This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

FA-RPN: Floating Region Proposals for Face Detection

Mahyar Najibi * Bharat Singh * Larry S. Davis
{najibi,bharat,lsd}@cs.umd.edu

Abstract

We propose a novel approach for generating region proposals for performing face detection. Instead of classifying anchor boxes using features from a pixel in the convolutional feature map, we adopt a pooling-based approach for generating region proposals. However, pooling hundreds of thousands of anchors which are evaluated for generating proposals becomes a computational bottleneck during inference. To this end, an efficient anchor placement strategy for reducing the number of anchor-boxes is proposed. We then show that proposals generated by our network (Floating Anchor Region Proposal Network, FA-RPN) are better than RPN for generating region proposals for face detection. We discuss several beneficial features of FA-RPN proposals (which can be enabled without re-training) like iterative refinement, placement of fractional anchors and changing size/shape of anchors. Our face detector based on FA-RPN obtains 89.4% mAP with a ResNet-50 backbone on the WIDER dataset.

1. Introduction

Face detection is an important computer vision problem and has multiple applications in surveillance, tracking, consumer-facing devices like iPhones *etc*. Hence, various approaches have been proposed towards solving it [41, 43, 17, 45, 19, 36, 44, 29, 25] and successful solutions have also been deployed in practice. So, expectations from face detection algorithms are much higher and error rates today are quite low. Algorithms need to detect faces which are as small as 5 pixels to 500 pixels in size. As localization is essential for detection, evaluating every small region of the image is important. Face detection datasets can have up to a thousand faces in a single image, which is not common in generic object detection.

Detectors like Faster-RCNN [30] employ a region proposal network (RPN) which places anchor boxes of different sizes and aspect ratios uniformly on the image and classifies them for generating object-like regions. However, RPN only uses a single pixel in the convolutional feature map for evaluating the proposal hypotheses, independent of the size of the object. Therefore, the feature representation in RPN entirely relies on the contextual information encoded in the high-dimensional feature representation generated at the pixel. It does not pool features from the entire extent of an object while generating the feature representation, see Fig. 1. Thus, it can miss object regions or generate proposals which are not well localized. Further, it is not possible to iterate and refine the positions of the anchor-boxes as part of the proposal network. If objects of different scale/aspect-ratios are to be learned or if we want to place anchors at sub-pixel resolution, filters specific to each of these conditions need to be added during training. Generating proposals using a pooling-based algorithm can alleviate such problems easily.

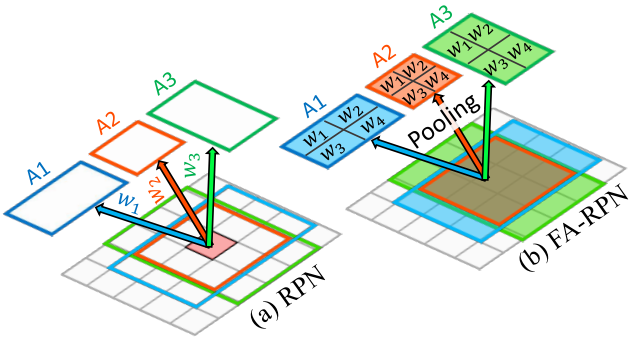


Figure 1: Difference between RPN and FA-RPN in terms of weight configuration. For simplicity we show 2x2 pooling for FA-RPN.

There are predominantly two pooling-based methods for the final classification of RoIs in an image - Fast-RCNN [13] and R-FCN [10]. Fast-RCNN projects the region-proposals to the convolutional feature map, and pools the features inside the region of interest (RoI) to a fixed size grid (typically 7×7) and applies two fully connected layers which perform classification and regression. Due to computational constraints, this approach is practically infeasible for proposal generation as one would need to apply it to hundreds of thousands of regions - which is the number of

*Equal Contribution

region candidates which are typically evaluated by a region proposal algorithm.

To reduce the dependence on fully connected layers, R-FCN performs local convolutions (7×7) inside an RoI for capturing the spatial extent of each object. Since each of these local filters can be applied to the previous feature-map, we just need to pool the response from the appropriate region corresponding to each local filter. This makes it a good candidate for a pooling-based proposal approach as it is possible to apply it to a large number of RoIs efficiently. However, in high-resolution images, proposal algorithms like RPN evaluate hundreds of thousands of anchors during inference. It is computationally infeasible to perform pooling on that many regions. Luckily, many anchors are not necessary (*e.g*. large anchors which are very close to each other). In this paper, we show that careful anchor placement strategies can reduce the number of proposals significantly to the point where a pooling-based algorithm becomes feasible for proposal generation. This yields an efficient and effective objectness detector which does not suffer from the aforementioned problems present in RPN designs.

A pooling-based proposal method based on R-FCN which relies on position sensitive filters is particularly well suited for face detection. While objects deform and positional correspondence between different parts is often lost - faces are rigid, structured and parts have positional semantic correspondence (*e.g*. nose, eyes, lips). Moreover, it is possible to place anchor boxes of different size and aspect ratios without adding more filters. We can also place fractional anchor boxes and perform bilinear interpolation while pooling features for computing objectness. We can further improve the localization performance of the proposal candidates by iteratively pooling again from the generated RoIs and all these design changes can be made during inference! Due to these reasons, we refer to our proposal network as *Floating Anchor Region Proposal Network* (FA-RPN). We highlight these advantages in Fig. 1 and Fig. 2. On the WIDER dataset [42] we show that FA-RPN proposals are better than RPN proposals. FA-RPN also obtains state-of-the-art results on WIDER and PascalFaces which demonstrates its effectiveness for face detection.

