FlagDetSeg: Multi-Nation Flag Detection and Segmentation in the Wild

Shou-Fang Wu
National Central University
Taiwan
stefan88717@g.ncu.edu.tw

Ming-Ching Chang
University at Albany - SUNY
Albany, NY, USA
mchang2@albany.edu

Siwei Lyu
University at Buffalo - SUNY
Buffalo, NY, USA
siweilyu@buffalo.edu

Cheng-Shih Wong
Academic Sinica, Taipei, Taiwan
r04945028@ntu.edu.tw

Abhineet Kumar Pandey
University at Albany - SUNY
Albany, NY, USA
apandey@albany.edu

Po-Chi Su
National Central University
Taiwan
pochyisu@gmail.com

Abstract

We present a simple and effective flag detection approach for multi-nation flag instance segmentation in-the-wild based on data augmentation and Mask-RCNN PointRend. To the best of our knowledge, this is the first multi-nation flag detection work incorporating recent deep object detection with code and dataset that will be released for public use. Flag images with binary segmentation are collected from public domain including the Open Image V6 and annotated for up to 225 countries. Additional flag images are generated from template flag images with cropping, warping, masking, and color adaption to hallucinate realistic-looking flag images for training and testing. Data augmentation is performed by fusing and transforming the segmented flags on top of natural image backgrounds to synthesize new images. To cope with the large variability of flags with the lack of authentic annotated flags, we combine the trained binary Mask-RCNN segmentation weights with the new multi-nation classifier for fine-tuning. For evaluation, the proposed model is compared with other popular detectors and instance segmentation methods including YOLACT++. Results show the efficacy of the proposed approach.

Keywords: flag detection, multi-nation, instance segmentation, Mask-RCNN, data augmentation, synthetic image generation, dataset, fine-tuning.

1. Introduction

With the rise of deep learning, visual detection and recognition using convolutional neural networks (CNN) have wide range of applications. We investigate and apply the latest CNN developments to the problem of flag detection and identification. We aim to develop a multi-nation flag detection and segmentation system that can localize and recognize flags *in the wild* for *real-time* real-world application use. See Figs. 1 and 2 for overview and example.

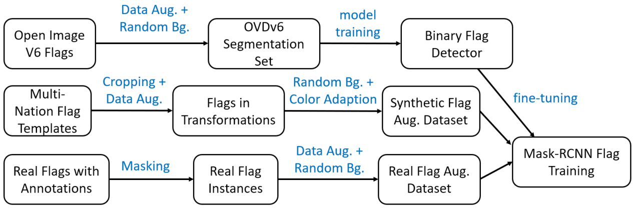

Figure 1. **Overview** of the aggregation, annotation, hallucination, and augmentation of our multi-nation flag dataset. Our flag detector is trained with multiple stages and fine-tuning. Our approach avoids reliance on excessive manual annotations.

Flag detection and identification in-the-wild have many important applications, including news and social media content understanding, scene understanding for tourism (traveling apps), and military use. The problem is challenging due to multiple factors: (1) Flags are non-rigid and can deform arbitrarily in tilted, rotated, skewed views under large aspect ratio and illumination changes. (2) Flags can appear in arbitrary scales and under heavy occlusion when hung. (3) There are hundreds of country flags and thousands of flags belonging to various business organizations. Objects such as balloons and banners that look like flags or flags appearing in reflected surfaces should not be confused by the detector. Thus, the intra-class variations are typically larger than inter-class variations. (4) The lack of a large annotated dataset and the labor-intensive nature in producing such training set hinder naive machine learning approaches. Meanwhile, flag detectors trained using only synthetic flag images does not perform well on nature images due to the large environmental variabilities and domain gaps. (5) The requirements of real-time execution under large viewpoint

