For these detections, we apply the Gaussian soft-NMS penalty function

Input: $\mathcal{M} = \{\mathbf{m}_1, ..., \mathbf{m}_N\}$, d_t, w_a
$\mathcal{S} = \{s_1, ..., s_N\}$, σ, n_t
$\mathcal{M}$ is the list of initial detections masks
$\mathcal{S}$ contains corresponding detection scores
w_a influences the area rescoring
d_t is the distance threshold
n_t is the NMS threshold for Greedy-NMS
σ is the Gaussian parameter for Soft-NMS

begin
  $\mathcal{F} \leftarrow \{\}$
  $\mathcal{A} \leftarrow area(\mathcal{M})$
  $\overline{\mathcal{A}} \leftarrow mean(\mathcal{A})$
  for s_i *in* $\mathcal{S}$ **do**
    $s_i \leftarrow f(s_i, a_i, \overline{\mathcal{A}}, w_a)$
  end
  Area-Rescoring
  while $\mathcal{M} \neq$ *empty* **do**
    $\mathbf{s}_{max} \leftarrow \text{argmax}\, S$
    $\mathbf{h}_m \leftarrow \mathbf{m}_{s_{max}}$
    $\mathcal{F} \leftarrow \mathcal{F} \bigcup \mathbf{h}_m$; $\mathcal{M} \leftarrow \mathcal{M} - \mathbf{h}_m$
    for $\boldsymbol{m}_i$ *in* $\mathcal{M}$ **do**
      if $distance(\boldsymbol{h}_m, \boldsymbol{m}_i) < d_t$ **then**
        if $iou(\boldsymbol{h}_m, \boldsymbol{m}_i) \geq n_t$ **then**
          $\mathcal{M} \leftarrow \mathcal{M} - \mathbf{m}_i$; $\mathcal{S} \leftarrow \mathcal{S} - s_i$
        end
        Greedy-NMS
        $s_i \leftarrow s_i f(iou(\mathbf{h}_m, \mathbf{m}_i, \sigma))$
        Soft-NMS
      end
    end
  end
  return $\mathcal{F}, \mathcal{S}$
end

Figure 2. **The Pseudocode** of our proposed approach is similar to greedy-NMS or soft-NMS. The main difference is utilizing instance masks $\mathcal{M}$ instead of bounding boxes $\mathcal{B}$. The pseudocode in the blue box exhibits additional modifications. Area-rescoring leads to favoring larger areas in order to select better-aligned detections. The distance function decreases the computational complexity.

to reduce the score of the detections with respect to their shape-based IoU. We repeat this step until $\mathcal{M}$ is empty. Finally, we return the final detection list $\mathcal{F}$ with their corresponding scores $\mathcal{S}$.

3.2 Implementation Details

Our proposed method is flexible and can be easily adjusted. We have implemented the following variations:

- **Shapes:** Two-dimensional object shapes can be described as a binary mask, a run-length encoded mask, or as a polygon. State-of-the-art instance segmentation pipelines usually output a binary mask in the size of the original image for every detected object. This can easily lead to a computational bottleneck for large-scale images with a high object density. Consequently, a compression method is needed. Merging all binary shapes into a single one-hot encoded map is not feasible due to overlapping masks. In

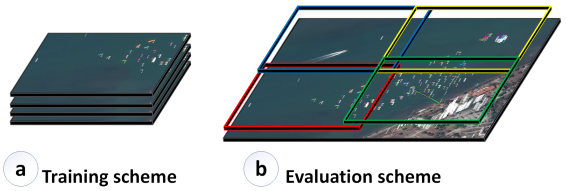

Figure 3. **Tiling Schemes.** We train our model on independent image patches, but for the evualuation process we produce detection lists on overlapping patches.

contrast, run-length encoded masks for every object can be an efficient approach to store and compare object shapes. For example, the MS COCO API (Lin et al., 2014) uses run-length encoding for evaluation purposes. Another procedure is to transform the masks into polygons. Polygons can be stored efficiently, are easier to visualize than run-length encoded masks, and can be comfortably implemented in other applications. As a result, we utilize polygons to store and calculate the overlap of object shapes.

