Exponential data augmentation methods for improving YOLO performance in computer vision tasks*

Yurii Myshkovskyi[1,†], Mariia Nazarkevych[1,2,*,†], Victoria Vysotska[1,2,†], and Rostyslav Yurynets[1,†]

[1] Lviv Polytechnic National University, 12 S. Bandery str., 79000 Lviv, Ukraine
[2] Ivan Franko University of Lviv, Lviv, 1 Universytetska str., 79007 Lviv, Ukraine

Abstract

This article examines data augmentation methods in the task of image recognition, specifically introducing the exponential augmentation approach to enhance the performance of deep neural networks, particularly YOLO, in object detection tasks. The proposed methodology is based on the sequential and repeated application of various transformations, including horizontal and vertical flipping, 90° rotation, Gaussian Blur, brightness and contrast adjustment. This approach ensures exponential dataset growth and significantly increases the diversity of training data, which is critical for improving the model's generalisation capability. Experimental results demonstrate that applying exponential augmentation leads to a significant improvement in detection performance, as indicated by increased mean Average Precision (mAP), Precision, and Recall, even when the initial dataset is limited. Additionally, the integration of the proposed approach with other effective augmentation techniques, such as Mosaic and MixUp, has been explored. The results indicate that combining exponential augmentation with these methods leads to more robust models that can better recognise objects under different lighting conditions, viewpoints, and noise levels. Beyond accuracy analysis, the study also investigates the impact of exponential augmentation on training stability, including the convergence speed of gradient descent and resistance to overfitting. It is shown that multiple data enrichment cycles allow neural networks to adapt more efficiently to challenging conditions and reduce the likelihood of memorising only specific examples from the training set. The proposed method can be particularly useful in computer vision tasks with limited or imbalanced datasets, as well as in scenarios where improving model accuracy is required without significantly increasing computational costs. The obtained results confirm that exponential augmentation is a promising approach for enhancing the performance of YOLO and other modern object detectors in complex image recognition scenarios.

Keywords

exponential augmentation, YOLO, object detection, computer vision, small datasets

CEUR-WS.org/Vol-4145/paper8.pdf

1. Introduction

Modern computer vision methods, in particular the YOLO (You Only Look Once) architecture, have become widespread in object detection and classification tasks. They have been successfully applied in various fields, including autonomous driving, video surveillance systems, and robotics. Despite significant progress in improving deep learning models, the diversity and volume of training data remains a critical issue. Insufficient number or unrepresentative distribution of images in the dataset can lead to a decrease in the accuracy and reliability of object detection, especially in difficult shooting conditions (changes in angles, lighting, noise, etc.).

One of the most common approaches to dealing with data limitations is augmentation, which is an artificial increase in the volume and diversity of the dataset using geometric and colour transformations. However, most existing methods involve either a simple random application of transformations or a limited set of operations, which does not always guarantee a significant improvement in the quality of training. In addition, in many tasks, it is crucial to ensure that all

* CPITS-II 2025: Workshop on Cybersecurity Providing in Information and Telecommunication Systems, October 26, 2025, Kyiv, Ukraine
* Corresponding author.
† These authors contributed equally.
yurii.i.myshkovskyi@lpnu.ua (Y. Myshkovskyi); mariia.a.nazarkevych@lpnu.ua (M. Nazarkevych); victoria.a.vysotska@lpnu.ua (V. Vysotska); rostyslav.v.yurynets@lpnu.ua (R. Yurynets)
0009-0004-0051-026X (Y. Myshkovskyi); 0000-0002-6528-9867 (M. Nazarkevych); 0000-0001-6417-3689 (V. Vysotska); 0000-0003-3231-8059 (R. Yurynets)
© 2025 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings ceur-ws.org ISSN 1613-0073
published 2025-12-29

image variants are balanced so that the model learns on the full range of possible situations and avoids overfocusing on certain types of data.

To address these challenges, an exponential data augmentation method for YOLO is proposed, which consists of a step-by-step and consistent expansion of the training set of images. At each step, transformations are performed (horizontal and vertical reflections, 90° rotation, blurring, etc.), and the resulting new images are also subject to the following augmentation stages. This approach allows us to exponentially increase the number of different examples and potentially significantly improve the generalisability of the model. In addition, the use of more advanced techniques (Mosaic, MixUp, etc.) at the final stages further enhances the result by combining several images into one and training the network to recognise objects in unusual, artificially created scenes.

The scientific novelty lies in the creation and experimental confirmation of the effectiveness of a step-by-step "exponential" expansion of training data, which involves the sequential application of transformations to already augmented images. This makes it possible to cover a variety of possible angles, lighting and distortions more evenly and comprehensively. Unlike traditional random or one-step augmentations, the proposed approach generates a much wider sample of training examples, increasing the generalisability of YOLO and thus providing better detection results in real-world applications.

Thus, the relevance of the proposed topic is due to the growing need to improve the reliability and accuracy of computer vision systems, especially in the context of limited or unbalanced training sets. The proposed exponential augmentation approach meets modern challenges, allowing to effectively expand the space of training examples and increase the robustness of models to input data variability. This makes the study a significant contribution to the development of applied deep learning and is practically relevant for a wide range of applications where the accuracy of object detection is critical.

2. Analysis of the latest research and publications

The original idea of a holistic (one-step) approach to object detection was proposed in the work of YOLO (You Only Look Once) [1]. This model proved that a convolutional neural network can directly predict the coordinates and classes of objects without sequentially dividing the task into feature extraction and classification. The next important step was the release of YOLOv4 [2], which paid increased attention to both the detector architecture and augmentation methods such as Mosaic. Thanks to this combination, the COCO set managed to achieve a balance between accuracy of ~43.5% mAP and speed of ~65 FPS, which was a significant improvement over previous versions of YOLO.

