Mathematical Problems of Computer Science 55, 54–61, 2021.
doi: 10.51408/1963-0073

UDC 004.8

Designing and Implementing a Method of Data Augmentation Using Machine Learning

Aren K. Mayilyan

National Polytechnic University of Armenia
e-mail: mayilyan96@gmail.com

Abstract

Efficiency of neural network (NN) models depend on the parameters given and the input data. Due to the complexity of environmental conditions and limitations the data for NN models, especially for the case of images, can be insufficient. To overcome this problem data augmentation has been used to enlarge the dataset. The task is to generate diverse set of images from a small set of images for NN training. Due to data augmentation transformation, 3105 new images out of 345 input data were created for classification, detection and image segmentation.

Keywords: Machine Learning, Convolutional Neural Networks, Data Augmentation, burn degrees.

Article Info: Received 10 February 2021; accepted 20 April 2021.

1. Introduction

Nowadays image classification problems are mainly solved via convolutional neural networks (CNN). Deep learning CNN needs a huge number of images for the model to be trained effectively. If the issue of data scarcity is faced, the simple, yet effective techniques such as transformations may pose a limited solution [1, 2].

Data augmentation techniques in data analysis are used to generate slightly modified copies or create synthetic data from a real dataset and hence artificially increase its size. CNN is known to be invariant, meaning that it can robustly classify objects with different transformations. Hence, the increase of relevant data in the dataset will result in a better accuracy for the CNN model.

Besides adding more images to our dataset, the data augmentation tools are beneficial for having images in a limitless set of conditions. In real-world scenarios pictures can be taken in different orientation, location, scale, brightness, etc. Therefore, various transformations of the same image such as rotation and cropping, can enhance the utility of the training set and increase the performance of ML algorithms.

There are two options for data augmentation:

- online augmentation or augmentation on the fly is applied in real time. This method performs transformations on the mini-batches and then fits into the model. This means that the online augmentation will see different images at each epoch. This kind of expansion is preferred for larger datasets, otherwise there would be an explosive increase in size.
- offline version transforms each image in the training set by rotating, cropping, etc. As a result, the size of the training dataset increases by a factor equal to the number of transformations performed on the image. Offline data augmentation is preferred for relatively smaller datasets as the goal is having more images to train for the model [3, 4].

2. Description of the Dataset

The collected dataset consists of around 400 images of human skin burns. The pictures of the burns are taken from different angles mainly in a monotone background. The dataset is labeled into three classes according to their burn degrees, examples of which are presented in Fig. 1.

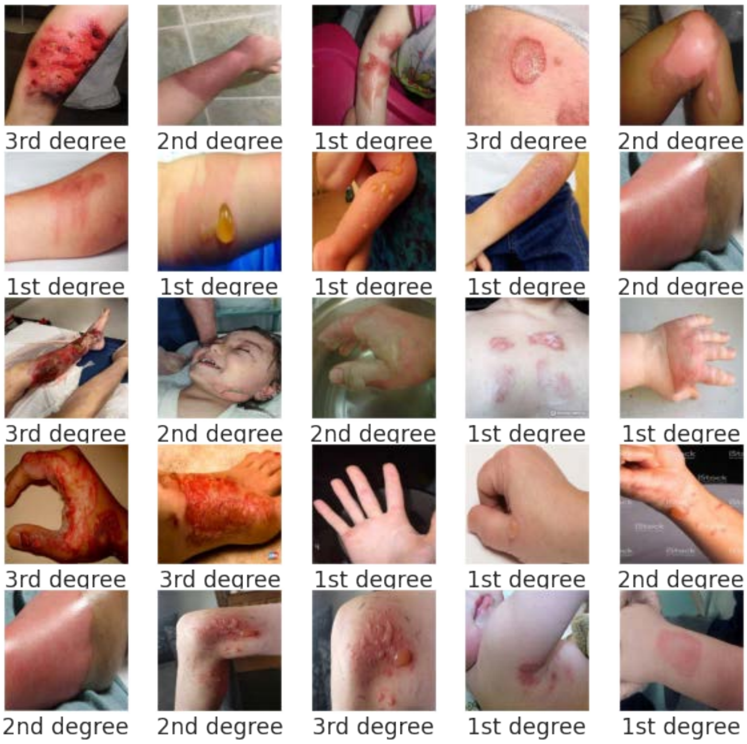

Fig.1. Some examples of images from the dataset

Before designing the model, the dataset is divided into two sets: training and test. On the first one the model is trained, while the second one is used to determine the accuracy of the designed model. In our case 345 out of 400 images form the training dataset. As the dataset was initially labeled into 3 different classes, the training and test sets should consist of items of three classes with each class containing nearly equal number of images. In our case we have approximately 120 images in each class for the training set and 30 images in each class for the test set (Fig. 2).