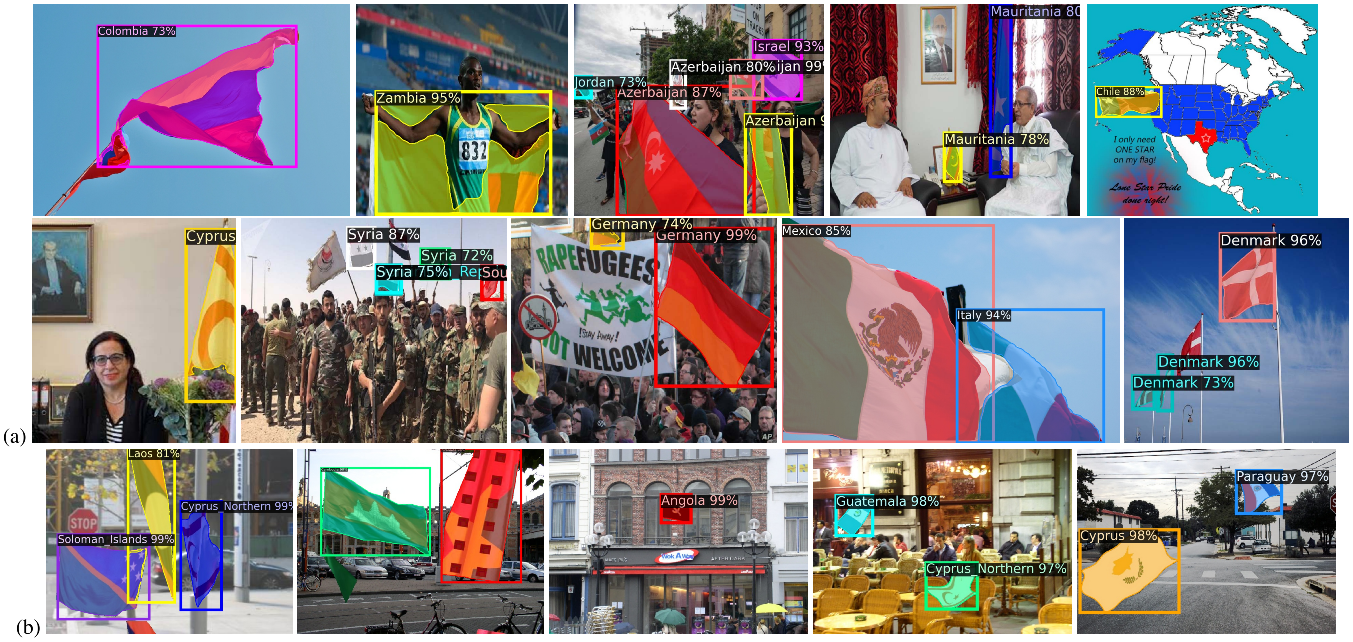

Figure 2. Examples of the proposed multi-nation flag detection and segmentation running on (a) real-world and (b) synthetic images.

and scale changes further complicate the problem.

Existing works on visual flag detection are very limited, and to the best of our knowledge, none of them address the need of flag detection in the wild (see complete survey in § 2). The country flag detector in [1] is the closest to our work. Their method is based on a VGG16 FCN that generates flag region proposals. Flags are then detected as bounding boxes without segmentation using hand-crafted color features. Only synthetic flag templates are used for training, and the training pipeline is not end-to-end. Only simple test cases are shown where the flags appear large at image center. Although deep visual object detectors have improved greatly in recent years [2], standard object detectors (*e.g.* YOLOv4 [3]) or instance segmentation (*e.g.* Mask-RCNN [4], YOLACT++ [5]) still cannot be directly applied to achieve a plausible solution of flag detection, due to the lack of a suitable dataset and the labor intensive nature of producing such dataset.

In this paper we develop a simple effective system for multi-nation flag detection with instance segmentation that can run in real tie in-the-wild. We adopt the widely-used Mark-RCNN [4] for flag detection and instance segmentation, and we further improve its performance by fine-tuning with datasets and integration with PointRend [6]. For constructing a proper flag image training set, our approach is based on effective data *aggregation*, *annotation*, *hallucination*, and *augmentation*. (1) We combine the flag segmentation of real-world images from the Open Image V6 dataset [1] (aggregation). (2) We also collect a new dataset by google search for flag images of specified countries and annotate flag contours (annotation). (3) We further generate synthetic flag images with transformations based on the template flags of 225 countries (hallucination). (4) Finally, we adopt various copy/paste and other data augmentation methods to create a large enough dataset for training (augmentation). We show that simple copy/paste of the masked flag pixels and superimposing on top of nature image backgrounds can effectively boost performance [7].

Contribution of this paper includes the following:

- The proposed **FlagDetSeg** can detect and identify flags among 200+ countries with precise segmentation from an image. Multiple (up to tens of) flags in an image can be detected in-the-wild at about 5 FPS on a standard GPU computer. To our knowledge, this is the first adoption of state-of-the-art visual object detector as a proper flag detection and identification solution.
- The proposed data *aggregation*, *annotation*, *hallucination*, and *augmentation* for generating a sufficiently large dataset for model training is simple and effective. With copy/paste, image cropping, warping, and background harmonization on natural images, our hallucination module can produce a rich, multi-nation synthetic flag dataset for training and evaluation.
- The newly constructed **FlagDetSeg Dataset** consists of both real-world and synthetic flag images after sufficient data augmentation. Both the code and the datasets are available on GitHub [2].

[1] https://g.co/dataset/open-images

[2] https://github.com/sfstefanwu/FlagDetSeg.git

- Extensive evaluation is performed on comparing the proposed Mask-RCNN with various backbones, fine-tuning, and PointRend [6] variants against the YOLACT++ instance segmentation. Our FlagDetSeg network achieves AP_{50} of 90.24 and 93.1 on our flag instance segmentation test set, for binary and multi-nation flag instance segmentation, respectively.

2. Background

Object detection is one of the most fundamental and challenging problems in deep learning and computer vision; see survey in [2]. Deep object detectors can be organized into two main categories: (a) *two-stage* detectors *e.g.*, Faster-RCNN [8] and Mask-RCNN [4], where object proposals are generated via a Region Proposal Network (RPN) for the subsequent subnet to perform classification, and (b) *one-stage* detectors such as SSD [9] and YOLO methods [3, 10, 11].