Further development in this direction has taken place in Scaled-YOLOv4 [3]. This paper demonstrates the ability to scale the model both towards lightweight versions (YOLOv4- tiny with mAP of about 20–25%) and advanced versions (YOLOv4-large, exceeding 55% mAP). This provides a flexible choice of the trade-off between Precision, Recall, mAP and performance. According to the authors, augmentation plays an important role in these achievements, as it increases the variety of training examples and reduces the risk of model overfitting.

In the context of the variety of augmentation methods, the Albumentations library stands out [3]. It covers more than 70 different transformations–from basic (shifts, reflections, rotations) to complex (CutOut, GridDistortion, etc.). A significant advantage of Albumentations is its optimisation for OpenCV and NumPy, which allows for faster processing of large image sets. More specialised compositional techniques include MixUp [4], which linearly mixes pixels of two examples, and CutMix [5], which cuts random parts of one image and pastes them into another. In classification tasks, both of these methods have demonstrated an increase in generalisability; however, in detection, their effectiveness requires more fine-tuning so that the model does not lose context for object localisation.

A separate area is approaches to synthesis or random distortion of a part of the image. For example, Random Erasing [6] paints over random fragments, simulating the conditions of partial

shading or frame damage. The Copy-Paste method [7] offers to "cut" entire objects from one image and paste them into another, which is especially useful for increasing the number of samples of complex scenes. Such techniques further enhance the ability of models to recognise objects under non-standard conditions and add examples to the training set that are not present in the original data.

In a series of review papers [8] and [9], researchers provide an overview of modern augmentation methods for deep learning, emphasising the importance of this process in the context of limited or non-standard datasets. The authors emphasise that the right augmentation strategy increases mAP and Precision, as the model is able to "see" more variations in potential objects and background situations.

In addition, in the works "AutoAugment: Learning augmentation policies from data" [10] and "AutoAugment: Learning augmentation strategies from data" [11] in the context of AutoAugment demonstrated that the automatic search for optimal augmentation policies can significantly improve the accuracy of detectors. For example, in experiments with the RetinaNet model on COCO, mAP increased by about 2–3% only due to a carefully selected combination of transformations, without any changes in the architecture or hyperparameters of the optimiser. This approach is particularly useful when the researcher is unable to manually search through all augmentation options.

At the same time, [12] demonstrated that knowledge distillation technologies can efficiently transfer high-level representations from larger networks to smaller ones while maintaining competitive accuracy. This approach can be combined with advanced augmentation in the future to create more compact yet robust detectors that are resistant to various distortions.

Therefore, the study of the method of exponentially increasing the diversity of the training set is a logical step, since the analytical data of [5, 13, 14], "AutoAugment: Learning augmentation policies from data" [15] and [16] show the ability of augmentation to significantly improve the mAP, Recall and Precision of models in computer vision tasks. This approach, when properly configured with sequential transformations, can significantly improve results even on relatively small datasets, expanding the range of possible scenarios and reducing the risk of overfitting.

The object of the study is the process of training YOLO networks in object detection tasks, including data preparation, model parameter optimisation, and the general YOLO architecture for different types of images.

The subject of the study is the method of exponential data augmentation, which includes multiple geometric and colour transformations of images, as well as their combination (Mosaic, MixUp) to improve the quality of training and generalisation of YOLO networks.

The aim of this study is to develop, substantiate and experimentally test the method of exponential data augmentation to improve the accuracy and generalisability of YOLO models in object detection tasks. The study is aimed at identifying the impact of gradual sample expansion using multiple augmentations on the quality of detector training, in particular, on the mAP (Mean Average Precision), Precision, and Recall indicators. Achieving the research goal involves solving the following tasks:

To study the effect of exponential augmentation on the efficiency of YOLO learning by conducting a series of experiments with different sample expansion configurations.

Comparative analysis of the effectiveness of the developed method with basic or random augmentation strategies

Study of the impact of the proposed methodology on the stability of the learning process—in particular, analysis of the rate of convergence of the gradient descent and the model's resistance to overtraining.

It is expected that the proposed exponential augmentation method will significantly improve the accuracy of YOLO detection without the need to expand the original dataset. In addition, its application can be especially useful for computer vision tasks where there are limitations on the number of available training images or where the dataset contains an imbalance of classes.

The results of this research can be applied in a wide range of real-world scenarios, including video surveillance, autonomous driving, military and industrial applications where the accuracy and robustness of the object detector to changing shooting conditions are important [17].

The exponential increase in the training dataset through the sequential application of various transformations is a logical development of the ideas of classical augmentation, which has long been proven effective in improving the accuracy and robustness of computer vision models [2, 3]. The essence of the exponential approach is that the results of the previous step (new images) become the "input" to the next stage of augmentation, which leads to a geometric (exponential) increase in the total number of examples.

2.1. The concept of data augmentation

Classical random augmentations (rotations, shifts, brightness changes) often do not cover the full range of spatial and colour distortions [2], while their sequential combination significantly expands the possible variations. At the same time, the more unique images the model sees, the lower the risk of "remembering" a particular sample, and thus the higher the generalisability, which is confirmed by a number of studies within YOLO [12, 16]. In addition, exponential augmentations can be applied to any dataset, do not require complex setup, and consist of basic transformations (flip, rotate, blur, etc.) that can be easily implemented using libraries such as Albumentations [5].