2. Related Work

Generating class agnostic region proposals has been investigated in computer vision for more than a decade. Initial methods include multi-scale combinatorial grouping [2], constrained parametric min-cuts [38], selective search [7] *etc*. These methods generate region proposals which obtain high recall for objects in a category agnostic fashion. They were also very successful in the pre-deep learning era and obtained state-of-the-art performance even with a bag-of-words model [38]. Using region proposals based on selective search [38], R-CNN [14] was the first deep learning based detector. Unsupervised region proposals were also used in later detectors like Fast-RCNN [13] but since the Faster-RCNN detector [30] generated region proposals using a convolutional neural network, it has become the *de-facto* algorithm for generating region proposals.

To improve RPN, several modifications have been proposed. State-of-the-art detectors can also detect objects in a single step. Detectors like SSH [25], SSD [22], RetinaNet [21], MS-CNN [5] generate multi-scale feature maps to classify and regress anchors placed on these feature-maps. These single-shot detectors are closely related to the region proposal network as they have specific filters to detect objects of different sizes and aspect ratios but also combine feature-maps from multiple layers of the deep neural network. No further refinement is performed after the initial offsets generated by the network are applied. Another class of detectors are iterative, like G-CNN [24], Cascade-RCNN [6], LocNet [12], FPN [20], RFCN-3000 [34], Faster-RCNN [30]. These detectors refine a pre-defined set of anchor-boxes in multiple stages and have more layers to further improve classification and localization of regressed anchors. One should note that even in these networks, the first stage comprises of the region proposal network which eliminates the major chunk of background regions. FA-RPN is closer to this line of work but, in contrast, it supports iterative refinement of *region proposals* during inference.

We briefly review some recent work on face detection. With the availability of large scale datasets like WIDER [42] which contain many small faces in high resolution images, multiple new techniques for face detection have been proposed [41, 43, 17, 45, 19, 36, 44, 29, 3]. A lot of focus has been on scale, combining features of different layers [17, 25, 44, 43, 8, 18] and improving configurations of the region proposal network [44, 43]. For example, in finding tiny faces [17], it is proposed to perform detection on an image pyramid and to have different scale filters for objects of different sizes. SSH [25] and S3FD [43] efficiently utilize the intermediate layers of the network. PyramidBox [37] replaces the context module in SSH by deeper and wider sub-networks to better capture the contextual information for face detection. Recently, even GANs [15] have been used to improve the performance on tiny faces [3].

In face detection, the choice of anchors and their placement on the image is very important [44, 43]. For example, using extra strided anchors were shown to be beneficial [44]. Geometric constraints of the scene have also been used to prune region proposals [1]. Some of these changes require re-training RPN again. In our framework, design decisions such as evaluating different anchor scales, changing the stride of anchors, and adding fractional anchors can simply be made during inference as we share filters for all object sizes and only pooling is performed for them. Moreover, a pooling-based design also provides precise spatial

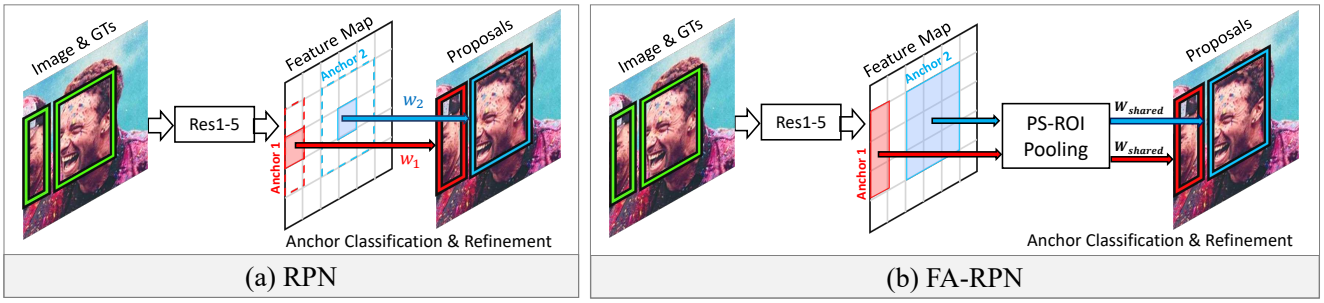


Figure 2: We highlight the major differences between RPN (a) and FA-RPN (b) proposals. RPN performs classification on a single pixel in the high-dimensional feature-map and uses different weights for classifying anchor-boxes of different sizes/aspect ratios. FA-RPN proposals, on the other hand, pool features from multiple bins in the image and share weights across objects of different sizes and aspect ratios.

features.