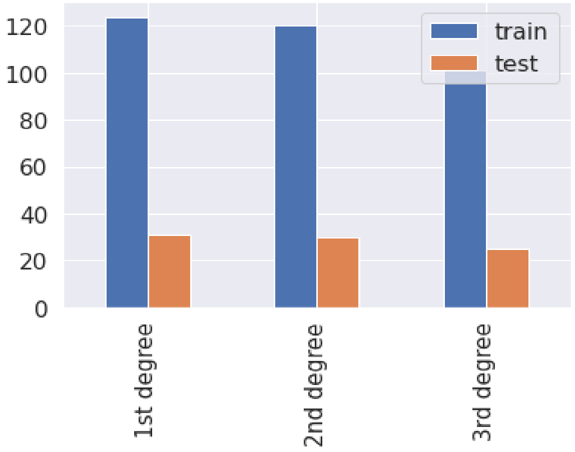

Fig.2. Original train and test sets according to three types of burn degrees

3. Description of the Method

Images are represented as arrays in our data. Colored images are a mixture of red, blue and green (RGB). The pixels of the images are tiny blocks of information arranged in the form of a 2D grid, and the depth of a pixel is the color information. Data augmentation basically shifts or transforms these pixels to get a new image.

In our data we have resized all the images to be of the shape of (150, 150, 3), where the first two numbers show the size of rows and columns of the matrix that form the 2D grid, and the third number indicating the RGB coloring. If we separate the images into three (150, 150, 1)-shaped matrices, we will get three separate red, blue and green colored pictures (Fig.3.)

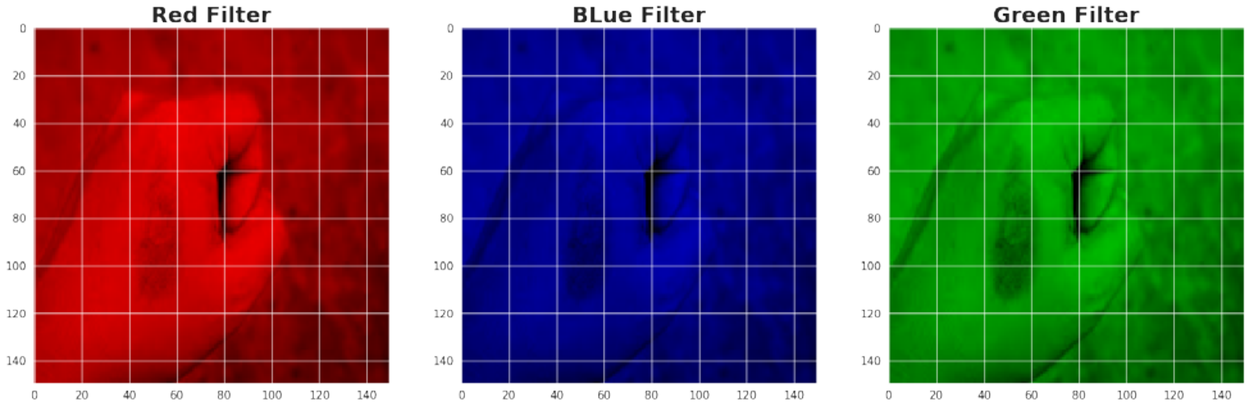

Fig.3. RGB representation of a picture.

Here are five basic and powerful augmentation techniques that we used to increase our dataset [5].

Table 1: 5 Data Augmentation Techniques Used in the program, their application and results

Data Augmentation Technique	Methods used in the program	Result of the Technique

Flip	np.fliplr(image)
	np.flipud(image)
		Flipping images horizontally and vertically

Rotate	rotate(image, angle = 45)
	rotate(image, angle = -45)
		Rotating images by a random angle chosen by the program

Crop	Function defined by the user to crop the main part of the image.	Randomly sampling a section from the original image, then resizing the original image size. This process is known as random cropping

Brightness adjustment	adjust_gamma(image, gamma = 0.5, gain = 1)
	adjust_gamma(image, gamma = 2, gain = 1)
		Changing the original image's darkness/brightness

Noise	random_noise(image)	Adding noise to the image, hence having blurred image with slightly different pixel values