2.2. Step-by-step implementation of exponential data augmentation

The first step is a simple copying of the original images and YOLO label files from the dataset_converted/ directory to the dataset_converted_augmented/ directory, while maintaining the division into train, valid and test. At this stage, no transformation is performed—only a basic set of files is created, on which augmentation will be performed later. This approach is consistent with the practice of researchers first saving the "clean" original data and then working with a copy of it [13].

With a horizontal flip (A.HorizontalFlip), the goal is to mirror the images from left to right and increase the set by a factor of 2: each image is read, a flip is performed, and the image is saved with the _hflip suffix (labels are processed in the same way), so the number of images is doubled. In the case of a vertical flip (A.VerticalFlip), the method is identical, but it uses top-down mirroring and reads all files (including those that have already been flip-flopped), which causes another doubling (4× the original number). Rotate by 90° (A.Rotate(limit=(90, 90))) changes the image orientation and allows you to get new angles: all images are rotated by 90° clockwise and saved with the _rot90 suffix, creating 8× the original volume. Finally, Gaussian Blur (A.GaussianBlur(blur_limit=(37, 37))) is designed to simulate various shooting conditions (out-of-focus or camera shake) (Buslaev, A. et al., 2020), implemented by passing each image through a Gaussian blur filter (kernel up to 37×37) and saving the result with a _blur file, which once again doubles the total size (16× of the original). It's important to note that at each step, the transformations are performed on all images, including those that appeared at the previous stage. Thanks to this, the total number of examples grows exponentially rather than linearly.

After the "exponential" phase is complete, an additional pass is performed on the current 16× images, in which RandomBrightnessContrast (p=0.2), RandomGamma (p=0.2), and CoarseDropout (p=0.5) are applied with a certain probability. Each image receives only one additional copy (depending on whether the transformations "worked"), which roughly doubles the total number compared to the folder after the previous steps. In the last step, the entire set is scanned again, and two advanced augmentations known as Mosaic [1] and MixUp are applied with probability mosaic_prob=0.3 or mixup_prob=0.3. Mosaic randomly selects 4 images (including the current one) and places them in a 2×2 grid with a given input_size, adjusting the bounding boxes to account for the offset, which helps the model to "see" composite scenes with different types of objects [7]. MixUp, in turn, selects two images and mixes them linearly using the formula

$$\text{MixedImage} = \alpha \cdot \text{Image}_1 + (1-\alpha) \cdot \text{Image}_2, \quad (1)$$

where α is a random variable from the Beta(0.5, 0.5) distribution, and merged bounding boxes reduce the risk of overlearning by increasing the variety of examples. Although Mosaic and MixUp do not increase the number of images as aggressively as flips or rotations, they further expand the set of possible scenes and angles.

The first step involves copying the original N images to a new folder without changes, resulting in N images in each sample. Next, exponential augmentation takes place: four operations (horizontal flip, vertical flip, 90 degree rotation, and GaussianBlur with probability $p=1.0$) are performed sequentially on all images in the folder, doubling the number each time. First, the horizontal flip yields $2N$ instead of N, then the vertical flip doubles $2N$ ($4N$ in total), the 90-degree rotation generates $8N$, and GaussianBlur brings the total number to $16N$. At the stage of random augmentation "in one pass" (all_at_once_augmentation function), each of the $16N$ images receives another copy, which again doubles the number to $32N$. Then, for each of the $32N$, the probability of applying Mosaic augmentation is 0.3, and MixUp is also 0.3; each of them (if triggered) generates one new image, and if both are triggered, two new images are obtained for one original. With a probability of 0.49, no additional images are created, with a probability of 0.42, only one new image is created, and with a probability of 0.09, two new images are created. On average, 0.6 new images are generated per input image, i.e. $32N + 0.6 \times 32N \approx 51.2$, although the actual value can be from $32N$ (when neither augmentation is "triggered") to 96N (if both are activated for each image). Examples of the obtained images are shown in Figure 1.