3. Background

We provide a brief overview of the R-FCN detector in this section. This detector uses RPN to generate region proposals. It classifies the top 2000 ranked proposals using the R-FCN detector. Classification is performed over all the foreground classes and the background class. The key component in R-FCN is local convolutions. It applies different filters in different sub-regions of an RoI for inferring the spatial extent of an object. These sub-regions may correspond to parts of an object. To accelerate this process over thousands of RoIs, convolution for each part in each object class is performed in the final layer. So, as an example, if there are 21 classes, the last feature-map would contain 21×49 channels. Then, given an RoI, *Position Sensitive RoIPooling* is performed on this feature-map to obtain the effect of local convolutions [10]. We refer the reader to the R-FCN [10] paper for further details on PSRoIPooling. Finally, the response is average pooled and used as the classification score of the object. In Deformable-RFCN [11], the regions for each bin where pooling is performed are also adjusted based on the input feature-map, which is referred to as deformable PSRoIPooling.

4. FA-RPN - Floating Anchor Region Proposal Network

In this section, we discuss the training of FA-RPN, which performs iterative classification and regression of anchors placed on an image for generating accurate region proposals. An overview of our approach is shown in Fig. 3.

4.1. Anchor Placement

In this architecture, classification of anchors is not performed using a single high-dimensional feature vector but by pooling features inside the RoI. Hence, there are no restrictions on how anchors can be placed during training and inference. As long as the convolutional filters can learn objectness, we can apply the model on RoIs of different sizes and aspect ratios, even if the network was not trained explicitly for those particular scales and aspect-ratios.

FA-RPN places anchors of different scales and aspect ratios on a grid, as generated in the region proposal network, and clips the anchors which extend beyond the image. While placing anchors, we vary the spatial stride as we increase the anchor size. Since nearby anchors at larger scales have a very high overlap, including them is not necessary. We change the stride of anchor-boxes to $max(c, s/d)$, where s is square-root of the area of an anchor-box, c is a constant and d is the scaling factor, shown in Fig. 3. In practice, we set c to 16 and d to 5. This ensures that not too many overlapping anchor-boxes are placed on the image, while ensuring significant overlap between adjacent anchors to cover all objects. Naive placement of anchor boxes of 3 aspect ratios and 5 scales with stride equaling 16 pixels in an 800×1280 image leads to a 2-3$\times$ slow-down when performing inference. With the proposed placement method, we reduce the number of RoIs per image from 400,000 to 100,000 for a 1280×1280 image for the above-mentioned anchor configuration. When we increase the image size, the computation for convolution also increases proportionally, so as long as the time required for pooling is not significant compared to convolution, we will not observe a noticeable difference in performance.

There is no restriction that the stride of anchors should be the same as the stride of the convolutional feature-map. We can even place RoIs between two pixels in the convolutional feature-map without making any architectural change to the network. This allows us to augment the ground-truth bounding boxes as positive RoIs during training. This is unlike RPN, where the maximum overlapping anchor is

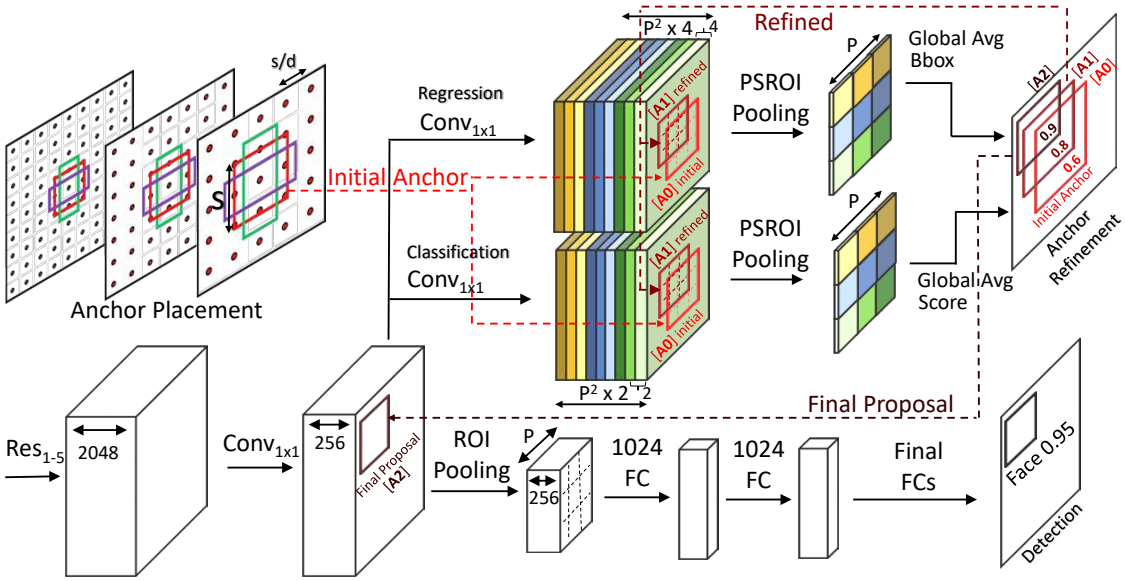


Figure 3: FA-RPN framework. FA-RPN uses multi-scale training. At each training iteration, an image scale is randomly selected and *suitable* anchor scales are placed over the image. This set of initial anchors are used to pool objectness scores and localization information from position sensitive filters (for simplicity only the localization branch is depicted in the figure). For improving localization, the top scoring initial anchors are further refined with subsequent poolings. The refinement process for an initial anchor (*[A0]*) is depicted in the figure. This anchor is first refined to *[A1]* based on the network prediction. Another pooling is performed over *[A1]* to form a new prediction for refining it further to the final anchor *[A2]*. Finally, a Faster-RCNN head is used to perform the final classification and regression.

assigned as positive when no anchor matches the overlap threshold criterion. We show qualitative examples of anchor placement for different scales and aspect ratios in FA-RPN in Fig. 3.