The flip and rotate transformations are alike. Particularly, flipping the image horizontally or vertically is the same thing as rotating the image by 90 or 180 degrees. In order to rotate our images around its center by specified number of degrees, we take the RGB values at every 2D location, rotate it as needed, and then write these values in the new location. Thus, having the location coordinates x and y, we apply the transformation matrix and get the new location for the same RGB value. The calculation is done with multiplication of the old coordinates with the transformation matrix, as shown in formula 1.

$$\begin{bmatrix} x^* \\ y * \end{bmatrix} = \begin{bmatrix} cos\theta & sin\theta \\ -sin\theta & cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix},$$

where θ is the angle by which we want to rotate, x* and y* are the new coordinates, and x and y are the old ones. In our case we have used 90, 180, 45 and -45 degrees. After having the images flipped horizontally and vertically, rotation of the images by 45 and -45 degrees is the most effective, because two other lesser degrees would result in the same original image, e. g. 10 degrees, and, rotating by a large degree will be very close, as to flipping horizontally or vertically. Hence, rotating by 45 degrees, which is the mean value of 0 and 90 degrees, is the best choice for our images.

After these 2 techniques we generate four rotated images, which we append to our training set.

Cropping could be done by randomly choosing a smaller rectangle on our image and cutting out that part. However, for the crop technique the essential condition is having the burn image fully displayed on our image even after cropping it. In order to reach this goal, in our code we first divide the y axis of our image into three equal parts, having the low, middle and high parts. Then we choose a random number that would belong to the low part, and another random number from the high part. As a result, we have two points on y axis. Same goes for the x axis. By having 2 points on the x axis and 2 on the y axis, we then form the rectangle that includes the middle part of the image and which will be cropped out. By doing so we generate one image from each original one and add to the training set.

Brightness adjustment means adjustment of the RGB value. In our technique we have multiplied each of R, G, B values by two constants. In the first case by 0.5, and as 0.5 is less than 1, then it made our images darker. In the second case we multiplied by 2, which made our images brighter. As a result of this technique, we generated two more images.

The last thing we have applied to our images is adding some blur. Most common are box blur and Gaussian noise. For the box blur we took a kernel, which is a 5x5 matrix, rolled on the image, and took the average color of pixels inside that matrix. As for the Gaussian noise, we add random variables from standard normal distribution to our RGB values. Based on these two techniques we get two new images, therefore, overall nine images from each image [10][11].

With application of our data augmentation techniques on each image (Fig. 4) we generated 3105 new images and, hence, together with the original data a new training set consisting of 3450 images [6][7] was produced.

In order to be able to fit into the CNN model, each image should preserve the size of the original dataset, which is 3-dimensional array of (150, 150, 3)-shape. However, in comparison with the original dataset, not all the newly created ones have the necessary shape. It occurs as a result of transforming the arrays. CNN models should have inputs of the same size, hence, it is crucial to bring all the data into the same shape before fitting to the model. The reshaping was done individually for each image after their transformation. After changing the sizes of the images, the information isn’t lost: it is just demonstrated via different shaped arrays [9].

For the transformations such as flip, crop, brightness adjustment, Gaussian noise, etc., the images will retain all the information. For some cases, such as for the rotations, the image does not have any information about things outside its boundary. As we see in Fig.4, the picture in the 2nd row and 1st column has black fillings outside its boundary. In such cases we need to make some assumptions. There are different ways of doing so:

Constant - filling the unknown region with some constant value

Edge - the edge values are extended after the boundary

Reflect - the image pixel values are reflected along the image boundary

Symmetric - at the boundary of reflection, a copy of the edge pixels is made

Wrap - the image is repeated beyond its boundary.

As our images are mainly taken in a monochromatic background, the space beyond the image’s boundary is assumed to be the constant 0 at every point and is displayed with black color [10].

After the application of all the above-mentioned techniques and bringing all the images to the same size, for each image we get nine copies with a slight difference. The augmented images are demonstrated in Fig.4.

There are other types of data augmentation such as Generative Adversarial Networks (GAN). It is questionable why NN data augmentation is not preferred while using CNN model. The answer lies in the data itself. The images that we have are not easily distinguishable even by the human eye. For example, if we were to classify buildings from forests, then with GAN we could generate the same picture of the building in summer, autumn, spring, winter and, hence, have four more pictures in this case. In our case by generating new images using GAN, we can turn an image with 2nd degree burn into a 3rd degree burn image and decrease the accuracy of the CNN model.