- **Classes:** Our proposed method can be applied individually to each object category or jointly to all of them. Using our proposed approach jointly can improve detection accuracy if many double detections with different classification results exist.

- **Distance:** In order to reduce the computational complexity, we compare only objects if their distance is smaller than a certain radius. We use the Euclidean distance between the center points of the detections to calculate the distance. Other alternatives would be using only the first vertex of the polygon to save further computational steps or utilizing bounding boxes to determine proximity.

- **Overlap metric:** IoU is still the primary metric to compare the overlap between two shapes. IoU can be formulated as follows

$$\text{IoU} = \frac{\mathbf{m}_i \cap \mathbf{m}_j}{\mathbf{m}_i \cup \mathbf{m}_j}, \quad \mathbf{m}_{i,j} \in \mathcal{M}, \tag{4}$$

where $\mathbf{m}_i$ and $\mathbf{m}_j$ are two shapes, and we divide their intersection over their union. One of the most suitable characteristics of IoU is that it is scale-independent. This can lead to problems with partial detections. The IoU of a partial detection is usually low and may, therefore, be overlooked. For the case of an accumulation of partial detections, we implement an additional metric: IoMin. IoMin can be described as

$$\text{IoMin} = \frac{\mathbf{m}_i \cap \mathbf{m}_j}{min(a_i, a_j)}, \quad \mathbf{m}_{i,j} \in \mathcal{M}, \; a_{i,j} \in \mathcal{A}. \tag{5}$$

The difference between IoU and IoMin is that we divide the intersection of both shapes by the area of the smaller shape instead of the union.

4. EXPERIMENTS

In order to demonstrate the effectiveness of our approach, we apply Mask R-CNN on the dataset iSaid to get lists of detections for large-scale images. We use these lists of detections to evaluate our method and compare them to box-based greedy-NMS and soft-NMS. For the evaluation, we use a considerable parameter grid. Furthermore, we show in the ablation study that our improvements to Mask-NMS are reasonable.

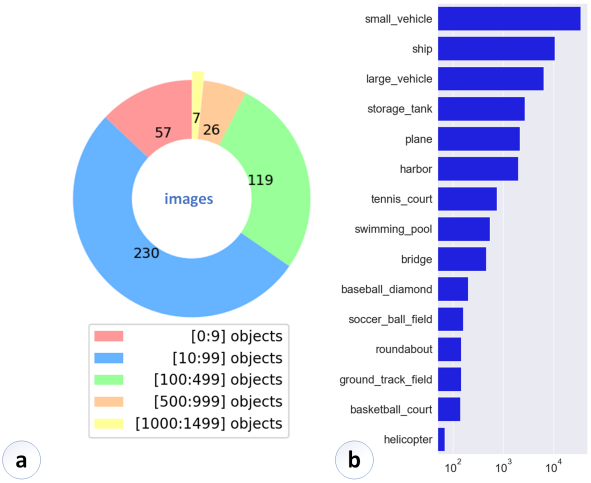

Figure 4. **Density and class distribution** of the evaluation dataset. From the iSaid validation dataset we select all images with less than 1500 objects for evaluation. (a) In this chart, we show the amount of images for the given object count intervals. (b) This barplot exhibits the class distribution of the evaluation dataset on a logarithmic scale.

4.1 Dataset

The iSaid (Waqas Zamir et al., 2019) dataset is a large-scale aerial image dataset and is derived from the DOTA (Xia et al., 2018) dataset. It appends to the DOTA dataset object shape informations. iSaid contains 2,806 high-resolution images collected from multiple sensors and platforms, including 655,451 object instances of 15 categories. The object categories vary from mobile categories like "ship' or "car" to static categories like "storage tank" or "bridge". Every category consists of a large number of instances. Furthermore, the dataset exhibits a high object scale variation. For example, the ship class ranges from small boats to large aircraft carriers. The distribution of the objects in a given scene is often imbalanced and uneven to represent real-life conditions. Due to performance issues of the COCO API with a large number of objects, we select only images up to a maximum of 1500 ground-truth objects, which are 439 of 458 images from the iSaid validation dataset. In Figure 4, we display the object density and class distribution.

