Quest Journals
Journal of Software Engineering and Simulation
Volume 10 ~ Issue 9 (2024) pp: 01-08
ISSN(Online) :2321-3795 ISSN (Print):2321-3809
www.questjournals.org

Research Paper

Multifunctional Image Enhancement Tool with Image Processing Methods

Caner Kara[1], Mehmet Ali Yalçınkaya[2], Enes Açıkgözoğlu[3], Ayhan Arısoy[4]
[1] Uluborlu Selahattin Karasoy Vocational School, Information Technologies, Isparta University of Applied Sciences, Isparta, Turkey, canerkara@isparta.edu.tr
[2] Faculty of Engineering-Architecture Computer Engineering, Kırşehir Ahi Evran University, Kırşehir, Turkey, mehmetyalcinkaya@ahievran.edu.tr
[3] Keçiborlu Vocational School, Computer Programming, Isparta University of Applied Sciences, Isparta, Turkey, enesacikgozoglu@isparta.edu.tr
[4] Bucak Emin Gülmez Vocational School of Technical Sciences, Web Design and Coding Programme, Burdur Mehmet Akif Ersoy University, Burdur, Turkey,
aarisoy@mehmetakif.edu.tr
Corresponding Author: Ayhan ARISOY

ABSTRACT: *The success of image-based machine learning and deep learning algorithms depends on the amount and variety of data sets. The richness and heterogeneity of the dataset has a direct impact on the capacity of the model to generalize to new, previously unknown data. Obtaining a large number and variety of images is a very difficult and laborious process, especially for practical applications to be developed in areas such as health, industry and agriculture. If this difficulty cannot be overcome, the performance of machine learning models may decrease or may not reach the desired level at all. This problem may lead to problems such as poor performance or overfitting of the developed models. To solve this problem, data augmentation techniques are used to increase the amount and variety of data sets. These strategies help to provide the necessary data set for the model to learn more successfully. In this publication, we present an image improvement tool that is specifically developed for academics working with limited data sets. This tool uses a variety of data augmentation techniques, including rotation, zooming, panning, noise addition, and blurring, to expand the scale and diversity of available data. This application offers a realistic method for dealing with the issues of limited data availability in machine learning research.*
KEYWORDS: *Image enhancement, image enhancement tool, image processing, image dataset preparation*

Received 23 Aug., 2024; Revised 01 Sep., 2024; Accepted 03 Sep., 2024 © The author(s) 2024. Published with open access at ***www.questjournas.org***

I. INTRODUCTION

Image-based machine learning and deep learning algorithms have been integrated and applied in many fields in recent years. Especially in areas such as health, automotive, security and agriculture. It is widely applied on MRI, tomography, radiology images in the field of health; intelligence acquisition, strategic positioning, etc. in the field of security; disease and species detection in the field of agriculture. The most important criterion for these applications to be applied to real life is that the success of the algorithms has a sufficient level. The main reason for this high success depends on the quantity, diversity and quality of the data sets used. Most of the time, it is not possible to obtain enough annotated images in real applications. Especially in sectors such as medical imaging and monitoring of unusual production processes, there may be a limited amount of data. This leads to difficulties such as overfitting, which reduces the generalizability of the models and can have a negative impact on the performance of the system.

Researchers have turned to alternate data augmentation strategies to overcome the problems of developing and collecting data sets for their investigations. The data augmentation technique creates new data samples by making mathematical and geometric changes to current ones. The methods used in the data augmentation process enrich the data set while eliminating some of the disadvantages that the trained model may have. These techniques, which are widely used in image processing, enable the model to learn from a heterogeneous data pool, thereby increasing model accuracy and decision-making ability. Thus, the risk of overfitting the model is reduced. In data augmentation processes, new images are created by rotating, resizing or shifting an image in the existing dataset. In this way, the trained model improves its detection capability by seeing the same image from different angles. In addition, data augmentation techniques such as adding noise and blurring to the original image help the model to perform classification under more difficult conditions. The image augmentation tool developed in this study creates a versatile and effective data augmentation environment by combining techniques such as rotating, zooming, panning, adding noise and blurring images.

Another data augmentation method is synthetic data augmentation. Synthetic data augmentation is a method that is not derived from existing data and is produced entirely with the support of artificial intelligence. Synthetic data augmentation methods include techniques such as GANs, simulations, 3D modelling, and data manipulation. The similarities and differences, advantages and disadvantages of classical data augmentation and synthetic data augmentation are shown in detail in Table 1.