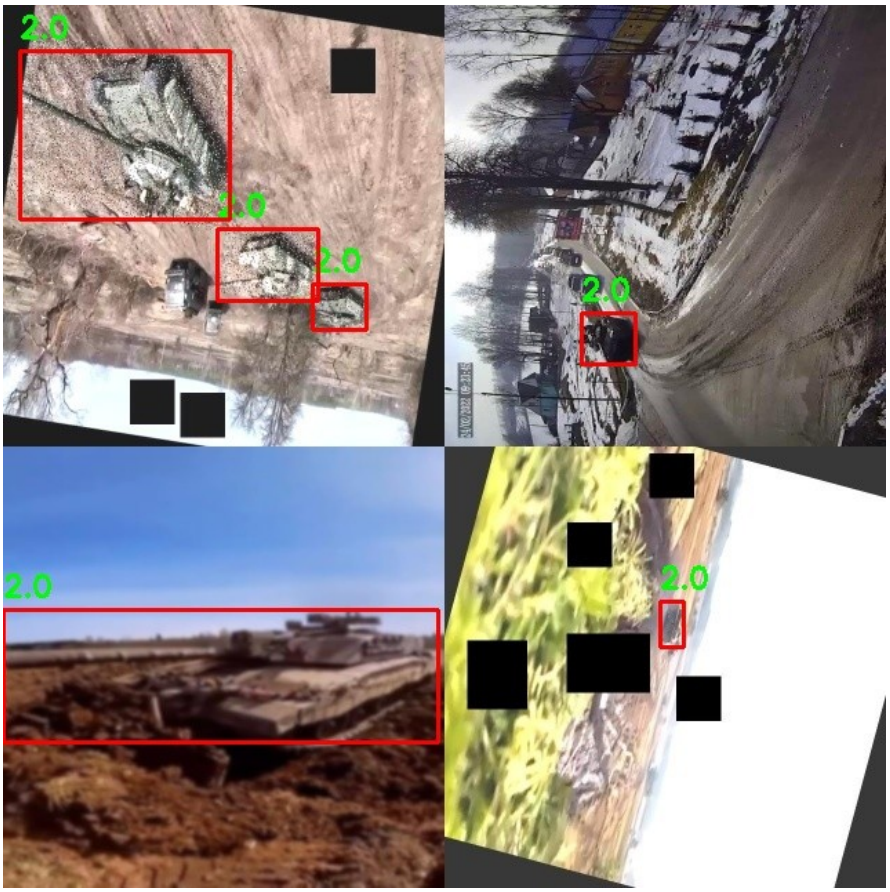

Figure 1: Examples from the resulting segmented dataset

2.3. Transform labels during geometric transformations

It is critical to correctly convert the coordinates of bounding boxes (labels) during all operations that change the geometry of the image. If a flip or rotation is performed without proper label correction, the detector will receive incorrect training data.

In the YOLO format, each label is described by 5 numbers:$(\text{class}, x, y, w, h)$, where x, y are the coordinates of the centre of the object, w, h are the width and height of the bounding box, normalised by the size of the image (i.e. in the range [0, 1]). Below are the transformation formulas.

Horizontal flip: When you mirror an image from left to right (horizontal flip), the point (x,y)(x,y) (x,y) goes to

$$x' = 1 - x,\ y' = y,\ w' = w,\ h' = h. \tag{2}$$

Accordingly, the width and height of the object do not change, as the image is simply displayed.

Vertical flip: Mirroring from top to bottom changes the y-coordinate:

$$x' = x,\ y' = 1 - y,\ w' = w,\ h' = h. \tag{3}$$

Rotate by 90° (clockwise): During a 90° rotation, the coordinates (x, y) are transformed using the formula:

$$x' = y,\ y' = 1 - x,\ w' = h,\ h' = w. \tag{4}$$

The width and height w and h are also interchanged here.

Photometric transformations (blur, contrast, etc.): If the augmentation only changes pixels (brightness, contrast, blur, fill, fragmentation), without affecting spatial dimensions, then the bounding boxes remain unchanged:

$$x' = x,\ y' = y,\ w' = w,\ h' = h. \tag{5}$$

Label correction for Mosaic: For Mosaic (or similar "collages", the logic is more complex, namely the transition to absolute coordinates: If the image is reduced or enlarged to input_size×input_size then:

$$\begin{aligned} x_{\text{abs}} &= x \cdot input_{\text{size}}, \\ y_{\text{abs}} &= y \cdot input_{\text{size}}, \\ w_{\text{abs}} &= w \cdot input_{\text{size}}, \\ h_{\text{abs}} &= h \cdot input_{\text{size}}. \end{aligned} \tag{6}$$

When placing 4 images in a 2×2 matrix, each image has an offset O_x, O_y for its quadrant:

$$x'_{\text{abs}} = x_{\text{abs}} + O_x,\ y'_{\text{abs}} = y_{\text{abs}} + O_y. \tag{7}$$

The resulting collage (of size 2·input_size×2·input_size) is usually reduced back to input_size×input_size. Accordingly:

$$\begin{aligned} x_{\text{final}} &= \frac{x'_{\text{abs}}}{2 \cdot input_{\text{size}}}, \\ y_{\text{final}} &= \frac{y'_{\text{abs}}}{2 \cdot input_{\text{size}}}, \\ w_{\text{final}} &= \frac{w_{\text{abs}}}{2 \cdot input_{\text{size}}}, \\ h_{\text{final}} &= \frac{h_{\text{abs}}}{2 \cdot input_{\text{size}}}. \end{aligned} \tag{8}$$

Label correction for MixUp: For MixUp (or similar techniques like CutMix [5], both images are usually pre-scaled and overlaid on top of each other. The pixels are summed using the formula (1), and the box coordinates are simply merged:

$$\text{AllBoxes} = \text{Boxes}_1 \cup \text{Boxes}_2. \quad (9)$$

Therefore, each geometric transformation requires a strict correspondence between the transformed image and its annotation. Errors in the implementation inevitably lead to a deterioration in the detector's accuracy

2.4. YOLO model

YOLO (You Only Look Once) [1] is one of the most popular deep learning architectures for real-time object detection tasks. The main idea behind the model is a one-step approach: the image is divided into a grid, and for each cell, the network predicts the object's coordinates, class, and confidence in its presence. This allows for significant image processing speeds while maintaining a high level of accuracy, which is especially valuable for real-time applications.

In the basic YOLO implementation, the input image is scaled to a fixed size (e.g. 416×416 or 640×640 pixels) and fed to a convolutional neural network consisting of several feature processing units. The architecture uses a backbone (e.g., CSPDarknet in YOLOv4/YOLOv5 versions), a neck (PANet or FPN), and a head to predict boundaries and object classes.

One of the key advantages of YOLO is that it performs object detection in just one run of the image through the model, unlike two-step approaches (such as Faster R-CNN) that first generate candidate regions and then classify them. This results in outstanding performance and compactness, allowing YOLO to be used even on devices with limited computing resources.

Today, there are several versions of YOLO, from the original YOLOv1 to the modern YOLOv8. This study utilized the YOLOv8S (small) version, which provides a good balance between accuracy and performance, making it suitable for training with augmented datasets even on medium-performance systems. YOLOv8S has an improved architecture compared to previous versions, including the use of a modular structure and block optimisation for more efficient spatial feature learning. The model is adapted to accept data in the YOLO markup format (.txt files with normalised rectangle coordinates), which facilitates integration with various image sets and automated augmentation pipelines. YOLO also supports training using advanced strategies such as Mosaic, MixUp, and in this case, exponential augmentation, which allows us to significantly increase the number of training examples and improve detection results on test samples.