4.2 Instance Segmentation

Detection lists with shape information are required to comprehensively evaluate our proposed method. For this, we utilize the popular instance segmentation method Mask R-CNN (He et al., 2017). For training the network, we crop the iSaid images into patches with a resolution of 800×800. We split the original iSaid training dataset randomly into a training dataset and a validation dataset. 95% of the images are used for training, the rest for validation. We train Mask R-CNN on all 15 categories of iSaid on two Nvidia V100 graphic processing units without pretraining except for the backbone. As a backbone, we use an Imagenet (Russakovsky et al., 2015) pre-trained Resnet101 (He et al., 2016) to extract features for further processing steps. We utilize the default anchor configuration following (He

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-V-2-2022-291-2022 | © Author(s) 2022. CC BY 4.0 License.

Table 1. Evaluation parameter grid.

method	[shape-based, box-based]	
penalty function	soft-NMS	greedy-NMS
IoU threshold	-	[0.01, 0.25, 0.5, 0.75]
sigma	[0.1, 0.2, 0.3]	-
score threshold	[0.01, 0.1, 0.2, 0.5]	
unique labels	[True, False]	
overlap metric	[IoU, IoMin]	
area-rescoring	[True, False]	

Table 2. ARM-NMS ablation results. We present the best mean average precision results under the defined settings.

metric	segmentation			
unique labels	True	True	True	False
overlap metric	IoU	IoMin	IoU	IoU
area-rescoring	True	True	False	True
AP [0.50:0.95]	**28.6**	28.2	25.9	27.6
AP 0.50	**53.5**	52.7	50.7	52
AP 0.75	**26.9**	26.5	23.2	25.8
AP[0.50:0.95] small	**18.9**	18.9	17.2	18.3
AP [0.50:0.95] medium	**32.0**	31.8	28.9	30.9
AP [0.50:0.95] large	**35.6**	35.0	33.8	34.4
AR [0.50:0.95] @100	**32.1**	31.1	31.7	31.4
AR [0.50:0.95] @1000	**37.0**	35.7	36.8	36.2
AR [0.50:0.95] @1500	**37.1**	35.7	36.8	36.2
AR [0.50:0.95] small	**26.9**	25.8	26.8	26.4
AR [0.50:0.95] medium	**38.0**	37.2	37.8	36.9
AR [0.50:0.95] large	**43.8**	42.2	43.4	42.8

et al., 2017). Nevertheless, we modify hyperparameters that limit the amount of maximum detections per image. Hence, we significantly increase the number of possible detections per image to 1000. During training, we start with a learning rate of $\alpha = 3 \cdot 10^{-3}$ and uniformly reduce α to 10^{-3} over 50 epochs. We choose a batch size of sixteen and utilize the AdamW optimizer (You et al., 2019) with a weight decay of 10^{-4}.

4.3 Evaluation Details

To demonstrate the effectiveness of our method, we compare our approach to the widely-used greedy-NMS and soft-NMS in large-scale images. We perform the experiments on the validation dataset of iSaid. Instead of evaluating our method on independent cropped images, we test on the full images. One example is given in Figure 3. Consequently, we utilize the object detector in a sliding-window approach with a kernel size of 800×800 pixels and a stride of 600 pixels in each direction. The resulting overlap of 200 pixels is quite large, but it ensures that no object is overlooked or divided into two parts because it lies on the border between two tiles. The produced binary masks are transformed into polygons, and their local vertices are adjusted to their location in the complete image. As a comparison metric, we utilize the common metrics mean average precision and mean average recall, which are implemented in the COCO API (Lin et al., 2014).

