Mathematical Problems of Computer Science 54, 53–68, 2020.

UDC 004.8

Improving UAV Object Detection through Image Augmentation

Karen M. Gishyan

University of Bath, Bath, United Kingdom
e-mail: karen.gishyan@bath.edu

Abstract

Ground-image based object detection algorithms have had great improvements over the years and provided good results for challenging image datasets such as COCO and PASCAL VOC. These models, however, are not as successful when it comes to unmanned aerial vehicle (UAV)-based object detection and commonly performance deterioration is observed. It is due to the reason that it is a much harder task for the models to detect and classify smaller objects rather than medium-size or large-size objects, and drone imagery is prone to variances caused by different flying altitudes, weather conditions, camera angles and quality. This work explores the performance of two state-of-art-object detection algorithms on the drone object detection task and proposes image augmentation [1] procedures to improve model performance. We compose three image augmentation sequences and propose two new image augmentation techniques and further explore their different combinations on the performances of the models. The augmenters are evaluated for two deep learning models, which include model-training with high-resolution images (1056×1056 pixels) to observe their overall effectiveness. We provide a comparison of augmentation techniques across each model. We identify two augmentation procedures that increase object detection accuracy more effectively than others and obtain our best model using a transfer learning [2] approach, where the weights for the transfer are obtained from training the model with our proposed augmentation technique. At the end of the experiments, we achieve a robust model performance and accuracy, and identify the aspects of improvement as part of our future work.

Keywords: Computer Vision, Deep Learning, Image Processing

1. Introduction

Deep learning-based computer vision algorithms for object detection in images and videos have had much success in the last decade. Object identification and detection from unmanned aerial vehicles (UAVs) have widely gained the attention of researchers, as UAV

[1] Image augmentation is a technique to expand the training dataset by creating modified versions of images in the dataset through different methods of processing.

[2] Transfer Learning is a research problem in Machine Learning that focuses on storing knowledge gained while solving one problem and applying it to a different but related problem.

and computer vision-based applications can be successfully deployed in search-and-rescue, surveillance, agricultural crop identification and counting [1, 2]. However, compared to the object detection models that are trained on ground-based images, there is a significant decline in the performance of detecting objects of such models when applied to UAV-based images. As drones fly at various altitudes and there is a constant change in the viewing angles, UAV-based models have to deal with more visual appearances of the same object to work successfully. As an example, UAVs can look at one object from front-view to side view in a very short period of time, which can result in arbitrary aspect ratios of the objects [3]. This work proposes image processing and augmentation techniques and evaluates their performance on YOlOv3-SPP and YOlOv5-Large object detection models. We compose augmentation methods such as (3×4) *Grid Augmentation* and *Geometric Sequence* that successfully improve performance across multiple models, and their combination through transfer learning helps to achieve the highest overall mean average precision (mAP) 0.5 accuracy with YOLOv3-SPP. We also identify procedures where giving significantly more annotations [3] to the model during the training process does not improve and sometimes even impairs performance, and we verify the previous findings that augmentation techniques are effective when there are certain amount of images in the training dataset, which come from the same distribution as the validation set, otherwise in most of the cases we do not observe performance improvement. The augmentation procedures are evaluated across images with (1056×1056) input dimensions, which we regard as high resolution images for our experiments. High resolution images, however, translate to more computational resources and an incremental increase in the image size is also not guaranteed to improve performance. In spite of the significant amount of annotations in the images in the dataset that we use, there still exists a class imbalance, meaning the model may see an instance of a *car* class much more frequently than an instance of a *bicycle* class, resulting in training overfitting and decreasing the overall model accuracy during validation, which we aim to improve with our proposed augmentation methods. Our results show that (3×4) *Grid Augmentation* and (3×5) *Grid Augmentation with Transfer Learning* with random resampling methodology can successfully improve accuracy. As part of future work, instead of random sampling, we will explore *Grid Augmentation* variations with an oversampling methodology and specifically target the images containing imbalanced classes.

2. Methodology

2.1. Models

In this work, for conducting our experiments, we use two model variants, which use one-stage object detection paradigm. The models are YOLOv3 with Spatial Pyramid Pooling (YOLOv3-SPP) from the works of [4, 5, 6] and YOLOv5-Large [7]. To solve the problem of YOLOv2 backbone's low ability to do feature extraction and inability to make a full use of multi-scale local region features, [5] propose DC-SPP-YOLO (Dense Connection and Spatial Pyramid Pooling Based YOLO) to improve the accuracy of YOLOv2. They design a new spatial pyramid pooling introduced to collect and concatenate the multi-scale local region

[3] Annotation is the process of labeling images, which provides information to the computer vision model about the objects in the image. COCO bounding box annotations store (x-top left, y-top left, width, height) values of the given object in the image in a json format, while Pascal VOC annotations store (x-top left, y-top left, x-bottom right, y-bottom right) values of the object in an xml format.

features for more comprehensive learning. They also show that DC-SPP-YOLO is more accurate than YOLOv2. We use a combination of spatial pyramid pooling and YOLOv3 by [6] and a recently developed YOlOv5-Large model by [7] for these experiments. [7] mentions that YOLOv5-Large achieves an average precision (AP) 0.5 score of 66.5% on a COCO test-dev dataset. In spite of being the official YOLOv5 repository, the model is fairly new, and the official paper is still to be released.