Instance segmentation [12] is an extension of object detection, where a pixelwise object mask is produced in addition to the detected bounding box of each object. In other words, both object detection and semantic segmentation are simultaneously solved. Mask-RCNN [4] is probably the first and most well-known method of the kind, which is based on an extension of Faster-RCNN. With an identical first stage of the region proposal network (RPN), in the second stage in parallel to predicting the class object box offset, a segmentation branch is incorporated to output the binary mask for each object RoI. This new branch is a Fully Convolutional Network (FCN) [13] on top of a CNN feature map. To avoid the misalignment caused by the original RoI pooling layer, a RoIAlign layer preserves the pixel-level spatial correspondence. With a backbone of ResNeXt101-FPN [14], Mask-RCNN achieved top results in the COCO benchmark. Mask-RCNN is simple to train, well-generalizable, and with only a small overhead to Faster-RCNN that runs at about 5 FPS. Recent variants and improvements of Mask-RCNN include Mask Scoring R-CNN [15] and TensorMask [16]. YOLACT [17] is a fast instance segmentation method based on fully-conv topology, where image segmentation is preformed by 2 parallel subtasks of prototype mask generation and mask coefficient prediction. The improved YOLACT++ [5] incorporates deformable convolution into the backbone to provide flexible feature sampling and strengthen instances with different scales, rotations, and aspect ratios.

Flag identification. Early works of flag recognition simplify the problem assumption by focusing on classifying the flag type from a given image. In [18], flags are identified based on the HSV color texture analysis and gradient features. In [19, 20], flags are identified based on color features and a fuzzy-neural algorithm respectively with a kNN classifier. In [21], flags are recognized via HOG and HSV color features via a Adaboost cascade classifier. In [22], a 5-layer CNN extracts features to recognize national flags.

Flag detection. Deep CNN features are first used in [23], where flags are matched in 3 fixed scales of sliding windows. Their network is trained on a dataset containing flags of different scales, styles, deformations and lighting conditions. However, their flag detection performance is far behind recent RPN-based detectors such as Faster-RCNN or Mask-RCNN. The country flag detector in [1] operates based on a local context network and Color-BRIEF features. The VGG16 FCN generates region proposals that can capture flag deformations. This method can only handle simple cases where the flags appear large enough in the image. It does not work well against large appearance variations or warping between the learned template flags and flags appearing in real-world images.

The following flag detectors are available on-line without paper publication. The Flagnet [24] is a YOLO-based network coming with a country flag dataset of 193 United Nations member countries. However in this dataset, flags appear mostly dominant at image centers, and there is no segmentation available. Flagnet thus cannot deal with flag deformations, scale variations or complex backgrounds. The SSD-based country flag detector in [25] can only detect 25 types of rectified “flag cards” in the images.

Data augmentation [26] is a straightforward but crucial step in training deep neural networks with many practical considerations. Standard tools such as `imgaug` [27] and `Albumentations` [28] can enlarge the training set by several tens in size. A simple copy/paste data augmentation can be surprisingly effective as shown in [7] especially in our case where the flag foreground segmentation is available. Data generation using GAN [29] has growing popularity to various applications where training samples are limited and self-supervised or semi-supervised approaches are preferred.

3. Method

We aim to develop a multi-nation flag detector with segmentation capability. We first construct a flag instance segmentation dataset. We start with aggregation existing flag datasets, and we also gather and annotation our own multi-nation flag dataset. Data generation and augmentation are particularly useful in developing our flag detector, as we can effectively transform flag foregrounds and paste onto nature image backgrounds to create a large number of realistic looking samples. We next elaborate on our approach in details.

§ 3.1 describes our collection and aggregation of existing flag datasets. § 3.2 describes our own efforts in creating a new multi-nation flag dataset by collecting and annotating flag segmentation and country labels on real-world images. § 3.3 describes how we generate realistic-looking synthetic flag images from the raw country flag templates; these synthetic images are warped and superimposed onto nature image backgrounds to create more samples with variabilities. § 3.4 describes our image data augmentation to enlarge both real and synthetic flag samples for training and testing. § 3.5 describes our flag detection and segmentation network based on Mask-RCNN with fine-tuning and PointRend.

3.1. Flag Dataset Aggregation

We start with aggregating the existing image datasets containing flag annotations. We try to combine and enrich available datasets. However, existing image datasets involving flags are either non-public (those surveyed in § 2), not specific for flag detection use, no segmentation available, or not containing realistic flags. The closest we found is the Google Open Images Dataset V6 (OIDv6) [30], which contains 5000+ flag images with precise flag contour segmentation.

Since there is no annotation on the country type of each flag in OIDv6, this dataset is only sufficient to train a *binary* flag detector. Also flags in OIDv6 is very unbalanced and do not cover many countries all over the world; for example the US flags appear most frequently in OIDv6. The Open Image flag annotations may contain minor errors such as missed or inconsistent labeled cases. We found that most flags come with segmentation annotations, but some are with only bounding boxes. We thus omit flags that are too small or without segmentation masks and do not use them in the precision/recall evaluation. We end up with $5,233$ flag images with segmentation annotations from OIDv6, and we name it the **FlagDetSeg OIDv6 Segmentation Set**. This dataset is useful for detecting flag objects and distinguishing flags from similar objects such as balloons or banners.