With its combination of speed, accuracy and flexibility, YOLO is an ideal platform for experimentally investigating the effect of augmentation on the performance of an object detector.

2.5. Research metrics

According to [8] and [11], one of the key metrics for evaluating pattern recognition systems is Accuracy, which shows the ratio of correctly identified examples to their total number. An important addition is the error matrix, which helps to visualise how often the model confuses classes with each other. The Precision parameter characterises the quality of the predicted positive alarms, while Recall shows how well the model finds positive samples among all available ones. The F-measure, in turn, is a generalised metric that balances Precision and Recall values, allowing for a comprehensive assessment of the algorithm's performance.

2.6. Results

The proposed method of exponential augmentation, which uses basic flips, rotations, blurring, as well as more complex techniques (Mosaic, MixUp), allows to significantly increase the variety of training examples. Unlike single-stage or random augmentation, where transformations are applied only once, the consistent "multiplication" of the set increases the model's chances of "seeing" a wide variety of scenes, angles, and shooting conditions.

This methodology is consistent with the findings of the studies [1], [2], and [12], which emphasise the importance of multidimensional augmentation during YOLO training. Correct label correction is an essential component for maintaining detector accuracy, which has been repeatedly emphasised in augmentation work [5], [13], [18]. The following sections will demonstrate the experimental results of the described scheme and compare it with simpler (one-step) augmentation strategies.

3. Experiment and results

To evaluate the impact of exponential augmentation on the detector accuracy, a number of experiments were conducted with YOLOv8s, based on the ideas of YOLO [1, 9, 12]. All input images were scaled to 640×640, and training continued until the gradient descent stopped producing tangible progress. First, we considered a scenario without augmentation, where the model was trained for 75 epochs with a batch size of 16, using 5000 images for training and validation, and limited to basic preprocessing (normalisation, scaling). Next, we tested a configuration with exponential augmentation using stepwise transformations (horizontal/vertical flip, 90-degree rotation), as well as All-at-One random transformations, Mosaic and MixUp [1, 4, 12]; in this case, the training lasted 22 epochs with a batch of 64 and covered a total of 220,000 images for training and validation. Importantly, in order to select the best model, the validation sample was also significantly expanded with similar augmentations, so the detector was evaluated on a much larger range of scenes and angles than in the baseline "no augmentation" mode.

3.1. Results without augmentation

The YOLOv8s model, trained without exponentially increasing the dataset, demonstrates the following features:

Precision and recall averaged 0.88–0.91 depending on the class, and mAP@50 was around 0.90.

The normalised error matrix indicates about 0.87 correct classifications for the bmp class, about 0.89 for btr and about 0.95 for tank.

At the same time, a significant number of objects were confused with the background, especially in cases of partial visibility or fragments of equipment.

Accuracy for all classes (including background) reached approximately 0.85–0.86, and the F1-measure ranged from 0.85–0.88.

Therefore, the baseline scenario without augmentation generally successfully recognises most of the key classes, but the model still confuses some related objects (bmp—btr) or misidentifies background as technique (and vice versa).