2.2. Data

This work uses a subset of VisDrone-DET2019 [4] dataset as a starting point for the experiments, and later scales upon it. The original dataset includes 10 categories, which are *pedestrian, person, car, van, bus, truck, motor, bicycle, awning-tricycle and tricycle*. For these 10 categories, there are about 540,000 bounding boxes, 6471 training, 548 validation and 1580 testing images [8]. We select 1000 images having a height of less than 768 pixels from the original training dataset for the experiment, which are later readjusted to 1056×1056 input image dimensions and use it for conducting our image augmentation experiments. These images are later divided into 700 training, 200 validation and 100 test sets. The original 10 categories are reduced to 7, and our class list contains all default categories except *motor*, *awning-tricycle* and *tricycle*. By using image augmentation sequences and techniques, we later generate 700 new images equal to the number of images in our training dataset, and our final dataset includes 1400 training, 200 validation and 100 test images. We keep 11.7% of our total images for validation, compared to the original dataset, where the number of validation images is 6.3% of the total.

2.3. Image Transformations and Augmentations

We present three image augmentation sequences and name them *Blending Sequence, Blurring Sequence, Geometric Sequence*, and 2 new image augmentation techniques and name them (3×4) *Grid Augmentation*, and (3×5) *Grid Augmentation*, which are applied to the training dataset to generate 700 new images, and for the main experiment, we compare the performance of the augmentation techniques against the base results[5], or simply *No Augmentation* results. To test whether the obtained results can be further approved, we later experiment with a combination of sequences and grid augmentation for the model providing the best result across the experiments. The first three sequences are explored as part of the data-warping augmentations, while the grid augmentations are explored as part of random-resampling augmentation to help decrease class imbalance. Each augmentation sequence includes from five to seven augmenters [6] and is grouped according to its relevant augmentation type. Using such approach allows each image in our dataset to undergo a unique image transformation procedure obtained from each augmentation sequence. After the augmentation procedures, we mostly end up with training images, which may be a little complex in form and not always observed in real life, however, as it will be seen from the results, most of them do positively affect the performance, and some of them do it much

[4]VisDrone dataset is a large drone-based dataset containing images and videos particularly made for object-detection and object-tracking tasks.

[5]Base results are obtained from evaluating the models on the original 700 images without any augmentation, and the respective model is called a base model.

[6]Augmenter is a distinctive image transformation technique.

more effectively than the rest. Each augmenter from a sequence is selected with a certain probability, and in the end one new augmented image is generated per image in the original dataset. With a defined probability, depending on the augmentation sequence, the original image undergoes from zero to two image processing procedures for each sequence, the probabilities varying for each one. We leave a small chance that no augmenter is applied for a given image. The grid augmentations are stochastic, as they select images randomly, while *Blending, Blurring and Geometric* sequences are deterministic, so for a set of images they always produce the same results making the experiments reproducible.

2.3.1. Blending Sequence

Blending sequence includes five different blending augmenters. For each image, either one or two of the five available augmenters is selected. Each augmenter itself, has a 90 % probability of happening when selected. The reason that the augmenters are not assigned full probabilities of happening is to prevent them from altering the default dynamics and over damaging the image. This can be better conceptualized for the case when two augmenters are selected, this method giving a small chance that not both of them are applied all the time. If for a given image two augmenters are selected, there is a 1 % probability that none of them is applied, and 10% probability that at least one is not applied. The Blending Sequence includes *Alpha Frequency Noise Blending with a nearest upscale method, Alpha Mask Blending, Alpha Checkerboard Blending with a hue upscale method, Alpha Linear Gradient Blending* and *Alpha Simplex Noise Blending with a linear upscale method.*

Alpha Frequency Noise Blending uses frequency noise masks for blending two images. The alpha masks are sampled from varying frequency noises, and as we use the nearest neighbour upsampling, it mostly results in the creation of blobs with sharp edges. *Alpha Mask Blending* augmenter generates an (H, W) or (H, W, C) channel-wise mask for each image, where H is the height, W is the width and C is the number of channels of the image. The mask is later used to alpha blend pixels between the foreground augmenter branch and the background branch. Clouds are added as a part of this augmenter, as can be observed at the bottom right hand side of Fig. 1b. *Alpha CheckerBoard Blending* uses a checkerboard pattern for blending the images. $(R \times C)$ grid, where R is the number of rows and C is the number of columns, is placed on each image. For this experiment, R is 1, and C is a number uniformly drawn from the [1,4] interval. The value for the hue of images is randomly drawn from the [-80,80] interval. *Alpha Linear Gradient Blending* blends two images along a vertical gradient. Here the gradient is applied to a pooled-image, and the starting and ending coordinates of Y-coordinate is chosen at random. This randomness causes the gradient to increase either at the top or at the bottom [9]. *Alpha Simplex Noise Blending* detects edges inside the image, which are marked black and white, then uses simplex noise to alpha blend with the original image. Here we use a linear upscaling method, which creates rectangles having smooth edges. A sample blending transformation example can be seen in Fig. 1.