4.2. Sampling

Since there are hundreds of thousands of anchors which can be placed on an image, we sample anchors during training. We observe that using focal loss [21] reduced recall for RPN (hyper-parameter tuning could be a reason), so we did not use it for FA-RPN. We use the commonly used technique of sampling RoIs for handing class imbalance. In FA-RPN, an anchor-box is marked as positive if its overlap with a ground-truth box is greater than 0.5. An anchor is marked as negative if its overlap is less than 0.4. A maximum of 128 positive and negative anchors are sampled in a batch. Since the probability of a random anchor being an easy sample is high, we also sample 32 anchor-boxes which have an overlap of at least 0.1 with the ground-truth boxes as hard negatives. Just for training FA-RPN proposals, all other RoIs can be ignored. However, for training an end-to-end detector, we also need to score other RoIs in the image. When training an end-to-end detector, we select a maximum of 50,000 RoIs in an image (prioritizing those which have at-least 0.1 overlap with ground-truth boxes first).

4.3. Iterative Refinement

The initial set of placed anchors are expected to cover the ground-truth objects present in the image. However, these anchors may not always have an overlap greater than 0.5 with all objects and hence would be given low scores by the classifier. This problem is amplified for small object instances as mentioned in several methods [43, 17]. In this case, no anchor-boxes may have a high score for some ground-truth boxes. Therefore, the ground-truth boxes may not be covered in the top 500 to 1000 proposals generated in the image. In FA-RPN, rather than selecting the top 1000 proposals, we generate 20,000 proposals during inference and then perform pooling again on these 20,000 proposals from the same feature-map (we can also have another convolutional layer which refines the first stage region proposals). The hypothesis is that after refinement, the anchors would be better localized and hence the scores which we obtain after pooling features inside an RoI would be more reliable. Therefore, after refinement, the ordering of the top 1000 proposals would be different because scores are pooled from refined anchor-boxes rather than the anchor-boxes which were placed uniformly on a grid. Since we only need to perform pooling for this operation, it is efficient and can be easily implemented when the number of

RoIs is close to 100,000. Note that our method is entirely pooling-based and does not have any fully connected layers like cascade-RCNN [6] or G-CNN [24]. Therefore, it is much more efficient for iterative refinement.

4.4. Complexity and Speed

FA-RPN is efficient. Namely, on 800 $\times$ 1280 size images, it takes 50 milliseconds to perform forward propagation for our network on a P6000 GPU. We also discuss how much time it takes to use R-FCN for end-to-end detection. For general object detection, when the number of classes is increased, to say 100, the contribution from the pooling layer also increases. This is because the complexity of pooling is linear in the number of classes. So, if we increase the number of classes to 100, this operation would become 100 times slower and at that stage, pooling will account for a significant portion of the time in forward-propagation. For instance, without our anchor placement strategy, it takes 100 seconds to perform inference for 100 classes in a single image on a V100 GPU. However, as for face detection, we only need to perform pooling for 2 classes and use a different anchor placement scheme, we do not face this problem and objectness can be efficiently computed even with tens of thousands of anchor boxes.

4.5. Scale Normalized Training

The positional correspondence of R-FCN is lost when RoI bins become too small. The idea of local convolution or having filters specific to different parts of an object is relevant when each bin corresponds to a unique region in the convolutional feature-map. The position-sensitive filters implicitly assume that features in the previous layer have a resolution which is similar to that after PSRoIPooling. Otherwise, if the RoI is too small, then all the position sensitive filters will pool from more or less the same position, nullifying the hypothesis that these filters are position sensitive. Therefore, we perform scale normalized training [33], which performs selective gradient propagation for RoIs which are close to a resolution of 224 $\times$ 224 and excludes those RoIs which can be observed at a better resolution during training. In this setting, the position-sensitive nature of filters is preserved to some extent, which helps in improving the performance of FA-RPN.

5. Datasets

We perform experiments on three benchmark datasets, WIDER [42], AFW [46], and Pascal Faces [40]. The WIDER dataset contains 32,203 images with 393,703 annotated faces, 158,989 of which are in the train set, 39,496 in the validation set, and the rest are in the test set. The validation and test set are divided into "easy", "medium", and "hard" subsets cumulatively (i.e. the "hard" set contains all faces and "medium" contains "easy" and "medium"). This is the most challenging public face dataset mainly due to the significant variation in the scale of faces and occlusion. We train all models on the train set of the WIDER dataset and evaluate on the validation set. We mention in our experiments when initialization of our pre-trained model is from ImageNet or COCO. Ablation studies are also performed on the validation set (i.e. "hard" subset which contains the whole dataset). Pascal Faces and AFW have 1335 and 473 faces respectively. We use Pascal Faces and AFW only as test sets for evaluating the generalization of our trained models. When performing experiments on these datasets, we apply the model trained on the WIDER train set out of the box.

