Analyzing deep learning algorithms with statistical method

Lucian Sergiu Trifan[1]
[1]Department of Computer Science and Information Technology
Faculty of Automation, Computers, Electrical Engineering and Electronics, Dunarea de Jos University of Galati
Galati, Romania
st119@student.ugal.ro

Simona Moldovanu[1,2]
[1]Department of Computer Science and Information Technology, Faculty of Automation, Computers, Electrical Engineering and Electronics, Dunarea de Jos University of Galati, Romania
[2]The Modelling & Simulation Laboratory, Dunarea de Jos University of Galati, 47 Domneasca Street, 800008
Galati, Romania
simona.moldovanu@ugal.ro

***Abstract*—A number of farmers use human inspection to visually classify the quality of their fruits, this process could be done automatically classification with Deep Learning algorithm. This paper investigates the classification of apples, peaches and oranges and verify the quality of classification with density of the edges detected in each image by Sobel filter. The novelty of the study consists of comparing of classification accuracy of convolutional neural network (CNN) and statistical analysis of edge histograms. The edges are main elements from what the CNN learns, this was the reason choosing verifying the deep learning algorithms with skewness, kurtosis and standard deviation metrics, extracted from histogram of edge density detected by Sobel filter. In empirical process four algorithm were proposed a CNN with one convolutional layer, a DNN without convolutional layers and two pretrained networks as AlexNet and GoogLeNet. From clustering of the row data and analyzing the accuracy, the best classification is gotten by simple CNN for 20 epochs when the apples vs. oranges were classified.**

Keywords—convolutional neural network, pretrained networks, skewness, kurtosis and standard deviation

I. Introduction

Deep learning is a subset of machine learning architectures that aims to represent high-level image classification, segmentation, or recognition objects. The convolutional neural network (CNN), one of the deep learning models, was demonstrated to be helpful in the object recognition and classification processes. For a successful classification a lot of pre-trained CNNs were implemented, to which this category belongs, such as ImageNet, VGG, GoogLeNet, AlexNet, or ResNet. These have shown strong performances in image identification and classification [1, 2, 3].

CNNs are DL methods that are frequently used to evaluate complex images, such as medical, satellite, or microscopic images. In addition to many other features of visual input, CNN algorithms are able to recognize areas, faces, people, pathologies, tumors, and usual objects.

Artificial intelligence (AI) research employing neural networks has been successful due to a number of factors, including large volumes of data over the internet to train neural network models, the number of convolutional layers, various deep learning techniques, including batch normalization, dropout, and rectified linear unit (ReLU) [1].

Usually, the architecture of CNNs contains, an input layer, convolutional layers, a pooling layer, and a fully connected layer [2, 3].

The performance of DL algorithms for automated detection of different classes of fruits has been studied by researchers, and the results of this study are presented in the next section. Our main goal was to accurately assess how well DL techniques classified apples, oranges, and preaches classes when original and pretrained CNN were used [3, 4].

Beside using four algorithms, the main contribution is the statistical analysis of the edges generated by Sobel, a first derivative filter. The first features that CNN learned are the edges; this was the reason for studying their density from fruit edges. If density edges were a factor in the classification results, then the results would be accurate.

II. Related work

This section provides a brief overview of current approaches that use CNN and transfer learning to classify different fruit categories (rotten and fresh).

Dandavate and Patodkar [4] classified banana, papaya, mango, and guava fruits with CNN, and the results were an accuracy of 97.74 % in 8 epochs and a 0.9833 validation set.

In order to recognize multiple fruits more accurately, in terms of similarity in color, shape, size, and texture features, Kausar *et al.* [5] proposed a pure convolutional neural network (PCNN) with a minimum number of parameters. PCNN was trained on 81 categories of fruits, with a test accuracy of 98.88%. The classification of fresh and rotten fruits by Palakodati *et al.* [6] was performed.