2.3.2. Blurring Sequence

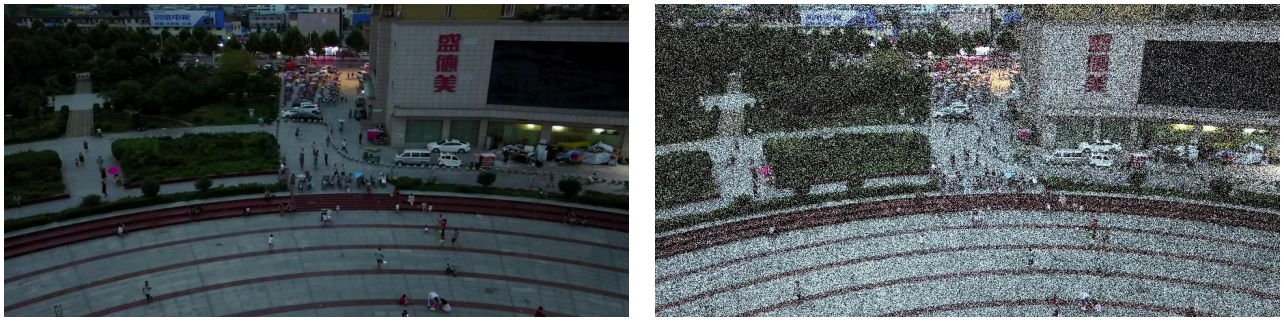
This sequence includes four blur-related augmenters and three contrast-related augmenters, and together they comprise the *Blurring Sequence*. For each image, one or two of the available augmenters are selected. As *Blurring Sequence* includes more augmenters, each augmenter has a 95% probability of being executed when selected (see Fig. 2). The augmenters are:

(a) Original image. (b) Transformed Image.

Fig. 1. *Blending Sequence* transformation.

Additive Gaussian Noise, Gaussian Blur, Motion Blur, Mean Shift Blur, Histogram Equalization, Sigmoid Contrast and *Linear Contrast*.

Additive Gaussian Noise adds Gaussian noise to the image. For each pixel, the noise is sampled from $F(0, s)$ where F follows a Normal Distribution, and s is sampled once per image. We choose s as 70. *Gaussian Blur* uses a Gaussian kernel for blurring the images. The standard deviation for the kernel is selected randomly from the range of [1,3] for each image. The *Motion Blur* augmenter applies a motion blur with a kernel size of 10×10 and a blur angle from the range of [-30,30], selected randomly per image. The *Mean Shift Blur* augmenter applies a pyramidic mean shift filter to each image. *Histogram Equalization* augmenter transforms the image into *RGB* color space, applies histogram equalization and the input channels are transformed back to the original color space. For the *Sigmoid Contrast* output, we use a (3,10) tuple. The multiplier for the sigmoid function is selected randomly from the interval of [3,10] per image, and the cutoff value, which defines the shift of the sigmoid function in the horizontal direction, is selected randomly from the range of [0.4,0.6] per image. Higher values result in darker pixels [9], and we select the range accordingly. *Linear Contrast* scales each pixel to $127+\alpha * (a - 127)$, where a is selected randomly from the range of [0.6,1.4] per image. The default range is preserved.



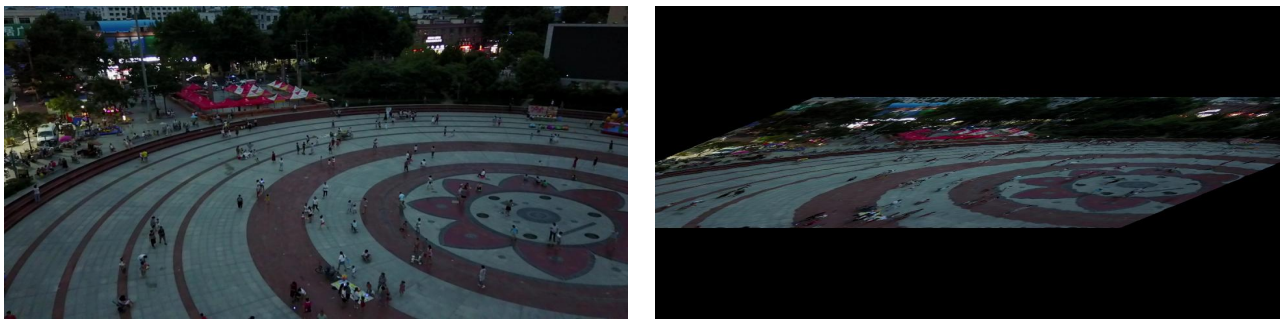
(a) Original image. (b) Transformed Image.

Fig. 2. *Blurring Sequence* transformation.

2.3.3. Geometric Sequence

For each image as before, either one or two augmenters get selected. For this sequence, each augmenter has a 93 % probability of being applied to the image, when selected. *The*

Geometric Sequence has the following augmenters: *Elastic Transformation, 90 Degree Rotation, Rotation from -30 to 30, Polar Warping, Shear Transformation in Y direction* and *Shear Transformation in X direction. Elastic Transformation* uses displacement fields and



(a) Original image. (b) Transformed Image.

Fig. 3. *Geometric Sequence* transformation.

moves pixels around the figure. The augmenter has an alpha parameter, and we pass a tuple of (20, 30), and each image is transformed with a random value from the range of the tuple. We specifically choose low values for the tuple to introduce pixel distortion. *90 Degree Rotation* rotates an image clockwise with a multiple of 90 degrees, resulting in various flipping transformations. *Rotation from -30 to 30* rotates an image with a randomly chosen value from the interval of [-30,30]. *Polar Warping* transforms an image into polar representation, which results in circular-warped patterns in the image. This transformation can be partially observed in subfigure 3b. *Shear Transformation in Y direction* applies a shear transformation across the Y axis, and *Shear Transformation in X direction* applies shear transformation across the X axis (Fig. 3b). The transformations intervals, from which the values are selected randomly for the transformation are [30,50] and [30,70], respectively. The values are chosen to replicate realistic viewing scenarios that a drone may be exposed to during a real-time flight. During a *Geometric Transformation*, however, some of the objects start to lie outside of the image and thus some of the bounding-box coordinates get dropped.