3.2. Multi-nation Flag Dataset Annotation

In order to create a multi-nation dataset to train our desired flag detector with both country and segmentation annotations, we next (1) use google search to query for the country name, (2) hand pick suitable images with the desired flags, and (3) perform flag contour annotations. Note that manual annotation of flags with segmentation contours is quite labor-intensive. We only annotate about **5** such images for each country. We essentially assume additional hallucination and augmentation techniques can generate a large enough dataset from these manual annotations; and results in § 4 support such research hypothesis. To maximize diversity, we carefully select flags of different poses, scales, and orientations for the 5 samples of each country.

We include country flags based on the Wikipedia list of 225 sovereign states [3] as in [18]. Most of this country flags belong to the UN Recognized Nations list (206), while a few are self-declared nations or dependent territories. The flag images were resized to 600×600, and we use the `makesense.ai` online tool [31] to perform annotation. There are a total of $1,120$ images and $2,207$ flag instances. These images are later augmented into $15,468$ samples using methods in § 3.4 for model training and testing. We name this multi-nation real-image flag dataset the **FlagDetSeg Real Image Set**.

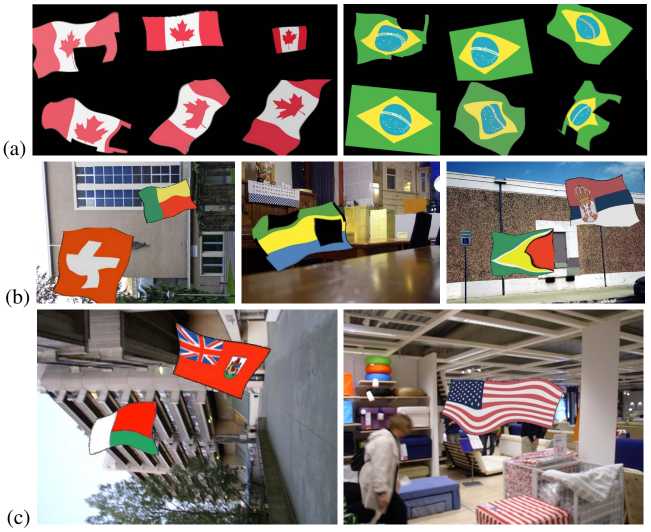

Figure 3. (a) Synthetic flag warping under different scales, cutting, and deformations. (b,c) The pasting of synthetic flags on top of nature image backgrounds with image harmonization.

3.3. Flag Image Hallucination

Another effective solution to address the lack of a sufficiently large real-image dataset is to hallucinate (generate) a synthetic flag database using standard flag template images. The advantage of such synthetic flag generation is obvious. Country flags are designed in standard templates, so we can easily collect ideal flag images (in perfect colors and upright views) and apply image deformations to simulate multiple variations including the shape, viewpoint, or lighting changes for the purpose of data augmentation. Another important advantage is that, with synthetic flag generation, we can control which country of flags to generate to directly produce samples for country flags that seldom appear in the real-image set. This can effectively balance the long-tail classes in the dataset. We obtained flag template images from *Dynamo Spanish* [4] for the 225 sovereign states listed in Wikipedia. Several 2D image transformations including elastic transform, distortion, cropping, HSV color jittering (with fixed hue), *etc.* are applied as shown in Fig. 3(a) to approximate the effect of a sheet-like piece of flag deformation in the 3D environment.

[3] https://en.wikipedia.org/wiki/List_of_sovereign_states

[4] https://dynamospanish.com/flags/downloads/

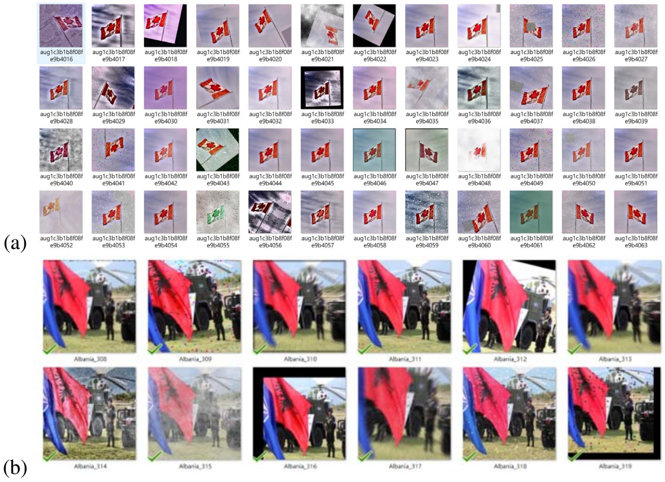

Figure 4. Flag transformations using (a) `imgaug` [27] and (b) `Albumentations` [28] for image augmentation.