In the paper [7] the authors used three types of fruits, such as apple, banana, and oranges have corroborated CNN with transfer learning models, so the results produced an accuracy of 97.82%.

The CNN and machine learning tools Fuzzy and Ensemble Decision Trees and K-Nearest Neighbors were applied in cherry fruit classification. In order to compare the proposed method (CNN) with histogram of gradient (HOG) and Local binary pattern (LBP), a classification process accuracy of 99.4 % was obtained [8].

Kang and Chen proposed a multi-tasking framework and transfer learning with ResNet-50 and ResNet-101 for classifying fresh and rotten fruits, their study provided an accuracy of 98.50% [9].

Cite as: L. S. Trifan and S. Moldovanu, "Analyzing deep learning algorithms with statistical methods", *Syst. Theor. Control Comput. J.,* vol. 4, no. 1, pp. 9–14.
DOI: 10.52846/stccj.2024.4.1.59

III. IMAGE DATASETS

The Apple2Orange[1] and Flickr Image Peaches datasets were used to test how accurate fruit classification was. The original images have a resolution of 256x256 pixels, being resized in the pre-processing stages to 80x80 pixels or 224x224 pixels depending on the network used. The images are represented in the RGB (red, green, and blue) colour model.

The images represented by the third class (peaches) were downloaded using the *FLICKR Api*[2]. The algorithm used is a Python script that uses the Flickr API to search for and download images, as well as resize those images to 256x256 pixels. To download the images, the "*flickr.photos.search*" method is used, which returns a list of photos that correspond to the search criteria (entered as parameters of the method) and the *url* addresses corresponding to each photo. The image will be copied from the URL using the Python *urllib* module and the "*urllib.request.urlretrieve*" method. The number of each category in Table 1 is shown.

TABLE I. NUMBER OF IMAGES FROM EACH DATATASETS

Number of images	**Train**	**Test**
Apples	995	266
Oranges	1019	248
Peach	999	176

IV. HARDWARE AND SOFTWARE

The computer used for training the learning models as well as for the statistical analyses has the following specifications: Processor: Intel(R) Core(TM) i7-4720HQ CPU @ 2.60GHz, Installed RAM: 16.0 GB, Windows Edition: Windows 10 Pro.

The libraries used in Python were:

(i) *sys, datetime* - for generating reports in text files regarding model data - structure, model parameters, accuracy, confusion matrix)

(ii) *sklearn.metrics* (for confusion matrix generation);

(iii) *matplotlib, numpy* (for displaying images from datasets)

(iv) *tensorflow.keras* (layers, models, utils) for creating the network/model structure, compiling and training the network, calculating the accuracy of the model, and testing it.

V. IMAGE PROCESSING AND DENSITY EDGES ANALYSIS

The assumption made by edge detection techniques is that the edges appear in an extremely steep discontinuity in the image's intensity function. An edge is characterized as having an intensity value that is significantly different from nearby pixels [10]. Edge detection involves four stages: detection, edge localization, image smoothing, and enhancement. There are two approaches to object detection with edge detection first and second derivative, in our study Some samples of images in Figure 1 are shown, and their processing in figure

2. Sobel edge detection was applied with the following kernels in Gx and Gy directions [11]:

$$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

The edges detected with Sobel filter were projected on horizontal (eq. 1) and vertical (eq.2)