6. Experiments

We train a ResNet-50 [16] based Faster-RCNN detector with deformable convolutions [11] and SNIP [33]. FA-RPN proposals are generated on the concatenated `Conv4` and `Conv5` features. On WIDER we train on the following image resolutions $(1800, 2800)$, $(1024, 1440)$ and $(512, 800)$. The SNIP ranges we use for WIDER are as follows, [0, 200) for (1800, 2800), [32, 300) for (1024, 1440) and [80, ∞) for (512, 800) as the size of the shorter side of the image is around 1024. We train for 8 epochs with a stepdown at 5.33 epochs. In all experiments, we use a learning rate and weight decay of 0.0005 and train on 8 GPUs. We use the same learning rate and training schedule even when training on 4 GPUs. In all our experiments, we use online hard example mining (OHEM) [32] to train the 2 fully connected layers in our detector. Hard example mining is performed on 900 proposals with a batch size of 512. RoIs which have an overlap greater than 0.5 with ground-truth bounding boxes are marked as positive and anything less than that is labeled as negative. We use Soft-NMS [4] with $\sigma = 0.35$ when performing inference. Since Pascal Faces and AFW contain low-resolution images and also do not contain faces as small as the WIDER dataset, we do not perform inference on the 1800×2800 resolution. All other parameters remain the same as the experiments on the WIDER dataset.

On the WIDER dataset, we remove anchors for different aspect ratios (*i.e.* we only have one anchor per scale with an aspect ratio of 1) and add a 16×16 size anchor for improving the recall for small faces. Note that extreme size anchors are removed during training with SNIP using the same rules which are used for training the detector. With these settings, we outperform state-of-the-art results on the WIDER dataset demonstrating the effectiveness of FA-RPN. However, the objective of this paper is not to show that FA-RPN is necessary to obtain state-of-the-art performance. FA-RPN is an elegant and efficient alternative to RPN and can be combined with multi-stage face detection methods to improve performance.

Method	AP
Baseline	87.2
Baseline + SNIP	88.1
Baseline + SNIP + COCO pre-training	89.1
Baseline + SNIP + COCO pre-training + Iteration	89.4

Table 1: Ablation analysis with different core-components of our face detector on the hard-set of the WIDER dataset (hard-set contains all images in the dataset).

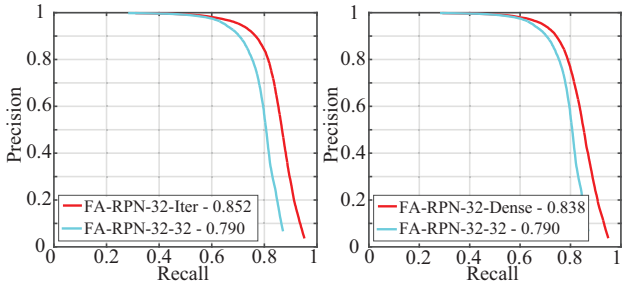


Figure 4: Ablation analysis: improving precision at inference time. *FA-RPN-32-32* represents a model which is trained by increasing the stride between anchors to 32 and uses the same stride at inference time. (a) *FA-RPN-32-Iter* is the same model when an additional anchor refinement step is performed at *inference*. (b) *FA-RPN-32-Dense* on the other hand, improves precision by reducing the anchor stride at inference time to our original FA-RPN stride.

6.1. Effect of Multiple Iterations in FA-RPN

We evaluate FA-RPN on WIDER when we perform multiple iterations during inference. Since FA-RPN operates on RoIs rather than classifying single-pixel feature-maps like RPN, we can further refine the RoIs which are generated after applying the regression offsets. As the initial set of anchor boxes are coarse, the RoIs generated after the first step are not very well localized. Performing another level of pooling on the generated RoIs helps to improve recall for our proposals. As can be seen in Table 1 and Fig. 4a, this refinement step helps to improve precision and recall. We also generate anchors with different strides - 16 and 32 pixels - and show how the final detection performance improves as we iteratively refine proposals.