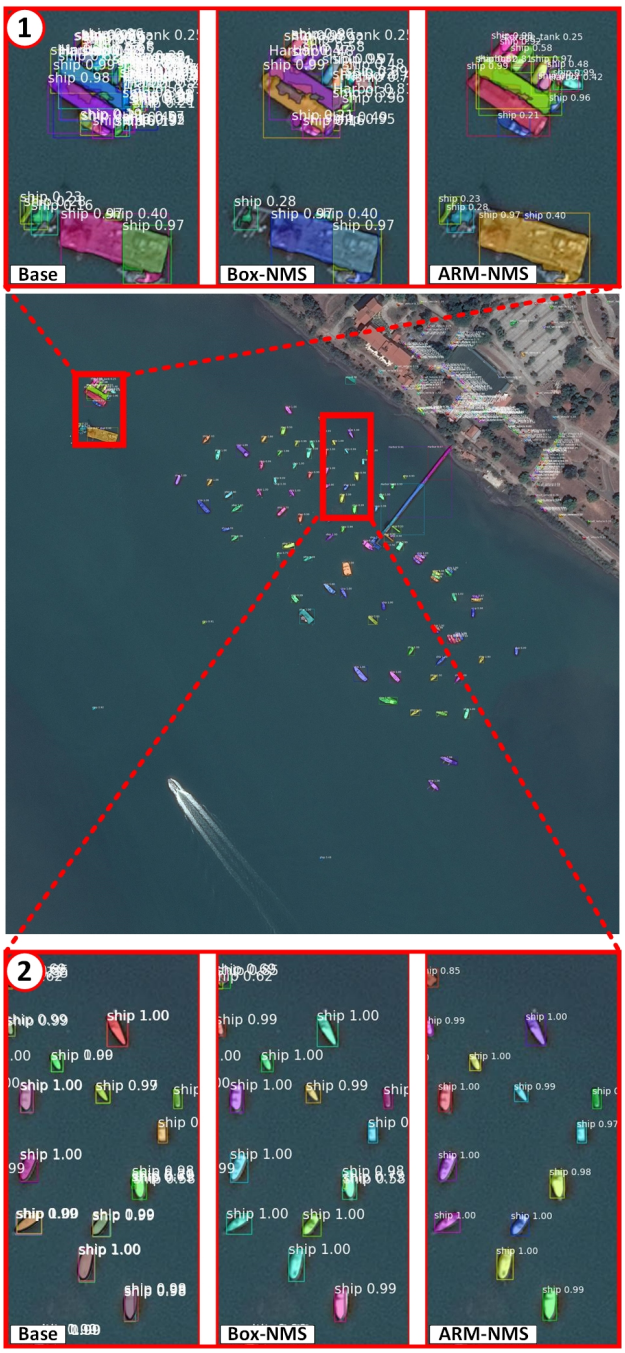

Figure 5. **Visualization results on the iSaid: coast.** The image sections show the unfiltered base detections, the box-based soft-NMS, and our proposed shape-based ARM-NMS results. (1) The annotation density in this clustered image section can be used to indicate the detection density. Box-NMS can reduce the number of detections, but ARM-NMS outperforms its counterpart significantly. (2) We display a relatively sparse scene with double detections among the vessels.

4.4 Parameter Grid

Non-maximum suppression methods are parameter-sensitive approaches. Table 1 shows the used parameters and the performed method variations. Fields marked with an "-" are not considered for the evaluation. We implement greedy-NMS and soft-Nms box-based and also integrate them into our shape-based ARM-

Table 3. NMS Results on the iSaid dataset. We compare our proposed shape-based ARM-NMS with box-based NMS.