We next paste one or more of the “hallucinated” (or generated) flags onto natural image backgrounds to enrich the variability of the synthetic images. We avoid any overlapping of pasted flags during the operations. Such copy/paste data augmentation is effective and is backed up by a recent paper [7]. Note that our work is independent of their paper, but with the sample simple thought for data augmentation. Poisson image editing [32] or recent image harmonization methods [33] can then be applied to improve the foreground/background compatibility of such composited images. To obtain the background images, we collected $1,466$ flag-free nature images from three sources: (1) the *Human Made Scene Collection* [5], (2) *Stanford Background Dataset* [6], and (3) pictures of natural scene collected in-house. We double-checked them to ensure the absence of flags that will otherwise cause error in annotations.

As a result, we obtained $23,785$ synthetic flag images in 225 country classes, where there are about 130 to 150 synthetic samples for each country. We name this dataset the **FlagDetSeg Synthetic Image Set**. A few samples of it are shown in Fig. 3(b,c).

3.4. Flag Image Augmentation

We use popular image augmentation tools including `imgaug` [27] and `Albumentations` [28] to augment both our real and synthetic flag images to obtain a sufficiently large and balanced dataset.

We augment the **OIDv6 Segmentation Set** by applying all available image transformation methods in `imgaug` to enrich the diversity of samples. These include (1) *geometric transformations*: flipping, cropping, scaling, perspective and elastic transformation, and translation, (2) *pixel transformations*: RGB channel shuffling, HSV color jittering, histogram equalization, grayscale and linear contrast adjustment, salt & pepper noise addition, region dropout, cropping and padding, and (3) *special effects*: Contrast Limited AHE, emboss, adding JPEG compression artifacts and other artifacts including rain, snowflake, spatter, cloud effects, ZoomBlur, and Fancy PCA. Fig. 4(a) shows sample results of these augmentations.

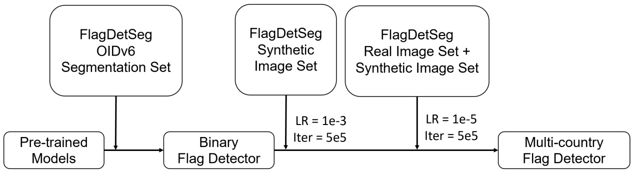

Figure 5. The proposed multi-nation flag detector training pipeline.

When augmenting the **Real Image Set** and **Synthetic Image Set** for multi-nation flag detection, we must restrain a certain transformations to ensure “safe” flag data augmentation [26], such that flags after augmentation still preserve the same country labels. For example the up-down-flipped Indonesia and Poland flags are nearly identical; so we must avoid image transformation that can possibly cause flag identity change. We avoid RGB-channel swapping or aggressive hue change in HSV color space. Specifically, the following `Albumentations` [28] transformations are randomly permuted and combined for augmentation: (1) *spatial transformations*: Dropout, ElasticTransform, GridDistortion, GridDropout, HorizontalFlip, OpticalDistortion, PiecewiseAffine, ShiftScaleRotate, (2) *pixel transformations*: ColorJitter, FancyPCA, ImageCompression, HueSaturationValue, MotionBlur. Fig. 4(b) shows sample results of these augmentations.

As a result, after copy/paste flag segmentation from real-world images to natural image backgrounds, we generated $45,620$ images of size 800×600. In total, we have generated 300k flag images and about 450k instance masks after combining both real and synthetic flag images after augmentation in our **FlagDetSeg Dataset**.

3.5. Mask-RCNN Flag Detection

The training of the proposed model consists of two steps. (1) We start with the ImageNet pre-trained Mask-RCNN [34] and re-train it on our *OVDv6 Segmentation Set* to obtain a strong binary flag detector with segmentation capability. (2) We then fine-tune the model with our *Real + Synthetic Image Set* to build the flag detector with multi-nation classification capability. The training is performed on a GPU machine with 3 GTX 1180 GPUs. Fig. 5 overviews our multi-nation flag detector training steps.

Training the Mask-RCNN binary flag detector. We used FPN as the network head to run experiments with ResNet-101 or ResNeXt-101 as the backbone. Each mini-

[5] http://natural-scenes.cps.utexas.edu/db.shtml

[6] http://dags.stanford.edu/projects/scenedataset.html

batch contains 3 images per GPU for ResNet and 2 for ResNeXt. Input image size is 550×550 (which was previously resized during data augmentation), and each image is with 256 sampled RoIs. The model is trained for $50,000$ iterations with the initial learning rate 0.001, which is decreased by one tenth at the $30,000$ iteration. For the experiment comparing with YOLACT++, we used ResNet101 and set the batch size to 15 (*i.e.* 5 images per GPU). The input image size is 550×550, which is the largest size limited by YOLACT++.

Fine-tuning the multi-nation flag detector. Once the performance of the above binary flag detector is ensured, we continue to fine-tune the network with multi-nation classification capability following a standard transfer learning paradigm. Expanding from the binary classifiers with a capability to distinguish the 200+ countries requires sufficient steps of training to learn a large set of weights and bias terms. In order to prevent collapse of model during training, we perform two sub steps of model fine-tuning. (1) We first fine-tune the model on the *Synthetic Image Set* to establish the initial foundation. We set the learning rate to be 0.001 and decreased by 10% at the $30,000$ and $40,000$ iterations, and finally became 10^{-5}. (2) We then fine-tuned the obtained model with the combined dataset of *Synthetic + Real Image Set* for another $50,000$ iterations.