2.3.4. Grid Augmentations

2.3.5. (3×4) Grid and (3×5) Grid

[10], in their recent work on YOLOv4, use a new data augmentation technique first proposed by [6], which combines randomly chosen four images together during the model training process to increase image variability. They name this technique *Mosaic Augmentation* and demonstrate that it helps to increase the mAP accuracy score on COCO dataset.
We extend the idea of *Mosaic Augmentation* and propose two new data augmentation techniques and name them (3×4) *Grid Augmentation* and (3×5) *Grid Augmentation*, the latter being used with transfer learning during the model training, which combine 3 images horizontally and 4 and 5 images vertically, respectively. The combined images with their bounding boxes for (3×4) *Grid Augmentation* can be seen in Fig. 4 [7]. It is a common practice and sometimes a requirement to choose the image size of the deep learning network as a multiple of 16 or 32 pixels. For this reason, for the (3×4) *Grid Augmentation*,

[7] A visual example of (3×5) *Grid Augmentation* is not provided as it follows a transformation pattern similar to (3×4) *Grid Augmentation.*

each image is resized to (352×352), which is thought to be an optimal size considering the increase in width and height that we get after grid augmentation. Each image, thus, has a square shape. The images are then combined horizontally and vertically, resulting in (1056×1408) size images, including annotations of 12 randomly selected images, which are resized accordingly. For the (3×5) *Grid Augmentation*, a random batch of 3 images is selected once a time. The 3 images are resized to the minimum image height in the given batch, and the widths are re-adjusted according to the new image height. The results in the creation of horizontal images. A random batch of 5 horizontally combined images are then selected to be combined vertically. The images are now resized based on the minimum image width of the given batch, and the heights are adjusted accordingly. This method allows us to preserve the relative sizes of the images, as well as different widths and heights. These differences, however, are sometimes not visible, as most of the images have relatively similar shapes, and this may create grid augmentations, where each image in the grid may appear to be square in shape. For the (3×5) *Grid Augmentation*, we join images, which have undergone a *Blur Sequence* transformation, unlike (3×4) *Grid Augmentation*, where the images are selected from the subset of original Visdrone dataset. Grid augmentation methods significantly increase the number of small object in a given image, and most of the grid-image combinations may be realistically analogous to the real world, as the drone sees objects from various settings during a real flight, especially from higher altitudes, an effect obtained from grid augmenting transformation. Though technically the proposed technique allows us to combine more images both horizontally and vertically, we should be careful about the computational costs, because if the given image has a number of annotations on average, the image after grid transformation contains $a \times w \times h$ annotations on average, where w and h are the horizontal and vertical numbers of combined images.

2.3.6. Blurring Sequence and Grid Augmentation

For the model that provides the best mAP 0.5 score, we test an augmentation sequence, where instead of blurring the images then combining them into a grid-like image as done for (3×5) *Grid Augmentation*, we apply a *Blurring Sequence* augmentation to the whole grid-augmented image.

2.3.7. All Augmentation Sequences and Grid Augmentation

As a final experiment, we select 700 images that have been transformed using (3×4) *Grid Augmentation* and divide them into 3 equal proportions. We transform the first, second, and third proportions using *Blending Sequence, Blurring Sequence, Geometric Sequence*, respectively. The results for *Blending Sequence* and *Geometric Sequence* can be seen by looking at Fig. 5. For training the model, we use only the transformed 700 images, unlike the rest of the experiments, which use 1400 images.

Combining images into a grid-like form also combines all separate annotations into one. We provide a summary of augmentation sequences including the total number of annotations for each technique in Table 1. We observe that grid augmentation techniques significantly increase the amount of annotations in the dataset, most of them being generated from (3×5) *Grid Augmentation*, which, over the original dataset with no augmentation, contains 15.3 times more labels.

(a) (3 × 4) grid-augmented image.

(b) (3 × 4) grid-augmented image with bounding boxes.

Fig. 4. (3 × 4) grid-augmented image.

(a) *Blending Sequence* applied to a (3 × 4) grid-augmented image.

(b) *Geometric Sequence* applied to a (3 × 4) grid-augmented image.

Fig. 5. Augmentation Sequences applied to (3 × 4) grid-augmented images.

Table 1: Augmentation Sequences applied to the original 700 training images.

Augmentation Sequence	**N^o Bounding Box Annotations**	**Total N^o of Images**
No Augmentation	40309	700
Blending Sequence	40309	1400
Blurring Sequence	40309	1400
Geometric Sequence	37967	1400
(3×4) Grid	471233	1400
(3×5) Grid	618256	1400
Blurring Sequence and (3×4) Grid	471233	1400
All Sequences and (3×4) Grid	458044	700

[a] Blending, Blurring, Geometric Sequences, (3×4) Grid, (3×5) Grid are applied to the original 700 images.

[b] Blurring Sequence and (3×4) Grid, All Sequences and (3×4) Grid are applied to 700 images that have been (3×4) grid-augmented.