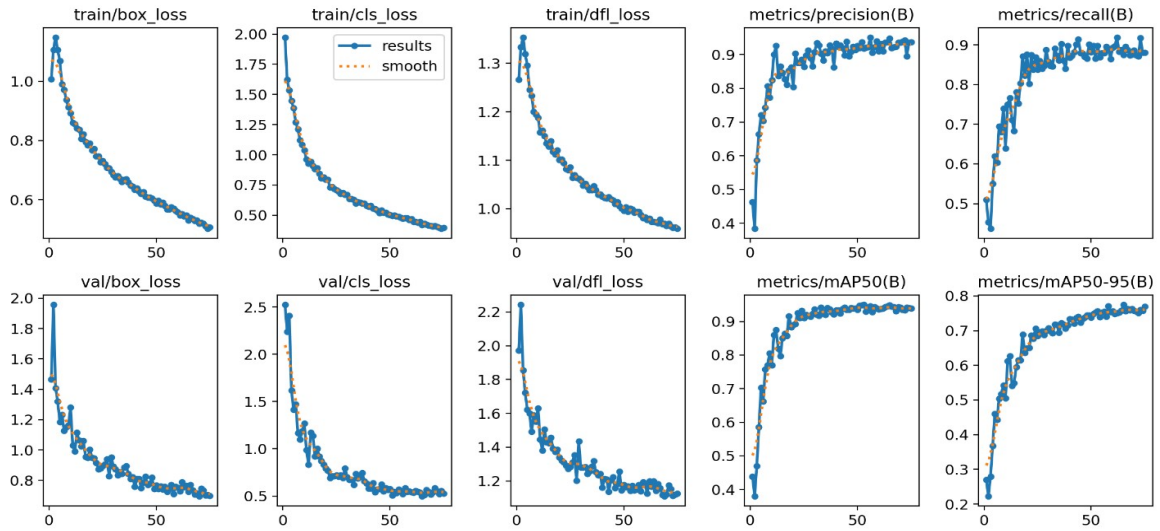

Figure 2: Experimental results without data augmentation

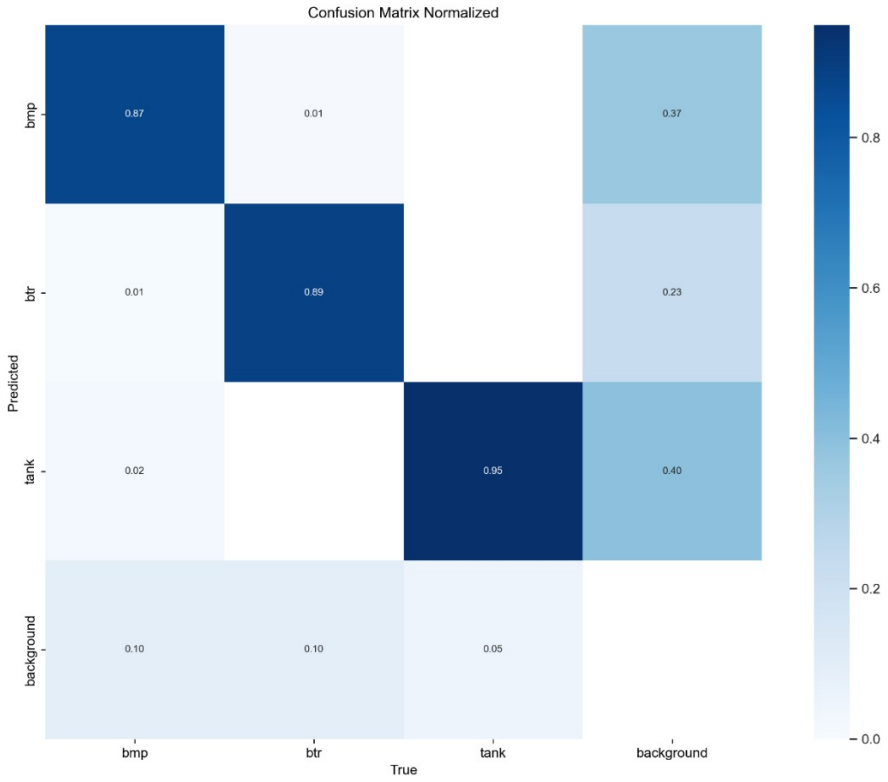

Figure 3: Error matrix for the experiment without augmentation data

3.2. Results with exponential augmentation

By applying step-by-step augmentation (flips, rotations, GaussianBlur, Mosaic, MixUp, etc.), the training set grew tenfold, and the validation set was also augmented (proportionally, with the same transformations). This allowed the model to see a much wider range of angles, lighting, and background scenes.

Precision and recall increased to 0.90–0.91 and 0.86–0.88, respectively, which confirms a more balanced recognition of all classes.

The mAP@50 reached about 0.92 (versus ~0.90 without augmentation), and the mAP@50-95 was about 0.74.

The error matrix shows approximately 0.86 correct detections for bmp, 0.93 for btr and 0.97 for tank. However, the proportion of confusion with background has decreased, although it remains (around 0.24–0.45) depending on the class.

Accuracy increased by 2–3% on average, and F1-measure by 1–2% depending on the class.

Thus, the exponentially increased data contributed to improved results, with a particularly noticeable reduction in errors between related classes and some reduction in false positives for background as a technique. Similar conclusions about the benefits of augmentation are made in [2], [14], and [15], where it is emphasised that a balanced increase and diversity of the sample significantly increase the robustness of the detector.

Comparison of the models without and with augmentation shows an increase in Accuracy from about 0.91 to 0.92–0.93, higher mAP values (by 2–3% depending on the IoU threshold), improvements in Precision and Recall by 1–3 percentage points, and a decrease in the frequency of confusion with background and between certain classes (bmp/btr).