6.2. Modifying Anchors and Strides during Inference

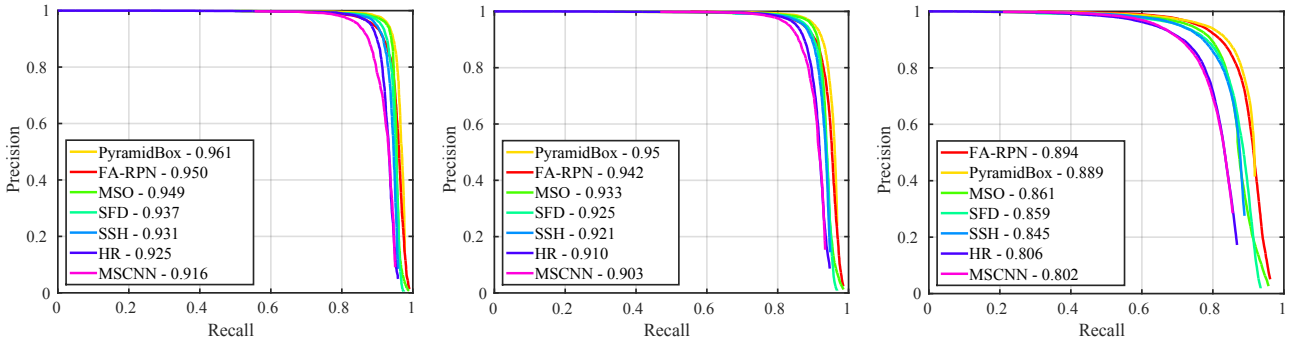
In this section, we show the flexibility of FA-RPN for generating region proposals. We train our network with a stride of 32 pixels and during inference, we generate anchors at a stride of 16 pixels on the WIDER dataset. The result is shown in the right-hand side plot in Fig. 4. We notice that the dense anchors improve performance by 3.8%. On the left side of the plot, we show the effect of iterative refinement of FA-RPN proposals. This further provides a boost of 1.4% on top of the denser anchors. This shows that our network is robust to changes in the anchor configuration, and can detect faces even on anchor sizes which were not provided during training. To achieve this with RPN, one would need to re-train it again, while in FA-RPN it is a simple inference time hyper-parameter which can be tuned on a validation set even after the training phase.

6.3. Effect of Scale and COCO pre-training on Face Detection

Variation of scale is among the main challenges in detection datasets. Datasets like WIDER consist of many small faces which can be hard to detect for a CNN at the original image scale. Therefore, upsampling images is crucial to obtaining good performance. However, as shown in [33], when we upsample images, large objects become hard to classify and when we downsample images to detect large objects, small objects become harder to classify. Therefore, standard multi-scale training is not effective when using extreme resolutions. In Table 1 we show the effect of performing SNIP based multi-scale training in our FA-RPN based Faster-RCNN detector. When performing inference on the same resolutions, we observe an improvement in detection performance on the WIDER dataset by 1%. Note that this improvement is on top of multi-scale inference. We also initialized our ResNet-50 model which was pre-trained on the COCO detection dataset. We show that even pre-training on object detection helps in improving the performance of face detectors by a significant amount, Table 1.

6.4. Comparison on the WIDER Dataset

We compare our method with MSCNN [5], HR [17], SSH [25], S3FD [43], MSO [44], and PyramidBox [37] which are the published state-of-the-art methods on the WIDER dataset. Our simple detector outperforms other methods on the "hard" set, which includes *all* the annotations in the WIDER dataset while achieving an average precision of 89.4%. We also perform well in the "easy" and "medium" subsets. The precision-recall plots for each of these cases are shown in Fig. 5. Note that we did not use feature-pyramids or lower layer features from `Conv2` and `Conv3` [25, 43, 17] , enhancing predictions with context [17] or with deeper networks like ResNext-152 [39]/ Xception [9] for obtaining these results. We also compare FA-RPN (baseline version in Table 1) with RPN quantitatively and qualitatively in Fig. 6 and Fig. 7 respectively. These results demonstrate that FA-RPN is competitive with existing proposal techniques as it can lead to a state-of-the-art face detector. We also do not use recently proposed tech-



(a) Easy (b) Medium (c) Hard (whole dataset)

Figure 5: We compare with recently published methods on the WIDER dataset. The plots are for "easy", "medium" and "hard" respectively from left to right. As can be seen, FA-RPN outperforms published baselines on this dataset. Note that, "hard" set contains the whole dataset while "easy" and "medium" are subsets.

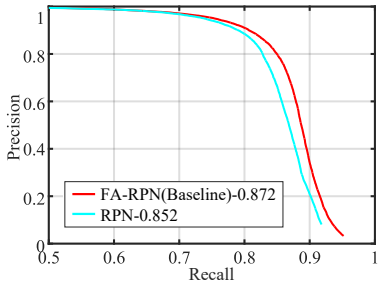


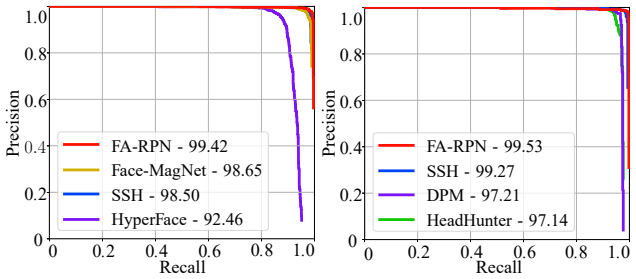
Figure 6: Comparison with RPN on the WIDER dataset.

Figure 7: Qualitative comparison between RPN and FA-RPN (Baseline). Gold rectangles are those detected by both, greens are detected by FA-RPN but missed by RPN.

niques like stochastic face lifting [44], having different filters for different size objects [17] or maxout background loss [43]. Our performance can be further improved if the above mentioned architectural changes are made to our network or better training methods which also fine-tune batch-normalization statistics are used [27, 35].

6.5. Pascal Faces and AFW Datasets

To show the generalization of our trained detector, we also apply it out-of-the-box to the Pascal Faces [40] and AFW [46] datasets without fine-tuning. The performance of FA-RPN is compared with SSH [25], Face-Magnet [31], HyperFace [29], HeadHunter [23], and DPM [28] detectors which reported results on these datasets. The results are shown in Fig. 8. Compared to WIDER, the resolution of PASCAL images is lower and they do not contain many small images, so it is sufficient to apply FA-RPN to the two lower resolutions in the pyramid. This also leads to faster inference. As can be seen, FA-RPN out-of-the-box generalizes well to these datasets. FA-RPN achieves state-of-the-art result on the PascalFaces and reduces the error rate to 0.68% on this dataset.