Table 1. Comparison of synthetic data augmentation and classical data augmentation methods

Feature	Synthetic Data Augmentation	Classic Data Augmentation
Description	The production of new images entirely artificially, usually by simulation or modelling.	Enhancement of existing images by manipulating them with various transformations (rotation, zoom, panning, etc.).
Data Diversity	A wide variety of new data samples can be generated.	It remains within the boundaries of the existing dataset; no new data can be derived.
Realism	It can detract from real-world data, especially in low-quality synthetic productions.	It retains its realism because it is derived from real-world images.
Data Requirement	Large data sets can be created without the need for a real data set.	It needs a basic dataset; no operation can be performed without a basic data set.
Cost and Time	It can be costly and time consuming, especially when complex models or simulations are required.	Low cost and fast; simple conversions are applied.
Application Areas	In specialized areas, it can be used where real data collection is difficult (e.g. autonomous vehicle simulations, medical images).	Widely used in general purpose applications.
Risk of overfitting	Poor quality or incorrectly generated synthetic data can lead to overfitting.	Over-transformation of the same data can lead to overfitting but is usually more controlled.
Generalization Ability	It can increase the ability to generalize as it can simulate various situations and conditions.	The ability to generalize depends on the limits of the transformations, it largely depends on the nature of the data set.
Technology Requirement	Requires high technology and expertise (e.g. GANs, 3D modelling).	Low technology and basic knowledge are sufficient (basic image processing techniques).
Data Volume	An unlimited amount of data can be generated, but quality control can be difficult.	The data volume depends on the size of the available data set and the number of transformations to be applied.
Ethical Issues	Synthetic data can lead to ethical problems (e.g. the use of falsified data).	Derived from real data, ethical issues are generally less.

In the second part of the study, previous studies in the literature about this study are analyzed. In the third section, the visual data augmentation tool developed in this study is presented in detail. The last section presents the conclusions and future directions of the study.

II. RELATED WORKS

[1] in their study, investigated the impact of convolutional network depth on accuracy in large-scale image recognition tasks. Their research demonstrated that increasing the network depth to 16-19 layers using small (3x3) convolution filters significantly improves performance compared to existing methods. [2] in their study, investigated the benefits of data augmentation using synthetically created samples in training machine learning classifiers. The study experimentally evaluated the effects of data augmentation methods obtained through transformations in dataspace and synthetic over-sampling in feature-space on performance. The results revealed that when plausible data transformations are known, data augmentation in dataspace provides greater benefits compared to augmentation in feature-space. Additionally, they highlighted the success of these architectures in the ImageNet challenge and noted that these models also perform effectively on other datasets. [3] in their study, developed a learning framework called "Residual Learning" to overcome the challenges of using deeper layers in deep learning models. This method allows layers to learn residual functions with reference to the input values rather than learning unreferenced functions directly, making it feasible to increase the network's depth. Their work demonstrates that these structures, known as ResNet, provide significant performance improvements on large datasets like ImageNet, greatly enhancing the deep representation capabilities of these models. [4] developed a data augmentation strategy called "Smart Augmentation." This strategy aims to enhance classification accuracy and reduce overfitting by creating a network that learns to generate augmented data in a way that minimizes network losses during the training of deep neural networks. Their study demonstrates that this method can significantly improve accuracy across various datasets and achieve high performance levels even with