Training with PointRend. We noticed an obvious drawback of shivering or over-smoothed edges of segmentation produced by Mask-RCNN as shown in Fig. 7(a). This imprecise segmentation boundary can be effectively improved by using the *Point-based Rendering* (PointRend) neural network [6]. PointRend works in a way similar to the efficient rendering method from computer graphics, which can address the over- and under-sampling challenges faced in the pixel labeling tasks in segmentation. In our case, PointRend can effectively improve the Mask-RCNN in producing crisp segmentation boundaries of the flags as shown in Fig. 7(b). However, we note that while PointRend can significantly improve the binary flag segmentation problem, our experimental results show that it does not improve the multi-nation flag segmentation problem. This will be discussed further in the next section.

	Backbone	AP	AP_{50}	AP_{75}	AP_s	AP_m	AP_l
YOLACT++	ResNet-101-FPN	17.26	20.32	18.27	-	-	-
Mask-RCNN	ResNet-50-FPN	69.90	87.72	78.16	18.00	45.62	77.07
Mask-RCNN	ResNet-101-FPN	73.85	89.13	80.35	17.02	47.32	81.96
Mask-RCNN	ResNeXt-101-FPN	72.30	88.81	79.02	**32.65**	48.76	79.36
PointRend	ResNet-101-FPN	**76.30**	**90.24**	81.90	20.40	47.68	**84.68**
PointRend	ResNeXt-101-FPN	76.21	89.63	**82.06**	25.91	**49.75**	84.30

Table 1. mAP of the proposed binary flag detection on the FlagDetSeg OIDv6 Segmentation test set.

	Backbone	AP	AP_{50}	AP_{75}	AP_s	AP_m	AP_l
Mask-RCNN	ResNet-101-FPN	**87.92**	**93.1**	**92.36**	40.8	80.03	**91.03**
Mask-RCNN	ResNeXt-101-FPN	85.81	90.75	90.11	**44.71**	**81.58**	88.84
PointRend	ResNet-101-FPN	83.91	87.68	87.13	31.49	77.82	88.04
PointRend	ResNeXt-101-FPN	82.05	85.23	84.94	32.96	79.22	85.34

Table 2. mAP of the proposed multi-nation flag detection with segmentation on the FlagDetSeg Real + Synthetic Image test set.

4. Experimental Results

We use the standard Mean Average Precision (mAP) [4] to evaluate the performance of the flag detection models. Note that the mAP can be well-defined for both the binary segmentation model and the multi-nation segmentation models intuitively. The mAP calculates the intersection of union (IoU) scores between predicted segmentation and annotated polygon. We use AP_N to denote the AP score with IoU threshold set to be $N\%$. Generally, a larger value of N represents a stricter condition in matching acceptance. We use AP_s, AP_m, and AP_l to indicate APs for detected objects that are small (less than 32×32), medium (96×96), and large (greater than 96×96), respectively. The AP_s, AP_m, and AP_l scores indicate how well the detector can detect small or large sized objects in the view.

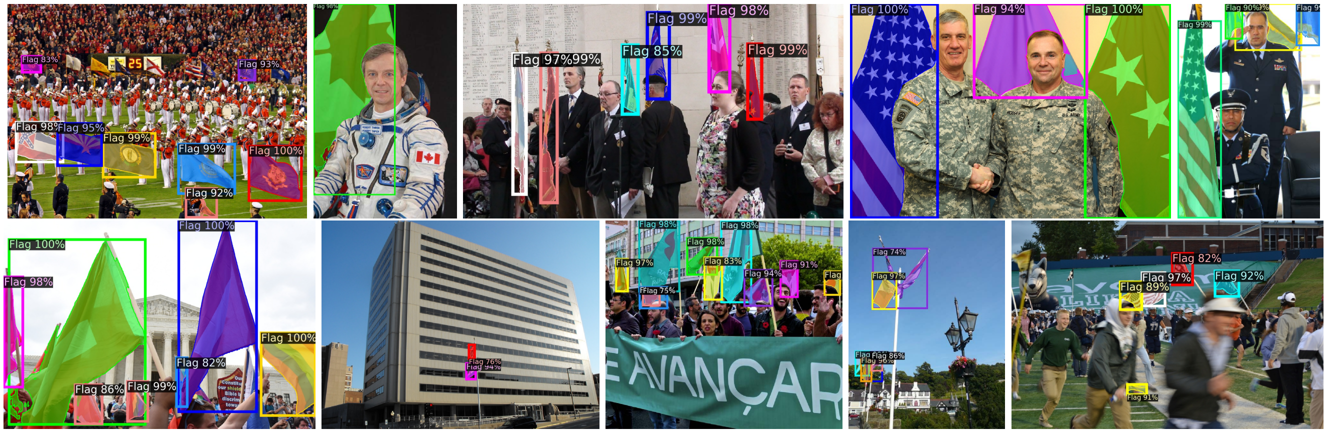

Figure 6. **Examples of binary flag detection.** We do not expect to detect tiny flags such as the ones on the uniform or space suit.