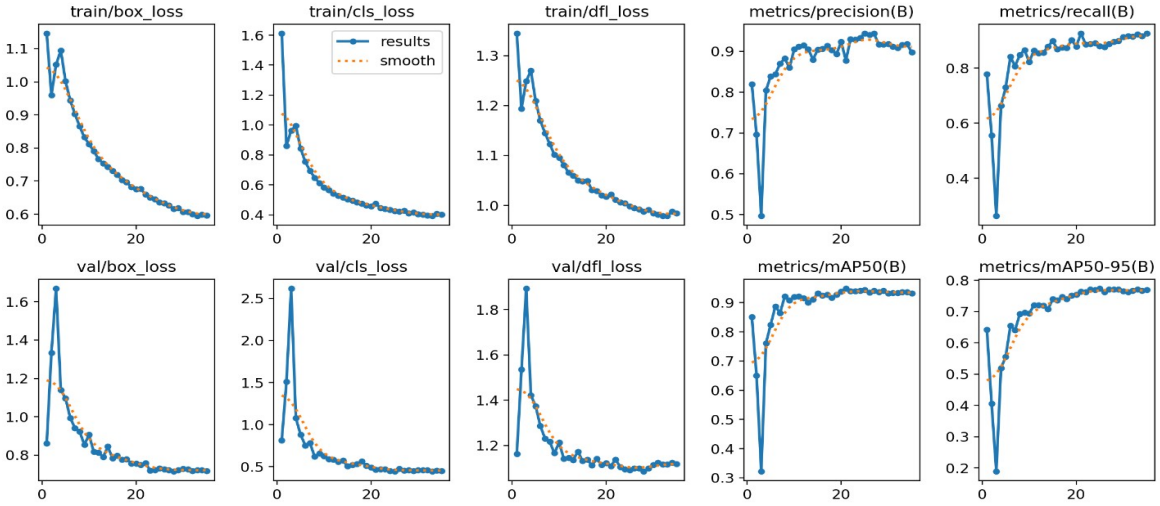

Figure 4: Results of the data augmentation experiment

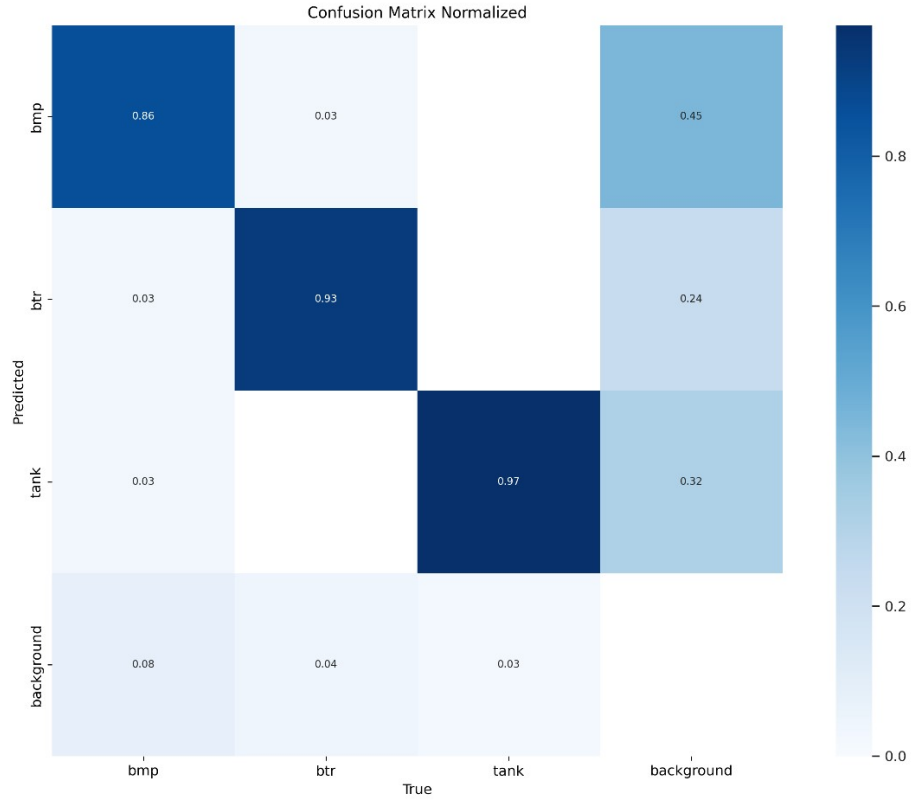

Figure 5: Error matrix for the data augmentation experiment

In addition, the model demonstrates better robustness even when validated on an extended set (the same augmentations as during training), which indicates that there is no “memorisation” of specific angles. Thus, exponential augmentation has a positive effect on the final quality of the detector, reducing the number of errors and increasing the generalisability of all major metrics, including Accuracy, Precision, Recall, F1 and mAP, which is consistent with the findings of a number of studies, on the importance of proper expansion and diversification of datasets in computer vision.

4. Discussion

The results correlate with the findings of a number of researchers who emphasise the importance of diverse and balanced datasets to improve detector accuracy. In particular, [1] and [12] emphasise that the introduction of diverse augmentations (Mosaic, MixUp) significantly enhances the ability of YOLO family architectures to capture complex patterns and reduces the risk of overfitting. Experiments combining an exponential increase in sample size with stepwise colour and geometric transformations confirmed these claims: Precision, recall, and mAP increased by 1–3% compared to the baseline scenario.

At the same time, the proposed approach is consistent with general reviews on image augmentation [7, 10], which emphasise that artificially increasing and diversifying data has a positive effect not only on the quality of detection but also on the stability of the learning process. This is especially true when the initial dataset is relatively small or has significant heterogeneity. Also in [2] and "AutoAugment: Learning augmentation strategies from dataV [3] emphasise that a successful combination of simple basic transformations (flip, rotate, blur) with techniques such as MixUp [14], CutMix [5] or Copy-Paste [4] is more effective than using any one method alone. In the implementation, the exponential transformation chain provided significant coverage of diverse variations, and the subsequent "mixing" of images (Mosaic, MixUp) allowed us to further increase the variety of artificially created scenes.

The findings are also consistent with the results of AutoAugment "AutoAugment: Learning augmentation policies from data", "AutoAugment: Learning augmentation strategies from data" [3], where the authors showed that a carefully selected augmentation policy can increase mAP by several percentage points without changing the network architecture. It has been demonstrated that a similar effect can be achieved without automated policy search, but by consistently "multiplying" the set.

In contrast to classical augmentation, which is mostly implemented once or randomly, the stepwise approach allows us to more effectively "disperse" various variations in the training space, reducing the sensitivity of the detector to specific shooting conditions or angles. An important confirmation of the effectiveness is the reduction of confusion between similar classes and an increase in overall accuracy by 2–3%, which is in line with the trends described in [1], [2], and [10].

Thus, in comparison to the reviewed works, the proposed exponential augmentation method stands out for its holistic approach to sequential expansion and multi-stage data modification. This makes it possible to combine the advantages of basic spatial transformations and more complex techniques such as MixUp without the risk of overtraining, as evidenced by the empirical results.

5. Conclusions

This study provides a detailed overview of modern image augmentation methods and their application to improve the accuracy of deep learning models in detection tasks. Particular attention is paid to YOLO architectures, which have proven to be effective in real-time object analysis in recent years.