smaller networks. [5] in their study, explored effective ways of data augmentation by applying simple transformations to expand datasets. They demonstrated that performing transformations in a learned feature space, rather than in the raw input space, is more effective in enhancing the performance of supervised learning models. This method helped neural networks generalize better across different datasets. [6] in their study, investigated the effectiveness of data augmentation techniques for image classification. They compared different data augmentation strategies to examine how they can improve model performance on limited datasets. Specifically, they evaluated the impact of using GANs (Generative Adversarial Networks) for style transfer in data augmentation, demonstrating how these methods effectively enhance classification accuracy. [7] in their study, explored how data augmentation techniques impact the performance of deep learning models. They specifically compared geometric and photometric data augmentation methods, evaluating which techniques are more effective in enabling deep neural networks to generalize better across different datasets. [8] developed a method to automate the data augmentation process by using task-specific and domain-specific transformations. They introduced a model that learns user-specified transformation functions and applies them in data augmentation using a Generative Adversarial Networks (GAN) approach. Their work demonstrates the effectiveness of this method in enhancing classification performance, particularly on limited labeled datasets, across both image and text datasets. [9] in their study, addressed the significance of data augmentation in deep learning models for image classification problems. The study compares traditional image transformations (such as rotation, cropping, zooming) with more advanced techniques like Style Transfer and Generative Adversarial Networks (GANs). The authors specifically developed a new style transfer-based data augmentation method to address data scarcity in medical imaging and tested its impact on skin melanomas, histopathological images, and breast MRI scans. [10] introduces a new and simple technique called "SamplePairing" for data augmentation in image classification tasks. This technique aims to create new samples by combining two randomly selected images from the training set. The author demonstrates that this method significantly improves classification accuracy even with limited datasets, making it particularly valuable in fields like medical imaging where data scarcity is a concern. [7] in their study, explored data augmentation techniques to enhance the performance of deep learning models in low-data regimes. Specifically, they developed a Data Augmentation Generative Adversarial Network (DAGAN) model that aims to generate new data samples using Generative Adversarial Networks (GANs), demonstrating how this model significantly improves the generalization performance of classifiers. [11] in their study emphasize the reliance of deep learning models on large datasets for achieving successful performance. They thoroughly discuss how data augmentation techniques can enhance the performance of deep learning models in domains with limited datasets. This study examines various data augmentation methods, including geometric transformations, color space changes, mixing, and random erasing. [12] in their study, emphasized the significance of data augmentation techniques to enhance the generalization capability of deep learning models. Specifically, they developed an automated data augmentation method called RandAugment. This method operates with a simpler search space compared to previous automated augmentation strategies and has achieved state-of-the-art results in both classification and object detection tasks. [13] in their study, developed a data augmentation strategy called CutMix to enhance the classification and localization performance of deep neural networks. Unlike existing augmentation strategies, this method replaces randomly removed regions of an image with patches from another image, rather than simply discarding them. This approach not only improves classification accuracy but also enhances the model's localization capabilities by compensating for information loss during training. [14] introduced Albumentations, a flexible and fast library for image processing and augmentation. This library efficiently performs a wide variety of image transformation operations to increase data diversity during the training of deep learning models, demonstrating its applicability to various computer vision tasks. The authors emphasize that Albumentations provides faster performance compared to other popular image augmentation tools while offering a broader range of transformations. [15] conducted a comprehensive review of various techniques used for medical image data augmentation in deep learning applications. They particularly examined basic, deformable, and deep learning-based data augmentation methods employed in radiological tasks, such as CT and MRI, and discussed their impact on artificial intelligence models in clinical settings. [16] in their study, compared various image data augmentation approaches to examine their impact on image classification performance. In addition to traditional data augmentation methods, they proposed new techniques such as Discrete Wavelet Transform (DWT) and Constant-Q Gabor Transform (CQT) and tested their effectiveness across different datasets. Their study demonstrated that the proposed data augmentation methods achieved state-of-the-art or comparable performance on both individual and ensemble models.

III. DEVELOPED TOOL

The visual data augmentation tool developed in this study has an interface that allows users to manipulate images according to certain parameters. In the interface, operations such as determining the input and output indexes, rotating the images at certain angles, and shifting them on the horizontal and vertical axis can be performed. In addition, advanced options such as adding blur and noise are also available to users. These features enable effective data augmentation on datasets with a limited number of images. The interface of the tool is shown in Figure 1.