2.4. Transfer Learning

The trainings begin from a pretrained checkpoint for both models. YOLOv3-SPP uses *yolov3-spp-ultralytics* weights, and YOLOv5-Large uses *yolov5l* weights. Using pretrained backbones helps to speed up learning and gives more improved performance than training the models from randomly-initialized weights, however, we believe the performance can be further improved if we use weights that have been trained specifically on the drone-object detection footage, as the pretrained weights available for initialization work well with larger objects, but there is a significant performance reduction when evaluated on a small object detection task. For this reason, we conduct transfer learning based on the weights that we have obtained from our experiments and as the results will later show it helps to further improve performance and obtain a better model. We use the model's weights, which have been trained on the dataset without augmentation (700 images) as an initialization point for YOLOv3-SPP and YOLOv5-Large and the weights that have been trained on the dataset with (3×4) grid-augmented images as an initialization point for YOLOv3-SPP trained with Geometric Augmentation as part of the performance-improvement experiment with the best model.

3. Experiments and Results

3.1. Model Specifications

3.2. YOLOv3-SPP

YOLOv3-SPP trained on higher resolution images provides the better results for both experiments. The base model provides mAP 0.5 score of 39.7 %. We further improve this performance up to 46.2 % with (3×4) *Grid Augmentation*, which becomes the procedure providing the highest increase in mAP score amongst all the models without using transfer learning[8]. *Geometric Sequence* and *Blending Sequence* provide almost equal results in terms

[8]Without transfer learning means not using custom weights obtained from our base model.

Table 2: Parameters of YOLOv3-SPP and YOLOv5-Large.

Parameters	**YOLOv3-SPP**	**YOLOv5-Large**
Batch Size	2	2
Epoch Size	15	15
Channels	3	3
Validation Interval	1	1
Initial Learning Rate	0.001	0.01
Image Size	(1056×1056)	(1056×1056)

of mAP 0.5; 45.3% and 45.2%, respectively. We will later see that the *Geometric Sequence* and the pretrained weights of (3×4) *Grid Sequence* together provide the highest results obtained for this work. The other augmenter, the *Blurring Sequence* provides a result of 45.1%, however, the results of (3×5) *Augmentation with Transfer Learning* is almost just as good as the base model, while containing about 620,000 more annotations than the base model and using pretrained weights as initialization weights for training. This shows that extensive augmentation procedure is not guaranteed to improve performance. We observe very smooth learning and increase in precision by looking at Fig. 6. Summary results can be found in Table 3.

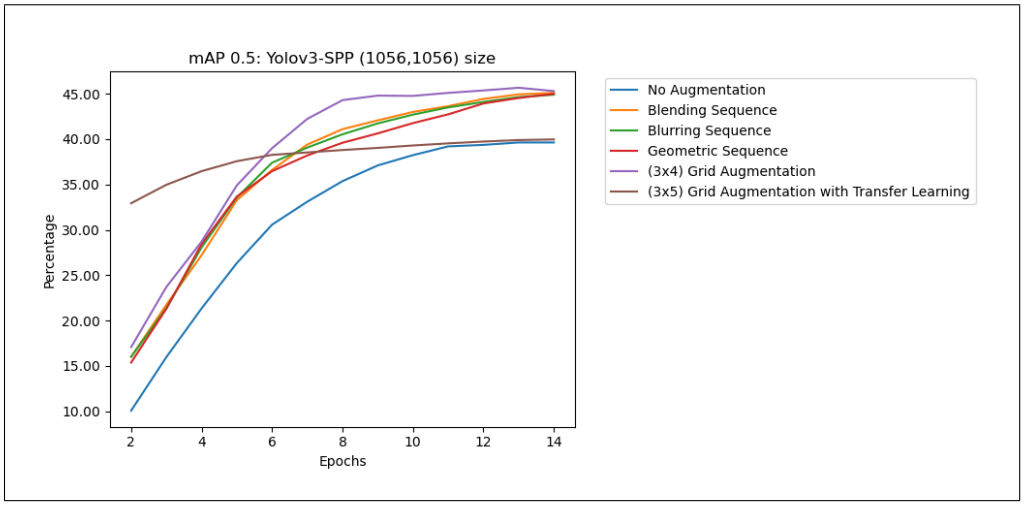


Fig. 6. mAP 0.5: Yolov3-SPP Results with (1056×1056) image size.

Table 3: Best Results Summary for YOLOv3-SPP: (1056×1056) size.

Rank	Model	mAP 0.5	F1 score
1	(3×4) Grid Augmentation	46.2%	48.9%
2	Geometric Sequence	45.3%	47.1%
3	Blending Sequence	45.2%	47.7%
-	No Augmentation	39.7%	41.2%

3.3. YOLOv5-Large

With YOLOv5-Large, we obtain an mAP 0.5 of 29.2%, achieved by (3×5) *Grid Augmentation with Transfer Learning*. Further results can be found in Table 4, showing that *Blending Sequence* and *Blurring* Sequence are important augmentation procedures for YOLOv5-Large with mAP 0.5 scores of 27.0 % and 26.3 %, respectively. We also observe an increase in accuracy provided by the augmentation sequences over the base model.