metric	segmentation						box					
method	none		ARM-NMS		Box-NMS		none		ARM-NMS		Box-NMS	
penalty function	-	-	soft	greedy	soft	greedy	-	-	soft	greedy	soft	greedy
overlapping tiles	False	True	True	True	True	True	False	True	True	True	True	True
iou threshold	-	-	-	0.5	-	0.5	-	-	-	0.25	0.1	0.5
sigma	-	-	0.1	-	0.1	-	-	-	0.1	-	-	-
score threshold	-	-	0.01	0.01	0.01	0.5	-	-	0.01	0.01	0.01	0.5
unique labels	-	-	True	True	True	True	-	-	True	True	True	True
overlap metric	-	-	IoU	IoMin	-	-	-	-	IoU	IoU	-	-
area-rescoring	-	-	True	True	-	-	-	-	True	True	-	-
AP [0.50:0.95]	24.6	20.7	**28.6**	28.2	25.3	25.2	28.4	24.2	**32.9**	32.4	29.4	29.3
AP 0.50	48.4	38.6	**53.5**	52.5	48.5	49	52.1	41.8	**57.1**	56	52.5	52.8
AP 0.75	22.2	19.3	**26.9**	26.5	23.1	22.6	27.5	25.2	**34.0**	33.6	29.6	29.1
AP[0.50:0.95] small	16.1	13.5	**18.9**	18.8	16.6	16.6	23	20.9	**25.7**	25.2	23.6	23.6
AP [0.50:0.95] medium	28.3	22.6	**32.0**	31.7	28.2	28.2	30.7	24.7	**35.6**	35.2	31.4	31.3
AP [0.50:0.95] large	31.5	28.5	**35.6**	34.9	33.5	33.6	29.5	25.0	**37.0**	36.5	32.7	32.6
AR [0.50:0.95] @100	30.0	31.3	**32.1**	31.1	31.6	31.4	33.4	35.2	**36.0**	35.1	35.3	35.0
AR [0.50:0.95] @1000	34.9	**37.9**	37.0	35.7	36.8	36.5	39.4	**43.5**	42.2	41.0	41.8	41.4
AR [0.50:0.95] @1500	35.0	**38.1**	37.1	35.7	36.9	36.6	39.5	**43.6**	42.2	41.1	41.8	41.5
AR [0.50:0.95] small	25.9	**27.7**	26.9	25.8	26.8	26.8	31.5	**33.7**	32.7	31.4	32.5	32.5
AR [0.50:0.95] medium	36.1	**38.7**	38.0	37.2	37.7	37.4	39.3	**43.7**	42.5	41.7	42.2	41.9
AR [0.50:0.95] large	39.9	**44.7**	43.8	42.1	43.4	43	42.6	**49.6**	48.1	46.8	47.4	46.9

NMS. For greedy-NMS, we apply different IoU thresholds n_t, which range from 0.001 to 0.75. In the case of soft-NMS, we use a σ of 0.1, 0.2, and 0.3. Additionally, we apply different score thresholds, ranging from 0.01 to 0.5. Furthermore, we perform further variations for the shape-based approaches. We filter each class separately and jointly. Additionally, we utilize IoU and IoMin as overlap metric. Finally, we examine the effectiveness of the proposed area-rescoring.

5. RESULTS

In this section, we evaluate the effectiveness of our shape-based NMS. First, we show the quantitative results of the box-based and the shape-based ARM-NMS approaches. Then, we discuss some qualitative results. Finally, we display the ablation study.