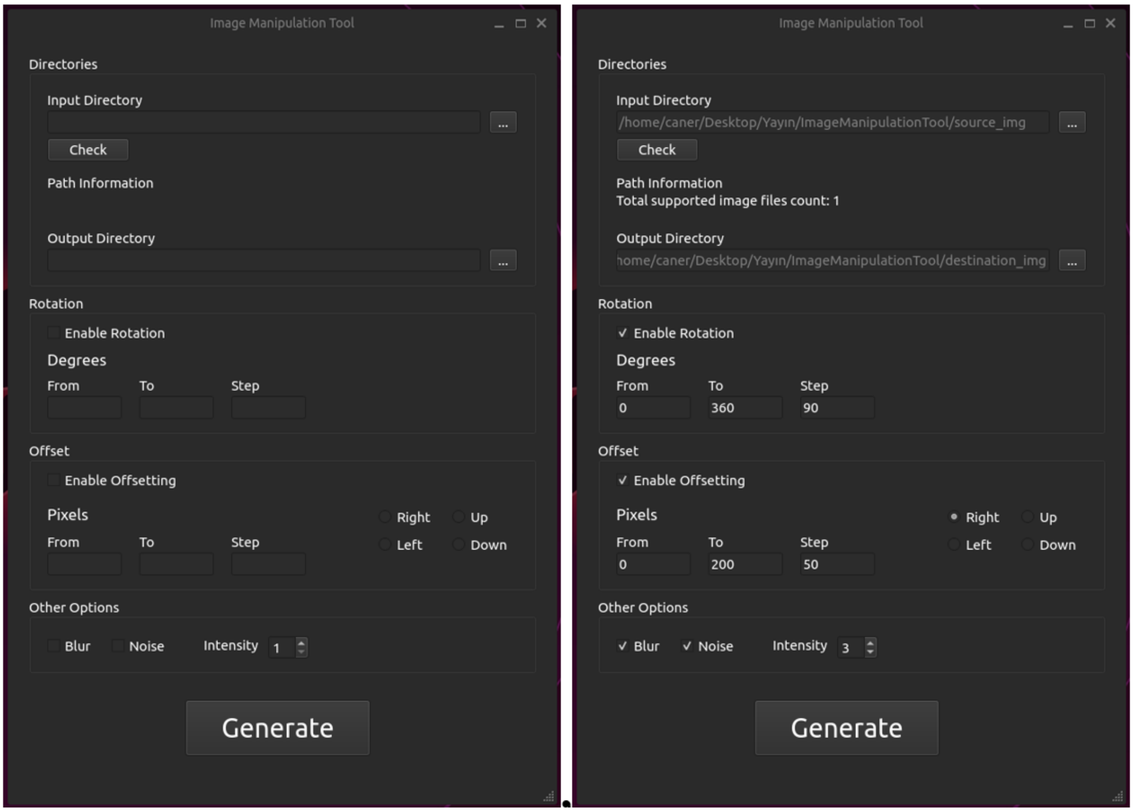


Figure 1. Interface of the developed tool

In the Input Directory field of the tool, the folder containing the source images to be processed is selected by the user by clicking the '...' button. The Check button can be pressed to display the number of supported image files in the selected source folder. As a result of this process, the total number of files in the selected folder is displayed in the Path Information field. This allows the user to quickly verify the contents of the source folder.

In the Output Directory field, the destination folder where the processed images will be saved is determined. This is done by clicking the '...' button again. The destination folder is where the newly generated images will be stored, and it is important to choose this folder correctly in order to manage the data in an organized way.

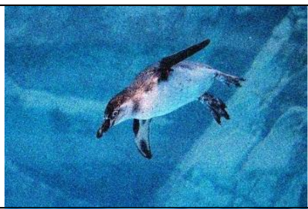
If rotation is desired, the Enable Rotation option should be selected. When this option is activated, it is possible to rotate the images starting from a certain angle (From), up to another angle (To) and in specified degree steps (Step). By specifying the rotation in degrees, the user can create images obtained from different angles for model training. If offsetting is to be performed, this process is activated by selecting the Enable Offsetting option. If this option is enabled, images are generated by offsetting starting from a certain pixel (From), up to another pixel (To) and in specified pixel steps (Step). Scrolling can be used to increase the flexibility of the model, especially when the visual data are placed in different locations. In the Other Options section of the tool, the user can optionally select the Blur or Noise options. These options contribute to training the model on more diverse and realistic data by adding blur or noise to the images. Intensity levels can be set by entering a value between 1 and 10 in the Intensity field. This value determines the intensity of the added effect.

Finally, after all adjustments are made, click on the Generate button. This process generates new images in the target folder specified by the user in accordance with the selected and entered parameters. The generated images can be used to enrich the dataset and improve the performance of the model. Table 2 shows the visuals of rotation and shift examples. Table 3 shows the visuals of the noise and blurring examples.

Table 2. Examples of the application of the developed tool with rotation and shift

Feature	Degree : 0	Degree : 90	Degree : 180	Degree : 270
Rotation				
Feature	**Shift_Right: 0**	**Shift_Right: 100**	**Shift_Right: 150**	**Shift_Right: 200**
Shift				

Table 3. Examples of noise and blurring application of the developed tool

Feature / Intense	Value : 1	Value: 3	Value: 5
Noised			
Blurred			

The algorithm of the tool created to generate the image is given in table 4. This algorithm basically creates new images by processing images and applying different transformations and saves them in the specified folder. The operations are performed according to the options selected by the user through the UI.