$$H[i] = \sum_{j=1}^{m-1} I[i,j] \quad (1)$$

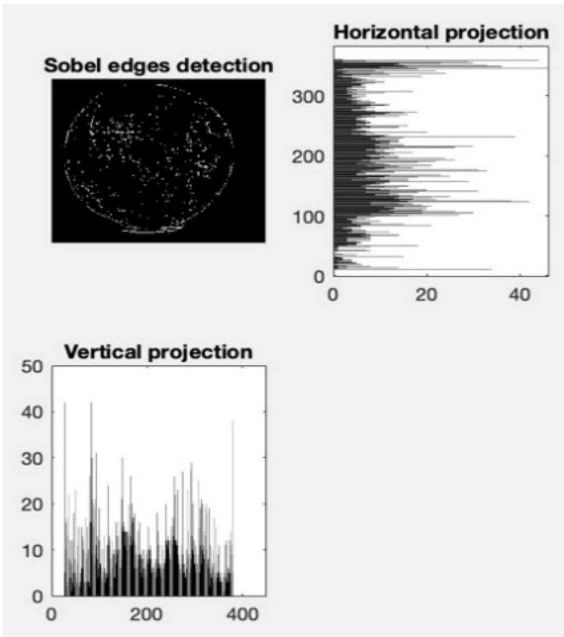
$$V[i] = \sum_{i=1}^{n-1} I[i,j] \quad (2)$$

where for an image with NxM size, $0 < i \leq N$, and o< $j \leq M$, and $I[i,j]$ is the intensity pixel [12].

(a) (b) (c)

Fig. 1 Samples of fruits from used datasets; (a) apple, (b) peach, (c)orange

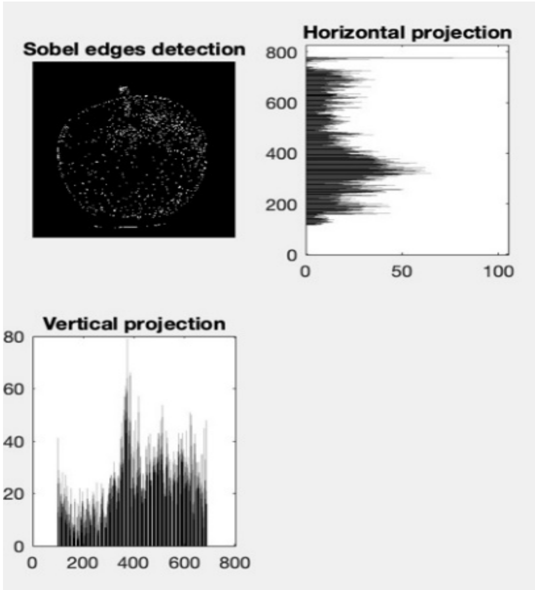
A binary image's vertical and horizontal histograms count the number of white pixels in each column and row, respectively. The following violin graphs were created by interpreting the metrics for standard deviation, skewness, and kurtosis that were taken from each vertical and horizontal histogram,This kind of graph combines the features of a kernel density plot with a box plot (see Figure 3). We can see information about the ranges that the data are spread out throughout and how they are centered in the violin plot. The probability density function (PDF) is used for plotting; a larger density indicates a higher frequency of the data, whereas narrower portions of the density can be seen to indicate that the data occur less frequently [13].



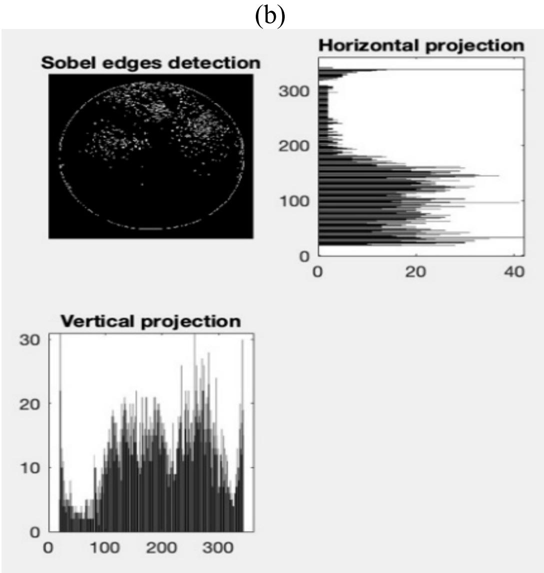
(a)