5.1 Quantative Results

In Table 3, we compare shape-based and box-based NMS approaches on detection lists produced by Mask R-CNN on the iSaid validation dataset. Fields with an "-" are considered irrelevant for the corresponding method. We apply the COCO-style mean average precision (mAP) and mean average recall (mAR) for segmentations and boxes. The unfiltered detection results are labeled "none" under the method row and are used as a baseline. For each case, segmentations and boxes, the average precision is generally higher for the unfiltered detections without overlapping tiles. In contrast, for the non-overlapping tiles, the detections indicate a higher average recall. This is reasonable because overlapping tiles lead to double detections, but they reduce the possibility of missing detections on the border of the tiles. On average, the mask-shaped approaches are 2.8 times slower than the box-shaped ones. The results seen here are selected from experiments on a comprehensive parameter grid and for each method. We present the best results regarding overall mean average precision. Regarding the metric for the segmentations, we can see that our proposed shape-based approach with a Gaussian soft-max penalty function achieves not only a higher average precision than the unfiltered detections but can beat the box-based alternative by 3.3 points. The implementation with the soft-NMS performs better in all metrics than the greedy-NMS variant. Our approach has a slightly lower average recall for a max detection limit of 1500 instances than the unfiltered detections with overlapping tiles, but obtains a higher average recall than those without overlapping tiles. However, the average precision improves by 7.9 points compared to the unfiltered detection with overlapping tiles and by 4.0 points without overlaps. Furthermore, ARM-NMS accomplishes a higher average recall than the box-based NMS methods. The best combination of parameters and variants for our method is using a Gaussian penalty soft-NMS function, detections with overlapping tiles, a sigma of 0.1, a score threshold of 0.01, filtering for each label separately, IoU as a comparison metric and with area-rescoring. A similar result can be seen for the box metrics. Our method accomplish the highest mean average precision while reaching, except for the unfiltered overlapping baseline, the highest mean average recall score.

5.2 Qualitative Results

We show a few qualitative results in Figures 5, 6 and 7. The detection results on entire images are filtered by ARM-NMS. Furthermore, we illustrate some image sections in more detail and display the unfiltered detections, detections filtered by ARM-NMS, and detections filtered by a box-shaped method. The settings of the NMS methods are based on the best results from Table 3. In Figure 5 we show a coastal scene with many vessels. In the image section one, we can see how ARM-NMS outperforms Box-NMS significantly in a cluster of object detections. Likewise, as seen in image section two, ARM-NMS handles sparse double detections well. Figure 6 displays an airplane graveyard. Airplanes are concave objects with a high

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-V-2-2022-291-2022 | © Author(s) 2022. CC BY 4.0 License.