The main achievement of the work is the development and implementation of exponential augmentation, a step-by-step approach to artificially increasing the training set, which involves the consistent application of basic geometric and photometric transformations (flips, rotations, GaussianBlur), as well as more complex techniques such as Mosaic and MixUp. Comparative analysis has shown that the proposed method:

- Significantly increases the diversity and volume of the dataset, improving the model's resistance to different angles, backgrounds and sudden noise;
- It improves key accuracy metrics (mAP, Precision, Recall) by 1–3% and overall accuracy by about 2–3% compared to a model trained without augmentations;

- Reduces the risk of overlearning by creating a wider space of training examples and accelerates convergence in the early stages of training.

Thus, exponential augmentation demonstrates its effectiveness in improving the quality of the YOLO detector and can be easily adapted to any computer vision tasks where there is a need to further expand or diversify the image sample. Further development in this area could include automating the search for optimal transformation sequences or combining exponential augmentation with other regularisation techniques, which would further enhance the generalisability of deep learning models.

Acknowledgments

The research was carried out with the grant support of the Ministry of Education and Science of Ukraine, "Methods and tools for detecting disinformation in social networks based on deep learning technologies" under Project No. 0125U001852. During the preparation of this manuscript/study, the author(s) used [ChatGPT 4o Available, Gemini 2.5 flash, Grammarly] to correct the style and improve the quality of the text, as well as to eliminate grammatical errors. The research results obtained in the article are entirely original. The authors have reviewed and edited the output and take full responsibility for the content of this publication.

Declaration on Generative AI

While preparing this work, the authors used the AI programs Grammarly Pro to correct text grammar and Strike Plagiarism to search for possible plagiarism. After using this tool, the authors reviewed and edited the content as needed and took full responsibility for the publication's content.

References

[1] A. Bochkovskiy, C. Y. Wang, H. Y. M. Liao, YOLOv4: Optimal Speed and Accuracy of Object Detection, arXiv preprint, 2020. doi:10.48550/arXiv.2004.10934

[2] A. Buslaev, et al., Albumentations: Fast and Flexible Image Augmentations, Information, 11(2) (2020) 125. doi:10.3390/info11020125

[3] E. D. Cubuk, et al., AutoAugment: Learning Augmentation Policies from Data, in: IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), 2019, 113–123. doi:10.1109/CVPR.2019.00020

[4] G. Ghiasi, et al., Simple Copy-Paste Is a Strong Data Augmentation Method for Instance Segmentation, in: IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), 2021, 2918–2928. doi:10.1109/CVPR46437.2021.00293

[5] P. Luo, et al., Face Model Compression by Distilling Knowledge from Neurons, in: AAAI Conf. Artif. Intell., 30(1) (2016). https://www.aaai.org/ocs/index.php/AAAI/AAAI16/paper/view/12311

[6] A. Mumuni, F. Mumuni, Data Augmentation: A Comprehensive Survey of Modern Approaches, Array, 16 (2022) 100258. doi:10.1016/j.array.2022.100258

[7] Y. Myshkovskyi, M. Nazarkevych, Method of fingerprint identification based on convolutional neural networks, SISN, 15 (2024) 1–14. doi:10.23939/sisn2024.15.001

[8] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You only Look Once: Unified, Real-Time Object Detection, in: IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2016, 779–788. doi:10.1109/CVPR.2016.91

[9] C. Shorten, T. M. Khoshgoftaar, A Survey on Image Data Augmentation for Deep Learning, J. Big Data, 6(1) (2019) 60. doi:10.1186/s40537-019-0197-0

[10] M. Tan, Q. V. Le, EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks, in: 36th Int. Conf. Mach. Learn. (ICML), 2019, 6105–6114. http://proceedings.mlr.press/v97/tan19a.html

[11] C. Y. Wang, A. Bochkovskiy, H. Y. M. Liao, Scaled-YOLOv4: Scaling Cross-Stage Partial Network, in: IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), 2021, 13029–13038. doi:10.1109/CVPR46437.2021.01283

[12] S. Yun, et al., CutMix: A Regularisation Strategy to Train Strong Classifiers with Localisable Features, in: IEEE/CVF Int. Conf. Comput. Vis. (ICCV), 2019, 6023–6032. doi:10.1109/ICCV.2019.00612

[13] H. Zhang, M. Cisse, Y. N. Dauphin, D. Lopez-Paz, Mixup: Beyond Empirical Risk Minimisation, in: Int. Conf. Learn. Represent. (ICLR), 2018. https://openreview.net/forum?id=r1Ddp1-Rb

[14] Z. Zhong, et al., Random Erasing Data Augmentation, in: Proc. AAAI Conf. Artif. Intell., 34(7) (2020) 13001–13008. doi:10.1609/aaai.v34i07.7000

[15] M. Nazarkevych, V. Buriachok, N. Lotoshynska, S. Dmytryk, Research of Ateb-Gabor filter in biometric protection systems, in: IEEE 13th Int. Sci. Tech. Conf. Comput. Sci. Inf. Technol. (CSIT), vol. 1, 2018, 310–313. doi:10.1109/STC-CSIT.2018.8526650

[16] M. Nazarkevych, et al., Application Perfected Wave Tracing Algorithm, in: IEEE 1st Ukraine Conf. Electr. Comput. Eng. (UKRCON), 2017, 1011–1014. doi:10.1109/UKRCON.2017.8100532

[17] V. Sokolov, P. Skladannyi, A. Platonenko, Video Channel Suppression Method of Unmanned Aerial Vehicles, in: IEEE 41st Int. Conf. on Electronics and Nanotech-nology (2022) 473–477. doi:10.1109/ELNANO54667.2022.9927105

[18] M. Nazarkevych, et al., Identification of Biometric Images by Machine Learning, in: IEEE 12th Int. Conf. Electron. Inf. Technol. (ELIT), 2021, 95–98. doi:10.1109/ELIT53502.2021.9501178