Table 4. Generate Image Algorithm

Algorithm 1 Generate Images

```
source path ← IMT UI.txt source path.text()
output path ← IMT UI.txt destination path.text()
for each file in os.listdir(source path) do
    if file ends with ".jpg" or ".png" then
        img ← Image.open(source path + "/" + file)
        np img ← np.array(img)
        if IMT UI.chk enable rotation.isChecked() then
            deg from ← int(IMT UI.txt deg from.text())
            deg to ← int(IMT UI.txt deg to.text())
            deg step ← int(IMT UI.txt deg step.text())
            for i from deg from to deg to with step deg step do
                rotated ← ndimage.rotate(np img, i, reshape = True)
                new path ← output path + "/rot" + str(i) + "deg" + file
                new img ← Image.fromarray(rotated.astype(np.uint8))
                new img.save(new path)
            end for
        end if
        if IMT UI.chk enable offset.isChecked() then
            pix from ← int(IMT UI.txt pix from.text())
            pix to ← int(IMT UI.txt pix to.text())
            pix step ← int(IMT UI.txt pix step.text())
            direction ← 1
```

```
            for i from pix from to pix to with step pix step do
                if IMT UI.rd shift right.isChecked() then
                    direction ← (0, i, 0)
                    path part ← ”/shifted right ”
                else if IMT UI.rd shift left.isChecked() then
                    direction ← (0, i ∗ −1, 0)
                    path part ← ”/shifted left ”
                else if IMT UI.rd shift down.isChecked() then
                    direction ← (i, 0, 0)
                    path part ← ”/shifted down ”
                else if IMT UI.rd shift up.isChecked() then
                    direction ← (i ∗ −1, 0, 0)
                    path part ← ”/shifted up ”
                end if
                shifted ← ndimage.shift(np img, direction)
                shifted path ← output path + path part + str(i) + ”pix” + file
                shifted img ← Image.fromarray(shifted.astype(np.uint8))
                shifted img.save(shifted path)
            end for
        end if
        if IMT UI.chk blur.isChecked() then
            intensity ← IMT UI.sp intensity.value()
            blurred     ← ndimage.gaussian filter(np img, sigma= (intensity, intensity, 0))
            blurred path ← outp1ut path + ”/blurred intensity ” + str(intensity) + ”” + file
            blurred img ← Image.fromarray(blurred.astype(np.uint8))
            blurred img.save(blurred path)
        end if
        if IMT UI.chk noise.isChecked() then
            intensity ← IMT UI.sp intensity.value()
            noise ← np.random.normal(0, intensity ∗ 10, np img.shape)
            noised ← np img + noise
            noised ← np.clip(noised, 0, 255)
            noised path ← output path+”/noised intensity ”+str(intensity)+ ”” + file
        end if
    end if
end for
```

IV. DISCUSSION AND CONCLUSION

The tests for the tool developed in the study were carried out on a computer with an i7 processor, an 8-core CPU and 8 GB of RAM. The image resolution used in the tests is 640x426 pixels. In the tests performed, the image reproduction process was measured using different capabilities of the developed tool.

Figure 2 shows the time taken when the ‘Blur + Noise’ image enhancement process is performed on different numbers of images. The horizontal axis represents the processing time in seconds (Benchmark Time), while the vertical axis represents the number of images processed (Picture Count). The results show that as the number of pictures increases, the processing time increases proportionally. While for 1 image the processing time is very low, for 10 images this time increases significantly. This shows that the computational load of the ‘Blur + Noise’ process is directly related to the number of images.