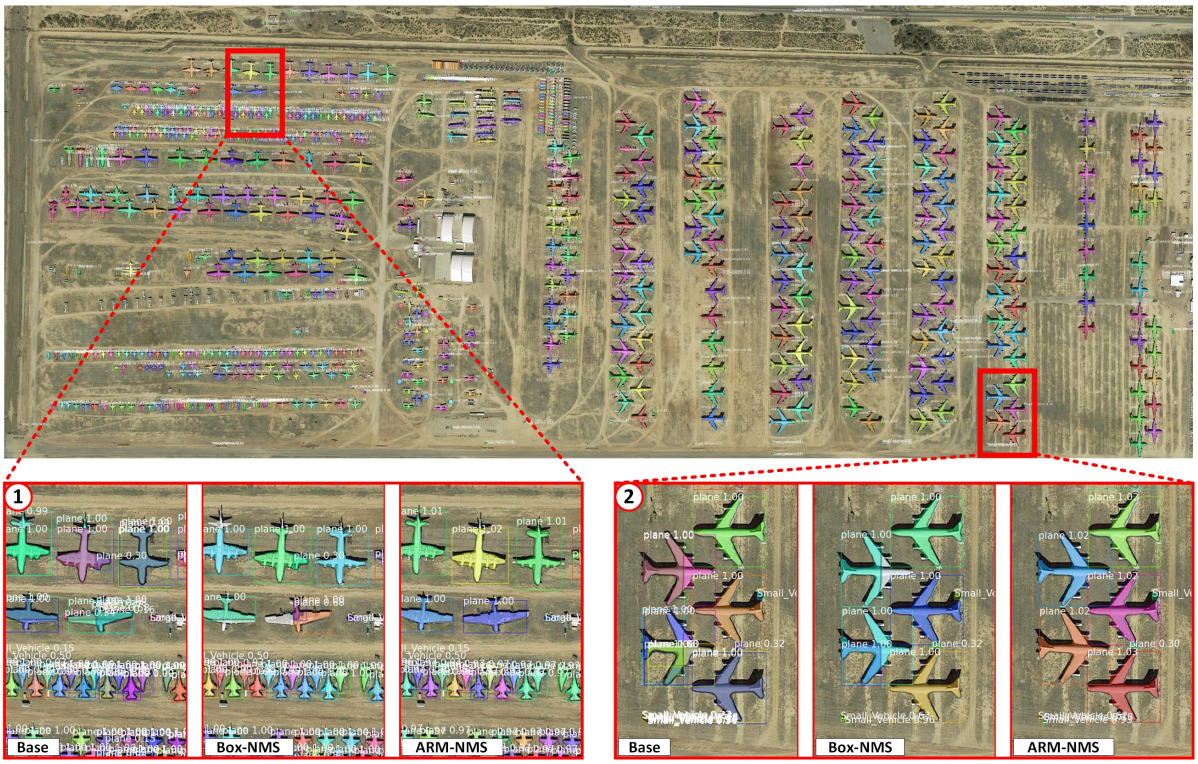

Figure 6. **Visualization results on the iSaid: airplane graveyard.** The image subsets show the unfiltered box-based soft-NMS, and our proposed shape-based ARM-NMS results: (1) Example for multiple detections. (2) Example for the intersection of two overlapping image patches.

Figure 7. **Visualization results on the iSaid: parking spot.** The diagonally-aligned buses in the image section are a challenge for box-based NMS approaches. In contrast to ARM-NMS, Box-NMS eliminates true-positive detections due to overlapping boxes.

overlap potential for bounding boxes. Image section one shows how ARM-NMS excels in partial detections. The box-based approach performs worse in this scene. This is also the case for the second image section. The narrow concave objects cause an overlap of bounding boxes despite the underlying objects not actually overlapping. Furthermore, this image section is part of the intersection of two overlapping image patches. Therefore, high-scored partial detections are present. ARM-NMS excels in this case, while Box-NMS falls behind. In the last example, Figure 7 shows a parking spot. The difficulty in this scene is the diagonally-aligned buses. Diagonally-aligned objects can lead to a high overlap of bounding boxes similar to concave objects. We observe that ARM-NMS outperforms Box-NMS in this scenario too.

5.3 Ablations

In order to determine the best composition for our approach and compare it to Mask-NMS, we evaluate the impact of the different variations on ARM-NMS. ARM-NMS with no area-rescoring, unique labels and IoU is idententical to Mask-NMS. Table 2 shows the best results on iSaid for a specific parameter combination regarding the COCO-style mAP metric.

- **Labels:** In the default case of NMS, each class is filtered separately. Therefore, double detections with different labels are ignored in the filtering process. Filtering all classes simultaneously may lead to better results in certain cases, but overall it decreases the average precision and the average recall score. Furthermore, it leads to an increased conputing time due to the increase of objects compared to each other.

- **Overlap:** IoU is an integral part of most of the NMS methods. In this case, where the compared detections differ strongly in size, NMS does not suppress any detections. IoMin may perform better in some cases than IoU, but it slightly decreases average precision.

- **Area-rescoring:** Area-rescoring adjusts the detection scores favoring larger areas. Ideally, partial detections should receive a drop in their score and thus, are less likely to suppress better-aligned detections. However, reducing the

scores of all small objects by a large margin can lead to missing final detections. In general, area-rescoring leads to a significant increase in average precision in our experiments. A possible reason for this observation could be the use of the applied tiling scheme. Partial detections on the edges of a patch could still reach a high detection score and, therefore, suppress the overlapping patch's better-aligned detections. The partial detections are smaller than their better-aligned counterparts and, therefore, receive a higher penalty from area-rescoring. Consequently, the chance that partial detections suppress better-aligned detections is reduced.