(a) (b)

Figure 8: Comparison with other methods on (a) Pascal Faces, and (b) AFW datasets.

6.6. Efficiency

Our FA-RPN based detector is efficient and takes less than 0.05 seconds on a 1080Ti GPU to perform inference on an image of size 800×1280. With advances in GPUs over the last few years, performing inference even at very high resolutions is efficient. Our detector takes less than 0.4 seconds to process an image of size 1800×2800 on a 1080Ti GPU. With improved GPU architectures and the use of lower precision like 16 or 8 bits, the speed can be further improved by two to four times (depending on the precision used in inference). As a comparison, the original imple-

Figure 9: Qualitative results on the validation set of the WIDER dataset. Green rectangles show the detection and brightness encodes the detection confidence.

mentation of SSH [1] takes 0.45s on a Titax X GPU (ours takes 0.41s on the same machine) to process a 1800×2800 pixels image. It should be noted that in high resolutions, the runtime is dominated by convolutional layers and the small difference may be because *e.g.* SSH uses a custom architecture with feature pyramids, we use a standard ResNet50 backbone, SSH is in Caffe, ours is in MxNet,*etc*. SSH has a better runtime at low-resolutions (*e.g.* on the 512x600 resolution, SSH takes 0.05 and FA-RPN takes 0.07 seconds). However, the runtime of current state-of-the-art methods largely depends on the high-resolution scale. The multi-scale inference used in FA-RPN can be further accelerated with AutoFocus [26].

6.7. Qualitative Results

Figure 9 shows qualitative results on the WIDER validation subset. We picked 20 diverse images to highlight the results generated by FA-RPN. Detections are shown by green rectangles and the brightness encodes the confidence. As can be seen, our face detector works very well in crowded scenes and can find hundreds of small faces in a wide variety of images. This shows that FA-RPN has a high recall and can detect faces accurately. It generalizes well in both indoor and outdoor scenes and under different lighting conditions. Our performance across a wide range of scales is also good without using diverse features from different layers of the network. It is also robust to changes in pose, occlusion, blur, and even works on old photographs!

7. Conclusion

We introduced FA-RPN, a novel method for generating pooling-based proposals for face detection. We proposed techniques for anchor placement and label assignment which were essential in the design of such pooling-based proposal algorithm. FA-RPN has several benefits like efficient iterative refinement, flexibility in selecting scale and anchor stride during inference, sub-pixel anchor placement *etc*. Using FA-RPN, we obtained state-of-the-art results on the challenging WIDER dataset, showing the effectiveness of FA-RPN for this task. FA-RPN also achieved state-of-the-art results out-of-the-box on datasets like PascalFaces showing its generalizability.

Acknowledgement This research is based upon work supported by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via IARPA R&D Contract No. 2014-14071600012. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the ODNI, IARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon.

[1] http://www.github.com/mahyarnajibi/SSH

References

[1] S. Amin and F. Galasso. Geometric proposals for faster r-cnn. In *Advanced Video and Signal Based Surveillance (AVSS), 2017 14th IEEE International Conference on*, pages 1–6. IEEE, 2017. 2

[2] P. Arbeláez, J. Pont-Tuset, J. T. Barron, F. Marques, and J. Malik. Multiscale combinatorial grouping. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 328–335, 2014. 2

[3] Y. Bai, Y. Zhang, M. Ding, and B. Ghanem. Finding tiny faces in the wild with generative adversarial network. *CVPR. IEEE*, 2018. 2

[4] N. Bodla, B. Singh, R. Chellappa, and L. S. Davis. Soft-nms improving object detection with one line of code. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 5562–5570. IEEE, 2017. 5

[5] Z. Cai, Q. Fan, R. S. Feris, and N. Vasconcelos. A unified multi-scale deep convolutional neural network for fast object detection. In *European Conference on Computer Vision*, pages 354–370. Springer, 2016. 2, 6

[6] Z. Cai and N. Vasconcelos. Cascade r-cnn: Delving into high quality object detection. *CVPR*, 2018. 2, 5

[7] J. Carreira and C. Sminchisescu. Constrained parametric min-cuts for automatic object segmentation. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 3241–3248. IEEE, 2010. 2

[8] C. Chi, S. Zhang, J. Xing, Z. Lei, S. Z. Li, and X. Zou. Selective refinement network for high performance face detection. *arXiv preprint arXiv:1809.02693*, 2018. 2

[9] F. Chollet. Xception: Deep learning with depthwise separable convolutions. *CVPR*, 2017. 6

[10] J. Dai, Y. Li, K. He, and J. Sun. R-fcn: Object detection via region-based fully convolutional networks. In *Advances in neural information processing systems*, pages 379–387, 2016. 1, 3

[11] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei. Deformable convolutional networks. *CoRR, abs/1703.06211*, 1(2):3, 2017. 3, 5