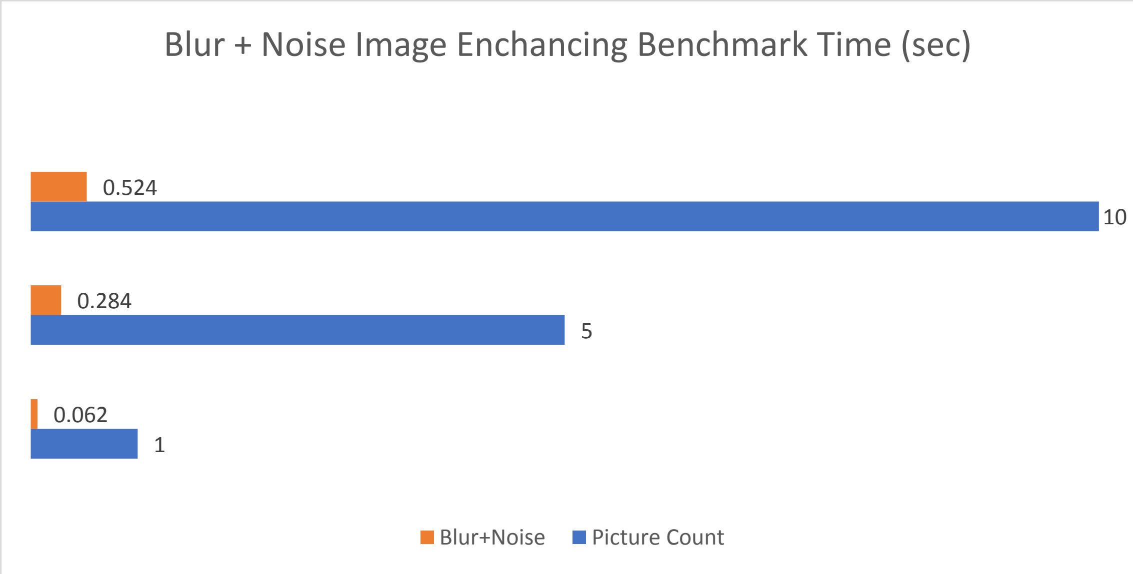


Figure 2. Interface of the developed tool

Figure 3 shows the benchmark times measured in seconds for enhancing images using a combination of blurring and noise application. The graph compares these times with three different image counts: 1, 5 and 10 images. The data clearly shows a linear increase in processing time as the number of images increases. This indicates that the computational complexity of the blurring and noise enhancement technique increases proportionally to the number of images processed. The findings are critical for understanding the performance implications of applying such techniques in scenarios involving large datasets where processing time becomes a significant factor.

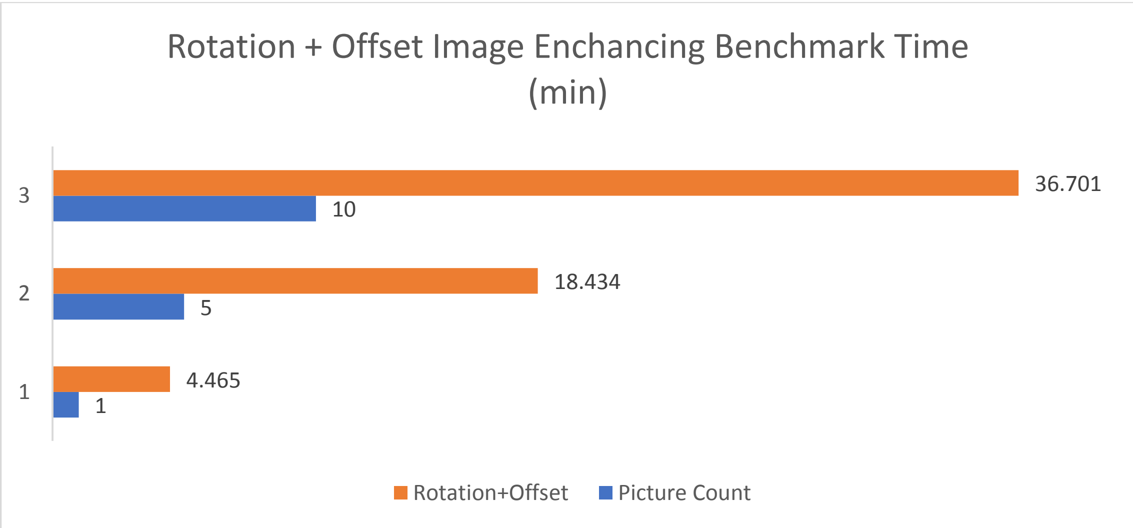


Figure 3. Interface of the developed tool

In conclusion, the developed visual data augmentation tool provides an effective solution for researchers with limited data sets. Benchmark tests on the 'Blur + Noise' process show that the processing time increases proportionally with the increase in the number of images. This result reveals that processing time is an important factor in the application of such image enhancement techniques on large data sets. By combining different data augmentation techniques, the developed tool allows for more general and more accurate training of machine learning models. In the future, the applicability of this tool on larger data sets and its potential for use in different fields will be investigated.