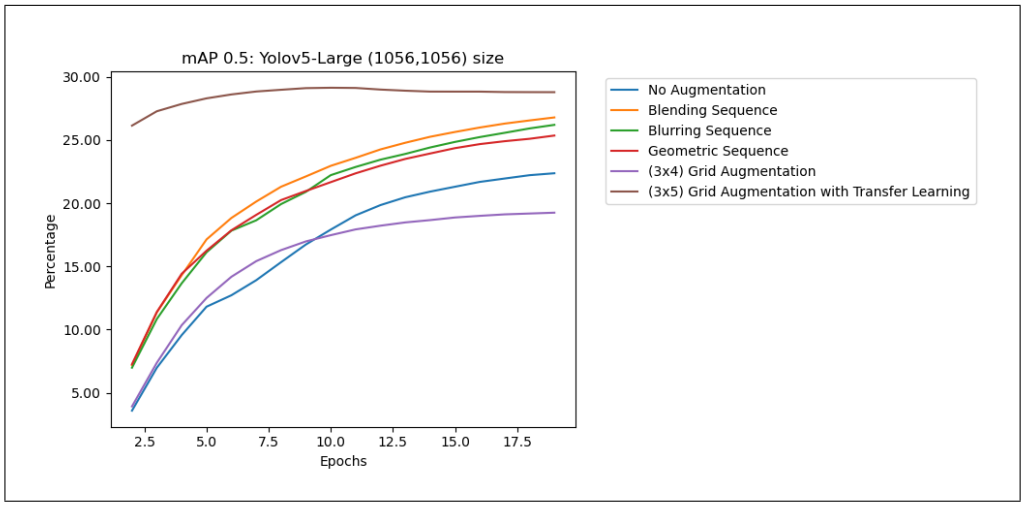


Fig. 7. mAP 0.5: YOLOv5-Large Results with (1056 × 1056) image size.

Table 4: Best Results Summary for YOLOv5-Large: (1056 × 1056) size.

Rank	Model	mAP 0.5	mAP 0.5:0.95
1	(3 × 5) Grid Augmentation with Transfer Learning	29.2%	14.0%
2	Blending Seqence	27.0%	14.6 %
3	Blurring Sequence	26.3 %	14.2 %
-	No Augmentation	22.5%	12.9%

4. Further Improvements

We conduct a few more experiments to see if the model performance can be further improved. We make one more experiment with YOLOv5-Large and 3 more experiments with YOLOv3-SPP. By training a model with *Blending Sequence* augmentation using pretrained weights from the base model training we are able to obtain the best model for YOLOv5-Large, with an mAP 0.5 score of 36.5%, which is a 14% improvement from the base model. We choose *Blending Sequence* as it is the second-best augmenter for YOLOv5-Large, in terms of mAP 0.5 score. For the YOLOv3-SPP model, we choose the weights obtained from (3 × 4) *Grid Augmentation* for YOLOv3-SPP and initialize it for our second-best *Geometric Sequence*-transformed dataset for training. We obtain mAP 0.5 score of 48.6%, which is a 2.4% improvement over our previous best score, and 8.9% improvement over our base model. We

also provide two examples where we train the models with the augmentation methods defined in 2.3.6., where we perform *Blurring Sequence* transformation on the dataset containing 700 original and 700 (3×4) *grid-augmented* images, and another experiment, where we take 700 training images, which have undergone a (3×4) *grid-augmentation* and apply *Blurring, Blending and Geometric Sequence* transformations, each augmenter transforming 1/3 of the images, as described in Section 2.3.7.. Moreover, for this augmenter, which was our most complex transformation, the mAP decreases by 5.9% compared to the base model. We observe that more annotations do not strictly increase accuracy. As a note, these final two augmentation experiments where we did not observe improvements contained no images from the original dataset, unlike the rest of the experiments. The experiments with the best overall results for YOLOv3-SPP and YOLOv5-Large can be observed from Fig. 8 and Fig. 9.

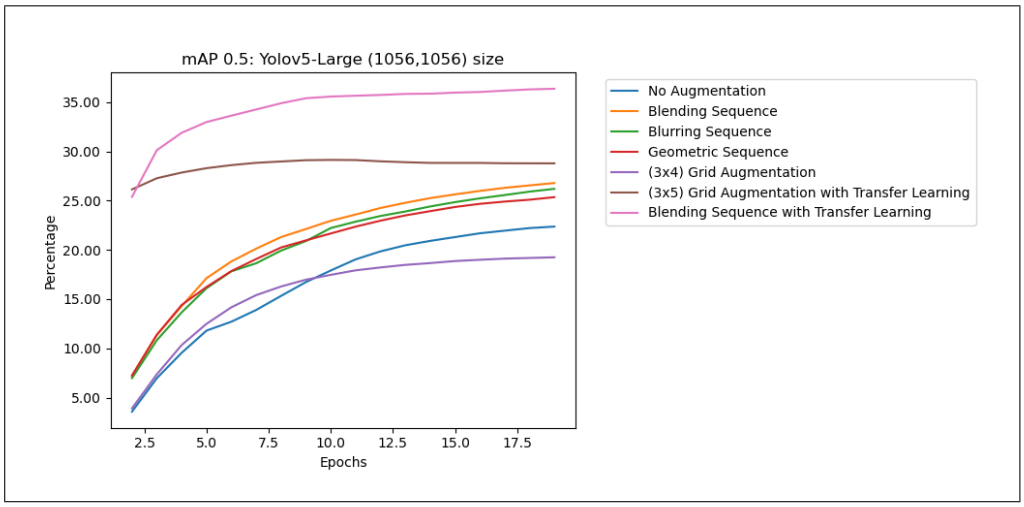


Fig. 8. mAP 0.5: YOLOv5-Large (1056×1056) size with improved performance.