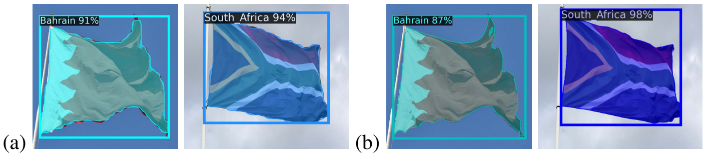

Figure 7. Comparisons of mask boarders predicted by (a) Mask-RCNN and (b) PointRend. Observe that PointRend segmentation contours are more accurate.

Flag detection with binary classification. Table 1 shows the comparison of three binary flag instance segmentation methods, namely, YOLACT++ [5], Mask-RCNN [4], and PointRend [6] with various backbones. Observe that YOLACT++ is not ideal for delineating complex flag instances, thus with inferior performance. While Mask-RCNN performs decently, PointRend can produce more precise flag segmentation contours. Fig. 6 shows examples of binary flag detection results produced by PointRend with ResNet-101-FPN. Our trained model can very well detect multiple flags including tiny ones from clutter background.

Flag detection with multi-nation classification. Table 2 shows the comparison of Mask-RCNN and PointRend with two backbones for multi-nation flag instance segmentation. Although ResNeXt can detect more small- and medium-sized objects, its overall performance is inferior to those of ResNet as the backbone. Visual results of this experiment are shown in Fig. 2. Observe that many flags that are under warping, cropping, and occluded views can all be successfully localized and classified. Note that although our model is trained using a multi-nation data, it can successfully detect many non-national flags that belongs to businesses or organizations, which will be classified into a 'Others' type.

Limitation and failure cases. Fig. 8 shows failure cases of our model for multi-nation flag classification. These failures occur when the flag instances are with substantial information loss, so the model had a hard time fitting them into the closest class. In Fig. 8(a), the smaller flag was classified as `Taiwan`, which can probably attributed to the blue block at its top-left corner. Such mis-classification is possibly due to insufficient information, as the flag can be successfully segmented out (*i.e.* binary segmentation is still successful). Fig. 8(b) show another case that the flag behind the speaker is heavily occluded and causes mis-classification due to missing critical features for classification. Similarly in Fig. 8(c), the rear flag of `Venezuela` is wrongly classified as `Columbia` because the stars on the flag are hidden. Another example in Fig. 8(d), the ambiguity lies in that both `Vietnam` and `Cameroon` have yellow stars in the middle while the head and the tail of the flag are clipped.

5. Conclusion

We presented a multi-nation flag detection approach that can detect and segment flags of multiple nations in-the-wild.

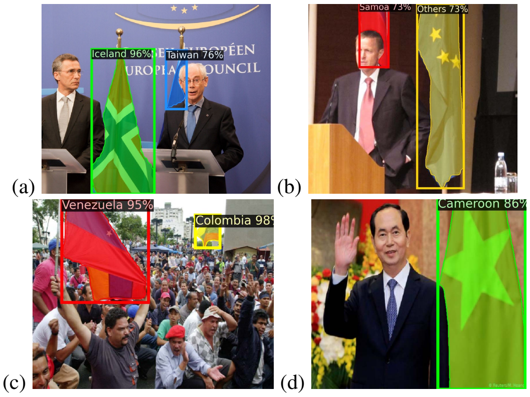

Figure 8. **Challenging cases.** Although flags can be detected in many cases, substantial information loss can lead to mis-classification.

Flag detection has wide applications in automatic media content understanding. Based on data aggregation, annotation, hallucination, and augmentation, we showed a way to effectively build a diverse multi-nation flag image dataset that can be used to train a flag instance segmentation model with little manual annotation burden. The proposed multi-stage training and fine-tuning approach can effectively cope with the large variability of flags with the lack of authentic annotated flags. Our experimental results show that the proposed method can not only detect and localize flags with precise segmentation, it can also accurately predict the flag types out of 200+ nations. The proposed method represents the new state-of-the-art for multi-nation flag detection with instance segmentation.

For future work, we will expand the model to recognize of flags from multiple business and organizations. We also plan to further generate synthetic dataset using 3D Computer Graphics tools that can better simulate 3D flag warping and folding using better transformation and sheet-based texture simulation. For synthetic flag image generation, image harmonization based on background domains can produce more visually realistic flag images for training. Finally, semi-supervised learning such as teacher-student paradigm might be an alternative to the supervised learning we took in this work.

Acknowledgment. This material is based on work supported in part by the U.S. Defense Advanced Research Projects Agency (DARPA) contract K003088-00-S04.

References

[1] T. Said and M. Barr, "Countries flags detection based on local context network and color features," *Multimedia Tools and Applications*, vol. 80, pp. 14 753–14 765, 2021.