[12] S. Gidaris and N. Komodakis. Locnet: Improving localization accuracy for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 789–798, 2016. 2

[13] R. Girshick. Fast r-cnn. *arXiv preprint arXiv:1504.08083*, 2015. 1, 2

[14] 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*, pages 580–587, 2014. 2

[15] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014. 2

[16] 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*, pages 770–778, 2016. 5

[17] P. Hu and D. Ramanan. Finding tiny faces. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1522–1530. IEEE, 2017. 1, 2, 4, 6, 7

[18] J. Li, Y. Wang, C. Wang, Y. Tai, J. Qian, J. Yang, C. Wang, J. Li, and F. Huang. Dsfd: dual shot face detector. *arXiv preprint arXiv:1810.10220*, 2018. 2

[19] Y. Li, B. Sun, T. Wu, and Y. Wang. Face detection with end-to-end integration of a convnet and a 3d model. In *European Conference on Computer Vision*, pages 420–436. Springer, 2016. 1, 2

[20] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature pyramid networks for object detection. In *CVPR*, volume 1, page 4, 2017. 2

[21] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár. Focal loss for dense object detection. *ICCV*, 2017. 2, 4

[22] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016. 2

[23] M. Mathias, R. Benenson, M. Pedersoli, and L. Van Gool. Face detection without bells and whistles. In *European conference on computer vision*, pages 720–735. Springer, 2014. 7

[24] M. Najibi, M. Rastegari, and L. S. Davis. G-cnn: an iterative grid based object detector. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2369–2377, 2016. 2, 5

[25] M. Najibi, P. Samangouei, R. Chellappa, and L. Davis. Ssh: Single stage headless face detector. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4875–4884, 2017. 1, 2, 6, 7

[26] M. Najibi, B. Singh, and L. S. Davis. Autofocus: Efficient multi-scale inference. *arXiv preprint arXiv:1812.01600*, 2018. 8

[27] C. Peng, T. Xiao, Z. Li, Y. Jiang, X. Zhang, K. Jia, G. Yu, and J. Sun. Megdet: A large mini-batch object detector. *CVPR*, 2018. 7

[28] R. Ranjan, V. M. Patel, and R. Chellappa. A deep pyramid deformable part model for face detection. *arXiv preprint arXiv:1508.04389*, 2015. 7

[29] R. Ranjan, V. M. Patel, and R. Chellappa. Hyperface: A deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017. 1, 2, 7

[30] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015. 1, 2

[31] P. Samangouei, M. Najibi, L. Davis, and R. Chellappa. Facemagnet: Magnifying feature maps to detect small faces. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 122–130, 2018. 7

[32] A. Shrivastava, A. Gupta, and R. Girshick. Training region-based object detectors with online hard example mining. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 761–769, 2016. 5

[33] B. Singh and L. S. Davis. An analysis of scale invariance in object detection-snip. *CVPR*, 2018. 5, 6

[34] B. Singh, H. Li, A. Sharma, and L. S. Davis. R-fcn-3000 at 30fps: Decoupling detection and classification. *CVPR*, 2018. 2

[35] B. Singh, M. Najibi, and L. S. Davis. Sniper: Efficient multi-scale training. *arXiv preprint arXiv:1805.09300*, 2018. 7

[36] X. Sun, P. Wu, and S. C. Hoi. Face detection using deep learning: An improved faster rcnn approach. *arXiv preprint arXiv:1701.08289*, 2017. 1, 2

[37] X. Tang, D. K. Du, Z. He, and J. Liu. Pyramidbox: A context-assisted single shot face detector. *ECCV*, 2018. 2, 6

[38] J. R. Uijlings, K. E. Van De Sande, T. Gevers, and A. W. Smeulders. Selective search for object recognition. *International journal of computer vision*, 104(2):154–171, 2013. 2

[39] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He. Aggregated residual transformations for deep neural networks. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pages 5987–5995. IEEE, 2017. 6

[40] J. Yan, X. Zhang, Z. Lei, and S. Z. Li. Face detection by structural models. *Image and Vision Computing*, 32(10):790–799, 2014. 5, 7

[41] S. Yang, P. Luo, C.-C. Loy, and X. Tang. From facial parts responses to face detection: A deep learning approach. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3676–3684, 2015. 1, 2

[42] S. Yang, P. Luo, C.-C. Loy, and X. Tang. Wider face: A face detection benchmark. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5525–5533, 2016. 2, 5

[43] S. Zhang, X. Zhu, Z. Lei, H. Shi, X. Wang, and S. Z. Li. S3fd: Single shot scale-invariant face detector. *ICCV*, 2017. 1, 2, 4, 6, 7

[44] C. Zhu, R. Tao, K. Luu, and M. Savvides. Seeing small faces from robust anchor's perspective. *CVPR*, 2018. 1, 2, 6, 7

[45] C. Zhu, Y. Zheng, K. Luu, and M. Savvides. Cms-rcnn: contextual multi-scale region-based cnn for unconstrained face detection. In *Deep Learning for Biometrics*, pages 57–79. Springer, 2017. 1, 2

[46] X. Zhu and D. Ramanan. Face detection, pose estimation, and landmark localization in the wild. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2879–2886. IEEE, 2012. 5, 7