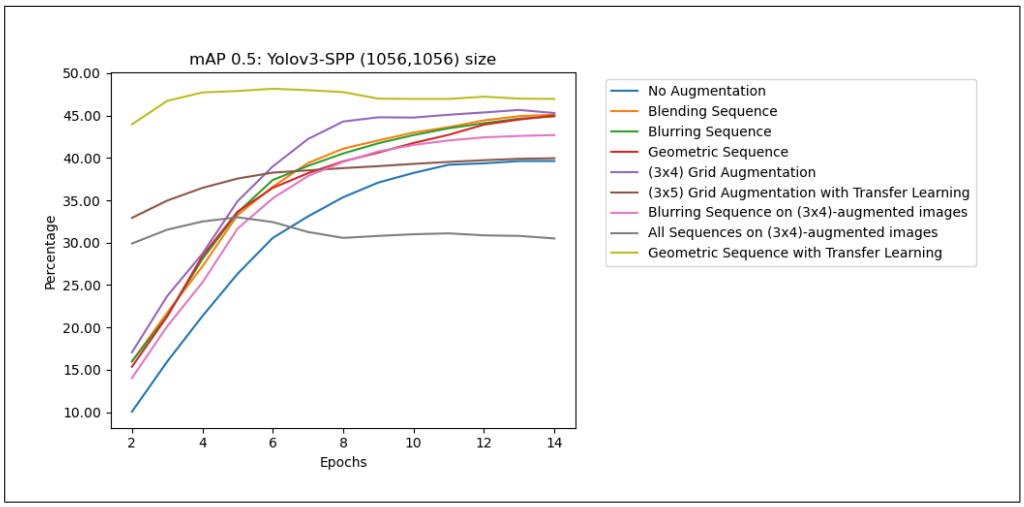


Fig. 9. mAP 0.5: YOLOv3-SPP (1056×1056) size with improved performance.

(a) Inference 1. (b) Inference 2.

Fig. 10. Inference results of YOLOv3-SPP.

5. Inference Results

From the inference results of the best YOLOv3-SPP model from Fig. 10a, we can observe that the model performs well on identifying almost all of the objects in the defined categories, however, we also see in Fig. 10b that there are pedestrians in the left hand part of the image that are not detected, so we identify room for improvement.

6. Conclusion

This work explored multiple-object detection task for drones using a component of VisDrone-DET2019 dataset. We constructed three-image-processing sequences each containing from 5 to 7 augmenters, and named them *Blending Sequence, Blurring Sequence, Geometric Sequence*. In an effort to reduce data overfitting with random resampling, we also developed grid augmentation techniques where we combined images in a custom-shape (3×4) square form and (3×5) form where the relative dimensions of the images were preserved after combination. We named these techniques (3×4) *Grid Augmentation* and (3×5) *Grid Augmentation*. We conducted multiple experiments where we tested how our augmentation techniques affected the performance of the models with an aim to identify the techniques that work best for the drone object detection task. We identified three augmentation procedures that help to improve the model performance the most, which are using *Geometric Sequence*, (3×4) *Grid Augmentation* transformations and (3×5) *Grid Augmentation with Transfer Learning*. We considered (3×4) *Grid Augmentation* as the overall best and *Geometric Sequence* as the second best augmentation method and the best one among all three sequences. We further conducted a few more experiments by using transfer learning from custom weights, and combining our sequences with grid augmenters in different forms for two of our best models; YOLOv5-Large and YOLOv3-SPP with (1056×1056) image size, achieving further increase in accuracy. We also showed that some augmentation techniques did not work well and resulted in performance deterioration, despite containing significant amount of annotations, as we saw when applying *Blurring Sequence* to the (3×4) grid-augmented dataset, and when applying all of our sequences to 700 (3×4) grid-transformed images. These datasets for training contained no images from the original dataset, and we can state that when training and validation datasets are not from the same setting, the augmentation procedures will most certainly not be successful. On the other hand, as a result of our further experiments, we obtained our overall best model by using the weights of the model trained with (3×4) *Grid Augmentation* for training a model with *Geometric Sequence*

augmentation, and improved the performance of the previous best result by 2.4% to mAP 0.5 score of 48.6% achieved with a YOLOv3-SPP model. The dataset sizes differ and the original dataset contains 3 more classes, so it is challenging to compare this setting to the original, however, we note that the winner of the VisDrone-DET 2019 challenge achieved an AP 0.5 score of 54.0%, and 47.98% was the tenth best score [8]. We take these results as benchmark results for future improvement.

References

[1] L Jangwon, J Wang, D Crandall, S Šabanovic and G Fox, "Real-Time, cloud-based object detection for unmanned aerial vehicles" *First IEEE International Conference on Robotic Computing (IRC)*, Taichung, Taiwan, DOI: 10.1109/IRC.2017.77, pp. 36 - 43, 2017.

[2] A. Carrio, C. Sampedro, A. Rodriguez-Ramos and P. Campoy, "A review of deep learning methods and applications for unmanned aerial vehicles" *Journal of Sensors*, https://doi.org/10.1155/2017/3296874, 2017.

[3] Zh. Wu, K. Suresh, P. Narayanan, H. Xu, H. Kwon and Zh. Wang, "Delving into robust object detection from unmanned aerial vehicles: A deep nuisance disentanglement approach", *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1201–1210, 2019.

[4] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement", *arXiv:1804.02767v1*, 2018.

[5] Zh. Huang and J. Wang, "Dc-spp-yolo: Dense connection and spatial pyramid pooling based yolo for object detection", *arXiv:1903.08589*, 2019.

[6] Glenn Jocher. Ultralytics yolov3. (2019). https://github.com/ultralytics/yolov3

[7] Glenn Jocher. Ultralytics yolov5. (2020). https://github.com/ultralytics/yolov5

[8] Dheeraj Reddy Pailla et al., "Visdrone-det2019: The vision meets drone object detection in image challenge results", *IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, DOI: 10.1109/ICCVW.2019.00030, 2019.