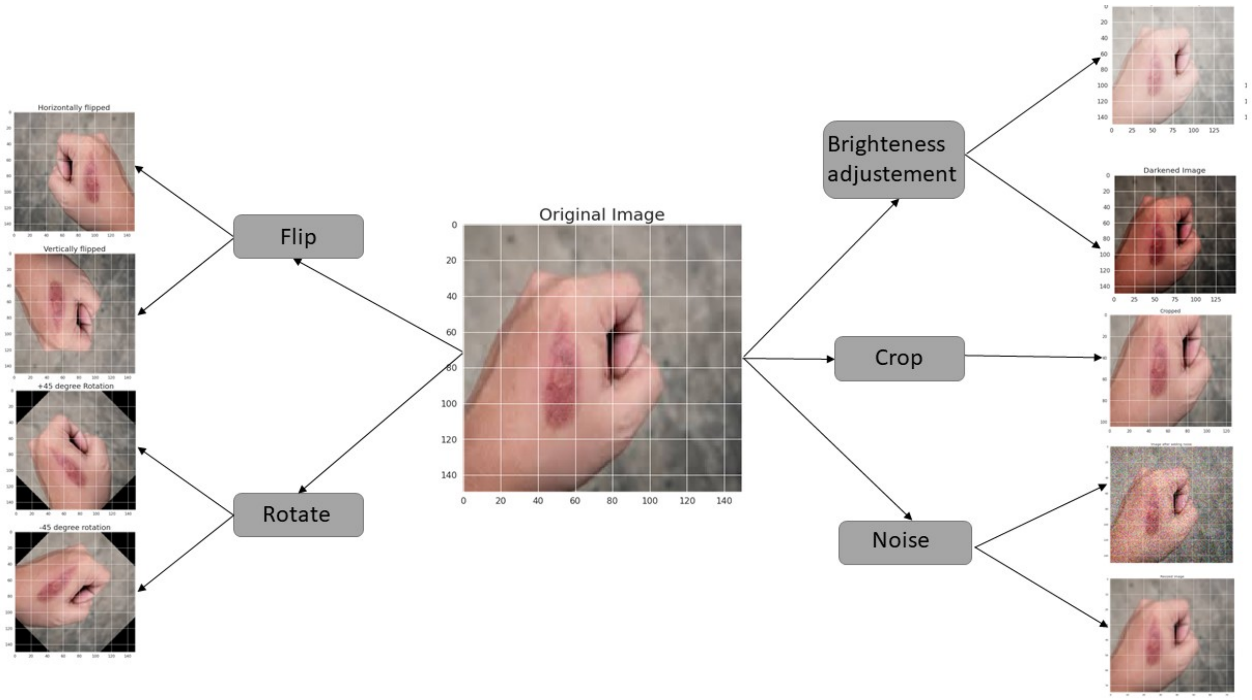

Fig. 4. Application of the Data augmentation techniques on an image.

After the transformations we have two training datasets. The first one is the original one containing 345 unique images, and the second one is the augmented, containing overall 3450 images. In order to understand whether data augmentation is beneficial for the application of the CNN model, we need to plug the original training set of 345 to the model, get the accuracy and then plug in the second augmented training set of 3450 and get the accuracy. The result will be seen by comparing the two accuracies. In our case we applied convolutional neural networks and got 59% accuracy out of original dataset and 75% accuracy out of the augmented dataset.

4. Conclusion

Data augmentation gives the opportunity to add more training data into the model, prevent data scarcity for better models, reduce data overfitting, create variability in data and resolve class imbalance issues in classification. For classification of human skin burns, the newly generated images should not lose much information from the original image. In comparison with our techniques, the other data augmentation tool - generative adversarial network, generates synthetic copies of an image, basically mixtures of images. In our case it is not expedient, as the difference between classes is very slight, and even a small mixture of images will result in bad input data.

Besides all these advantages the transformation of the images reduces costs of collecting and labeling data.

To fully take advantage of the data augmentation, the below mentioned steps were taken:

- Observe data to understand which augmentation tools are better to use;
- Implement and apply those tools on each image to produce 3105 new items (from 345 original images) adding to our training set;

- Create two training datasets. The first one is the original dataset consisting of 345 images and the second is the augmented dataset, containing 3450 images (from which 345 were the original images and 3105 newly generated);
- Fit both training sets into a CNN and get the corresponding models;
- Test the two models on the test set and get the accuracies for each.

Thus, as a result of data augmentation the training set increased from 345 items to 3450 and the CNN accuracy from 59% to 75%.