6. CONCLUSION

In this paper, we propose ARM-NMS. ARM-NMS utilizes shapes in order to filter unnecessary object detections. It does not require any retraining and, thus, can be easily implemented in existing instance segmentation methods. To demonstrate the effectiveness of our method, we created detection lists by the popular Mask R-CNN detector applied to entire images of the iSaid validation dataset. ARM-NMS outperforms box-shaped filtering algorithms by more than three points on the COCO-style mAP metric. Furthermore, we confirmed our hypothesis that rescoring detections based on the shape and area of the objects leads to an improvement in detection performance. Future research will address the improvement of the final detections by analyzing different approaches for merging overlapping detections.

References

Dalal, N., Triggs, B., 2005. Histograms of oriented gradients for human detection. *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, 1, Ieee, 886–893.

He, K., Gkioxari, G., Dollár, P., Girshick, R., 2017. Mask r-cnn. *Proceedings of the IEEE international conference on computer vision*, 2961–2969.

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

He, Y., Zhang, X., Savvides, M., Kitani, K., 2018. Softer-nms: Rethinking bounding box regression for accurate object detection. *arXiv preprint arXiv:1809.08545*, 2(3).

Jiang, B., Luo, R., Mao, J., Xiao, T., Jiang, Y., 2018. Acquisition of localization confidence for accurate object detection. *Proceedings of the European conference on computer vision (ECCV)*, 784–799.

Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S., 2017. Feature pyramid networks for object detection. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2117–2125.

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

Liu, S., Huang, D., Wang, Y., 2019. Adaptive nms: Refining pedestrian detection in a crowd. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6459–6468.

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B., 2021. Swin transformer: Hierarchical vision transformer using shifted windows. *arXiv preprint arXiv:2103.14030*.

Qiao, S., Chen, L.-C., Yuille, A., 2021. Detectors: Detecting objects with recursive feature pyramid and switchable atrous convolution. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10213–10224.

Redmon, J., Divvala, S., Girshick, R., Farhadi, A., 2016. You only look once: Unified, real-time object detection. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 779–788.

Ren, S., He, K., Girshick, R., Sun, J., 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 91–99.

Rosenfeld, A., Thurston, M., 1971. Edge and curve detection for visual scene analysis. *IEEE Transactions on computers*, 100(5), 562–569.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M. et al., 2015. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3), 211–252.

Solovyev, R., Wang, W., Gabruseva, T., 2021. Weighted boxes fusion: Ensembling boxes from different object detection models. *Image and Vision Computing*, 1-6.

Tian, Z., Shen, C., Chen, H., 2020. Conditional convolutions for instance segmentation. *European Conference on Computer Vision*, Springer, 282–298.

Wang, X., Kong, T., Shen, C., Jiang, Y., Li, L., 2020a. Solo: Segmenting objects by locations. *European Conference on Computer Vision*, Springer, 649–665.

Wang, X., Zhang, R., Kong, T., Li, L., Shen, C., 2020b. Solov2: Dynamic and fast instance segmentation. *Advances in Neural information processing systems*, 33, 17721–17732.

Waqas Zamir, S., Arora, A., Gupta, A., Khan, S., Sun, G., Shahbaz Khan, F., Zhu, F., Shao, L., Xia, G.-S., Bai, X., 2019. isaid: A large-scale dataset for instance segmentation in aerial images. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 28–37.

Xia, G.-S., Bai, X., Ding, J., Zhu, Z., Belongie, S., Luo, J., Datcu, M., Pelillo, M., Zhang, L., 2018. Dota: A large-scale dataset for object detection in aerial images. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3974–3983.

You, Y., Li, J., Reddi, S., Hseu, J., Kumar, S., Bhojanapalli, S., Song, X., Demmel, J., Keutzer, K., Hsieh, C.-J., 2019. Large batch optimization for deep learning: Training bert in 76 minutes. *arXiv preprint arXiv:1904.00962*.

Zheng, Z., Wang, P., Ren, D., Liu, W., Ye, R., Hu, Q., Zuo, W., 2021. Enhancing Geometric Factors in Model Learning and Inference for Object Detection and Instance Segmentation.