[9] A. Jung. imgaug. (2020). Online.Available: https://github.com/aleju/imgaug

[10] A. Bochkovskiy, C.-Yao Wang and Hong-Yuan Mark Lia, "Yolov4: Optimal speed and accuracy of object detection", *arXiv preprint arXiv:2004.10934*, 2020.

Submitted 22.07.2020, accepted 19.11.2020.

ԱԹՍ-ից ստացված առարկաների ճանաչման բարելավում պատկերի աուգմենտացիայի միջոցով

Կարեն Մ. Գիշյան

Բաթի Համալսարան
e-mail: karen.gishyan@bath.edu

Ամփոփում

Առարկաների հայտնաբերման ալգորիթմները, որոնք հիմնված են ցամաքային պայմաններում ստացված լուսանկարների վրա, վերջին տարիների ընթացքում բարելավվել և մեծ հաջողություններ են գրանցել այնպիսի բարդ տվյալների հավաքածուների վրա, ինչպիսիք են COCO-ն և PASCAL VOC-ը: Այս մոդելները, այնուամենայնիվ, այնքան էլ լավ արդյունքներ չեն գրանցում ԱԹՍ-ի վրա հիմնված առարկաների հայտնաբերման ժամանակ և հաճախ նկատվում է կատարողականության վատթարացում: Պատճառն այն է, որ մոդելների համար շատ ավելի բարդ է հայտնաբերել և դասակարգել առավել փոքր օբյեկտները, ի տարբերություն միջին և խոշոր չափի օբյեկտների, և դրոններից արված լուսանկարները հակված են դրսևորել շեղումներ տարբեր թռիչքային բարձրությունների, եղանակային պայմանների, տեսախցիկի անկյան և որակի պատճառով: Այս աշխատանքը ուսումնասիրում է երկու ժամանակակից առարկաների հայտնաբերման ճանաչման ալգորիթմների կատարողականությունը դրոններից արված լուսանկարների հայտնաբերման խնդրի դեպքում և առաջարկում նկարների աուգմենտացիաների եղանակներ բարելավելու մոդելների կատարողականությունը: Մենք կազմում ենք նկարների աուգմենտացիաների երեք շարքեր և առաջարկում երկու նոր միջոցներ՝ ուսումնասիրելով դրանց տարբեր համակցությունները մոդելների կատարողականության վրա: Աուգմենտացիաները գնահատվում են խորը ուսուցման երկու մոդելների վրա, որոնք ներառում են մոդելների ուսուցում բարձր որակի նկարներներով (1056 x 1056 պիքսելներ)՝ մոդելների ընդհանուր արդյունավետությունը դիտարկելու համար: Մենք համեմատում ենք աուգմենտացիոն տեխնիկաներ յուրաքանչյուր մոդելի համար: Մենք ցույց ենք տալիս երկու աուգմենտացիոն ընթացակարգեր, որոնք ավելացնում են առարկաների հայտնաբերման/ճանաչման ճշգրտությունը շատ ավելի արդյունավետ կերպով, քան՝ մյուսները և ստանում, ընդհանուր առմամբ, մեր լավագույն մոդելը՝ օգտագործելով փոխանցման ուսուցման մոտեցում, որտեղ փոխանցման կշիռները ձեռք են բերվում ուսուցանելով մոդելը մեր կողմից առաջարկված աուգմենտացիոն մեթոդով: Փորձարկումների արդյունքում մենք ստանում ենք մոդելի լավ կատարողականություն և ճշգրտություն և մատնանշում բարելավման ասպեկտներ որպես հետագա աշխատանքի մաս:

Բանալի բառեր՝ համակարգչային տեսողություն, խորը ուսուցում, պատկերի մշակում:

Улучшение обнаружения объектов БПЛА с увеличением изображения

Карен М. Гишян

Батский университет
e-mail: karen.gishyan@bath.edu

Аннотация

Алгоритмы обнаружения объектов на основе наземных изображений значительно улучшились за последние годы и дали хорошие результаты для сложных наборов данных изображений, таких как COCO и PASCAL VOC. Однако эти модели не так успешны, когда дело доходит до обнаружения объектов с помощью БПЛА, и обычно наблюдается ухудшение характеристик. Это связано с тем, что для моделей гораздо сложнее обнаруживать и классифицировать более мелкие объекты, чем объекты среднего или большого размера, а изображения с дронов подвержены трансляционным отклонениям, вызванными разной высотой полета, погодными условиями, ракурсами и качеством камеры. В этой работе исследуется эффективность двух современных алгоритмов обнаружения объектов в задаче обнаружения объектов с помощью дрона и предлагаются процедуры увеличения изображения для повышения производительности модели. Мы составляем три последовательности увеличения изображения и предлагаем два новых метода увеличения изображения, а также дополнительно исследуем их различные комбинации на характеристиках моделей. Аугментеры оцениваются для двух моделей глубокого обучения, которые включают обучение модели с изображениями с высоким разрешением (1056 x 1056 пикселей), чтобы оценить их общую эффективность. Мы предоставляем сравнение методов увеличения для каждой модели. Мы определяем две процедуры дополнения, которые повышают точность обнаружения объектов более эффективно, чем другие, и получаем нашу лучшую общую модель с использованием подхода с переносом обучения, при котором веса для переноса получаются в результате обучения модели с помощью предлагаемой нами техники увеличения. В конце экспериментов мы достигаем надежных характеристик и точности модели и определяем аспекты улучшения как часть нашей будущей работы.

Ключевые слова: компьютерное зрение, глубокое обучение, обработка изображения.