References

[1] A. Kwasigroch, A. Mikolajczyk and M. Grochowski, “Deep convolutional neural as a decision support tool in medical problems-malignant melanoma case study”, *Trends in Advanced Intelligent Control, Optimization and Automation*, *Advances in Intelligent Systems and Computing*, Springer, Cham, vol 577, pp. 848-856, 2017.

[2] Z. Chen, Z. GaoChen, R. Gao, K. Mao, P. Wang, R. Yan and Zhao, *Deep Learning and Its Applications to Machine Health Monitoring: A Survey*. CoRR, abs/1612.07640, 2016.

[3] M. Frid-Adar, E. Klang, M. Amitai, J. Goldberger and H. Greenspan, “Synthetic data augmentation using gan for improved liver lesion classification”, ArXivPrepr. ArXiv180102385, 2018.

[4] T. A. Rutkowski and F. Prokopiuk, “Identification of the contamination source location in the drinking water distribution system based on the neural network classifier”, *The 10th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes* August 29-31, Warsaw, Poland, 2018.

[5] A. HaydarOrnek and M. Ceylan, “Comparison of traditional transformations for data augmentation in deep learning of medical thermography”, *Telecommunications and Signal Processing (TSP) 2019 42nd International Conference*, pp. 191-194, 2019.

[6] S.-L. Wannipa, W. Wettayaprasit and P. Aiyarak, “Convolutional neural networks using mobilenet for skin lesion classification", *Computer Science and Software Engineering (JCSSE) 2019 16th International Joint Conference on*, pp. 242-247, 2019.

[7] B. Jahić, Nicolas and G. BenoîtRies, “Software engineering for dataset augmentation using generative adversarial networks”, *Software Engineering and Service Science (ICSESS) 2019 IEEE 10th International Conference*, pp. 59-66, 2019.

[8] I. Laina, Ch. Rupprecht, V. Belagiannis, F. Tombari and N. Navab, “Deeper depth prediction with fully convolutional residual networks”, CoRR, abs/1606.00373, 2016.

[9] I. Goodfellow, Y. Bengio and A. Courville, *Deep Learning,* MIT press, 2016.

[10] (04 Oct 2021) [Online]. Available: https://en.wikipedia.org/wiki/Box_blur

[11] (04 Oct 2021) [Online]. Available: https://en.wikipedia.org/wiki/Gaussian_blur

Մեքենայական ուսուցմամբ տվյալների բազայի ընդլայնման մեթոդի մշակումը և կիրառումը

Արեն Կ․ Մայիլյան

Հայաստանի ազգային պոլիտեխնիկական համալսարան
e-mail: mayilyan96@gmail.com

Ամփոփում

Նեյրոնային ցանցի մոդելների էֆֆեկտիվությունը պայմանավորված է տրված պարամետրերով և մուտքային տվյալներով: Արտաքին գործոններով պայմանավորված դժվարություններից և սահմափակումներից ելնելով՝ նեյրոնային ցանցի մոդելների համար հավաքագրված տվյալները սովորաբար բավարար չեն հատկապես պատկերների դեպքում: Այս խոչընդոտը հաղթահարելու նպատակով օգտագործվում են մեքենայական ուսուցման մեխանիզմներ, որոնք թույլ են տալիս ընդլայնելու տվյալների նախնական բազան: Վերջինս ներկայացնող մուտքային պատկերի վերափոխումամբ ստեղծվում է պատկերների բազմազան հավաքածու, հնարավորություն տալով արդյունավետորեն լուծելու դասակարգման, նույնականացման և սեգմենտավորման/մասնատման խնդիրներ:

Բանալի բառեր՝ Մեքենայական ուսուցում, կոնվոլյուցիոն նեյրոնային ցանցեր, տվյալների ընդլայնում, այրվածքի աստիճաններ:

Разработка и применение метода расширения базы данных с помощью машинного обучения

Арен К. Маилян

Национальный политехнический университет Армении
e-mail: mayilyan96@gmail.com

Аннотация

Эффективность моделей нейронных сетей (NN) зависит от заданных параметров и входных данных. Из-за сложности и ограничений, обусловленных внешними факторами, данных для моделей NN, довольно часто бывает недостаточно, особенно в случае изображений. Чтобы решить эту проблему, обычно используется расширение данных для увеличения их набора. Преобразование входных данных создает разнообразный набор изображений из небольшого набора изображений для классификации, обнаружения или сегментации изображения.

Ключевые слова: Машинное обучение, сверточная нейронная сеть, нейронные сети, расширение данных, степени ожога.