REFERENCES

[1]. K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," 3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc., Sep. 2014, Accessed: Aug. 30, 2024. [Online]. Available: https://arxiv.org/abs/1409.1556v6

[2]. S. C. Wong, A. Gatt, V. Stamatescu, and M. D. McDonnell, "Understanding Data Augmentation for Classification: When to Warp?," 2016 Int. Conf. Digit. Image Comput. Tech. Appl. DICTA 2016, Dec. 2016, doi: 10.1109/DICTA.2016.7797091.

[3]. K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," 2016. Accessed: Aug. 30, 2024. [Online]. Available: http://image-net.org/challenges/LSVRC/2015/

[4]. J. Lemley, S. Bazrafkan, and P. Corcoran, "Smart Augmentation Learning an Optimal Data Augmentation Strategy," IEEE Access, vol. 5, pp. 5858–5869, 2017, doi: 10.1109/ACCESS.2017.2696121.

[5]. T. DeVries and G. W. Taylor, "Dataset Augmentation in Feature Space," 5th Int. Conf. Learn. Represent. ICLR 2017 - Work. Track Proc., Feb. 2017, Accessed: Aug. 30, 2024. [Online]. Available: https://arxiv.org/abs/1702.05538v1

[6]. L. Perez and J. Wang, "The Effectiveness of Data Augmentation in Image Classification using Deep Learning," Dec. 2017, Accessed: Aug. 30, 2024. [Online]. Available: https://arxiv.org/abs/1712.04621v1

[7]. A. Antoniou, A. Storkey, and H. Edwards, "Data Augmentation Generative Adversarial Networks," J. Phys. A Math. Theor., vol. 44, no. 44, pp. 1–13, Nov. 2017, Accessed: Aug. 30, 2024. [Online]. Available: https://arxiv.org/abs/1711.04340v3

[8]. A. J. Ratner, H. R. Ehrenberg, Z. Hussain, J. Dunnmon, and C. Ré, "Learning to Compose Domain-Specific Transformations for Data Augmentation," Adv. Neural Inf. Process. Syst., vol. 30, 2017.

[9]. A. Mikołajczyk and M. Grochowski, "Data augmentation for improving deep learning in image classification problem," 2018 Int. Interdiscip. PhD Work. IIPhDW 2018, pp. 117–122, Jun. 2018, doi: 10.1109/IIPHDW.2018.8388338.

[10]. H. Inoue, "Data Augmentation by Pairing Samples for Images Classification," Jan. 2018, Accessed: Aug. 30, 2024. [Online]. Available: https://arxiv.org/abs/1801.02929v2

[11]. C. Shorten and T. M. Khoshgoftaar, "A survey on Image Data Augmentation for Deep Learning," J. Big Data, vol. 6, no. 1, pp. 1–48, Dec. 2019, doi: 10.1186/S40537-019-0197-0/FIGURES/33.

[12]. E. D. Cubuk, B. Zoph, J. Shlens, and Q. V. Le, "Randaugment: Practical Automated Data Augmentation With a Reduced Search Space," 2020.

[13]. S. Yun, D. Han, S. J. Oh, S. Chun, J. Choe, and Y. Yoo, "CutMix: Regularization Strategy to Train Strong Classifiers With Localizable Features," 2019. Accessed: Aug. 30, 2024. [Online]. Available: https://github.com/clovaai/CutMix-PyTorch.

[14]. A. Buslaev, V. I. Iglovikov, E. Khvedchenya, A. Parinov, M. Druzhinin, and A. A. Kalinin, "Albumentations: Fast and Flexible Image Augmentations," Inf. 2020, Vol. 11, Page 125, vol. 11, no. 2, p. 125, Feb. 2020, doi: 10.3390/INFO11020125.

[15]. P. Chlap, H. Min, N. Vandenberg, J. Dowling, L. Holloway, and A. Haworth, "A review of medical image data augmentation techniques for deep learning applications," J. Med. Imaging Radiat. Oncol., vol. 65, no. 5, pp. 545–563, Aug. 2021, doi: 10.1111/1754-9485.13261.

[16]. L. Nanni, M. Paci, S. Brahnam, and A. Lumini, "Comparison of Different Image Data Augmentation Approaches," J. Imaging 2021, Vol. 7, Page 254, vol. 7, no. 12, p. 254, Nov. 2021, doi: 10.3390/JIMAGING7120254.