[2] L. Liu, W. Ouyang, X. Wang, P. Fieguth, J. Chen, X. Liu, and M. Pietikainen, "Deep learning for generic

object detection: A survey," *IJCV*, vol. 128, pp. 261–318, 2020.

[3] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal speed and accuracy of object detection," in *arXiv:2004.10934*, 2020.

[4] K. He, G. Gkioxari, P. Dollar, and R. Girshick, "Mask R-CNN," in *ICCV*, 2017.

[5] D. Bolya, C. Zhou, F. Xiao, and Y. J. Lee, "YOLACT++: Better real-time instance segmentation," in *arXiv:1912.06218*, 2019.

[6] A. Kirillov, Y. Wu, K. He, and R. Girshick, "PointRend: Image segmentation as rendering," in *CVPR*, 2020.

[7] G. Ghiasi, Y. Cui, A. Srinivas, R. Qian, T.-Y. Lin, E. D. Cubuk, Q. V. Le, and B. Zoph, "Simple copy-paste is a strong data augmentation method for instance segmentation," in *CVPR*, 2021.

[8] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *NIPS*, 2015, pp. 91–99.

[9] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," 2016.

[10] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," in *arXiv.org*, 2018.

[11] ——, "YOLO9000: Better, faster, stronger," in *arxiv:1612.08242*, 2016.

[12] A. M. Hafiz and G. M. Bhat, "A survey on instance segmentation: state of the art," *IJMIR*, vol. 9, pp. 171–189, 2020.

[13] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *CVPR*, 2015, pp. 3431–3440.

[14] S. Xie, R. Girshick, P. Dollar, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *CVPR*, 2017.

[15] Z. Huang, L. Huang, Y. Gong, C. Huang, and X. Wang, "Mask scoring r-cnn," in *CVPR*, 2019.

[16] X. Chen, R. Girshick, K. He, and P. Dollár, "Tensormask: A foundation for dense object segmentation," in *arXiv*, 2019.

[17] D. Bolya, C. Zhou, F. Xiao, and Y. J. Lee, "YOLACT: Real-time instance segmentation," in *ICCV*, 2019.

[18] S. Jetley, A. Vaze, and S. Belhe, "Automatic flag recognition using texture based color analysis and gradient features," in *ICIIP*, 2013.

[19] E. Hart, S.-H. Cha, and C. Tappert, "Interactive flag identification using image retrieval techniques," in *CISST*, 2004.

[20] A. Lodh and R. Parekh, "Computer aided identification of flags using color features," *International Journal of Computer Applications*, vol. 149, no. 11, 2016.

[21] K. Hao, Z. Qu, and Q. Gong, "Color flag recognition based on HOG and color features in complex scene," in *ICDIP*, 2017.

[22] H. H. Duc and K. Jung, "Applying tensorflow with convolutional neural networks to train data and recognize national flags," in *International Conference on Multimedia and Ubiquitous Engineering*, 2017.

[23] M. Gu, K. Hao, and Z. Ou, "Flag detection with convolutional network," in *CSAI*, 2018.

[24] V. Manojlović, "Flagnet," 2019, https://github.com/iamvukasin/flagnet.

[25] P. Pai, "SSD based object detection for country flag cards," 2017, https://github.com/Prasad9/Detect-Flags-SSD.

[26] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *Journal of Big Data*, vol. 6, no. 1, 2019.

[27] A. B. Jung, K. Wada, J. Crall, S. Tanaka, J. Graving, C. Reinders, S. Yadav, J. Banerjee, G. Vecsei, A. Kraft, Z. Rui, J. Borovec, C. Vallentin, S. Zhydenko, K. Pfeiffer, B. Cook, I. Fernández, F.-M. De Rainville, C.-H. Weng, A. Ayala-Acevedo, R. Meudec, M. Laporte *et al.*, "imgaug," https://github.com/aleju/imgaug, 2020.

[28] A. Buslaev, V. I. Iglovikov, E. Khvedchenya, A. Parinov, M. Druzhinin, and A. A. Kalinin, "Albumentations: Fast and flexible image augmentations," *Information*, vol. 11, no. 2, 2020. [Online]. Available: https://www.mdpi.com/2078-2489/11/2/125

[29] A. Antoniou, A. Storkey, and H. Edwards, "Data augmentation generative adversarial networks," in *ICLR*, 2018.

[30] A. Kuznetsova, H. Rom, N. Alldrin, J. Uijlings, I. Krasin, J. Pont-Tuset, S. Kamali, S. Popov, M. Malloci, A. Kolesnikov, T. Duerig, and V. Ferrari, "The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale," *IJCV*, 2020.

[31] P. Skalski, "Make Sense," https://github.com/SkalskiP/make-sense/, 2019.

[32] P. image editing, "Patrick perez and michel gangnet and andrew blake," *ACM Transactions of Graphics*, vol. 22, 2003.

[33] W. Cong, L. Niu, J. Zhang, J. Liang, and L. Zhang, "BargainNet: Background-guided domain translation for image harmonization," in *ICME*, 2021.

[34] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick, "Detectron2," https://github.com/facebookresearch/detectron2, 2019.