[1] "Apple2orange Dataset", accessed by 05. 04.2024, https://www.kaggle.com/datasets/balraj98/apple2orange-dataset?resource=download

[2] "Flickr API", https://www.flickr.com/services/api/flickr.photos.search.html



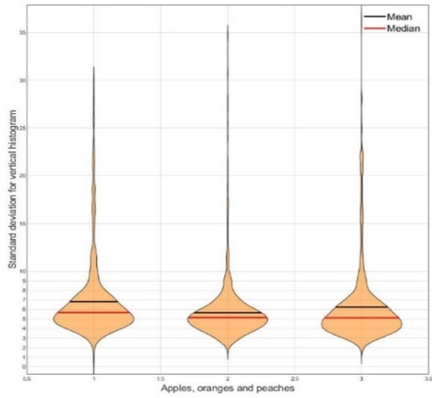
(b)



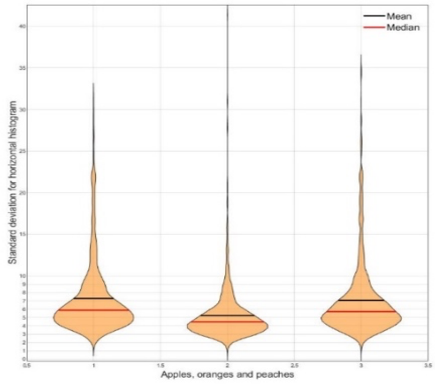
(c)

Fig. 2. Edges detection and projection ther on vertical and horizontal histograms; (a) apple, (b) peach, (c) orange

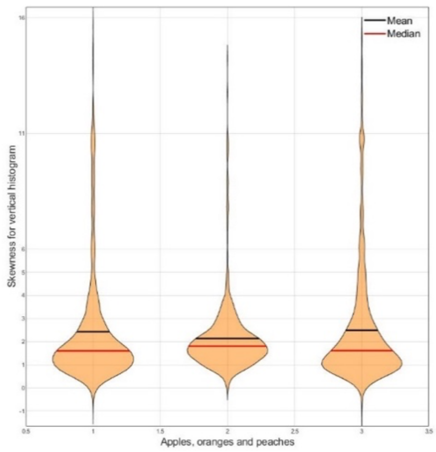
In this study, the violin plots are useful for comparing multiple distributions for three fruit classes and vertical and horizontal histograms. In addition, two markers are indicated: the median of the data, which means the middle point when data points are arranged from high to low, and the mean of all data [14].



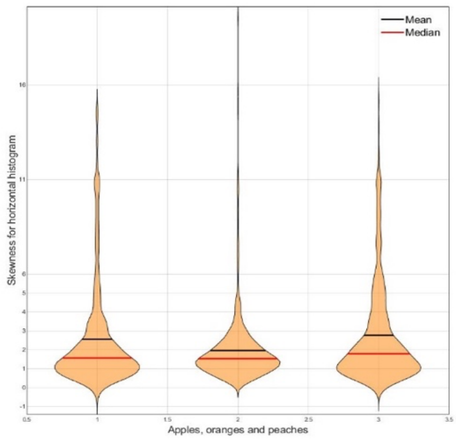
(a1)



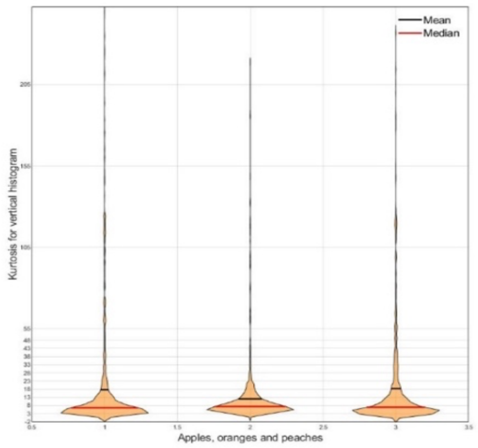
(a2)



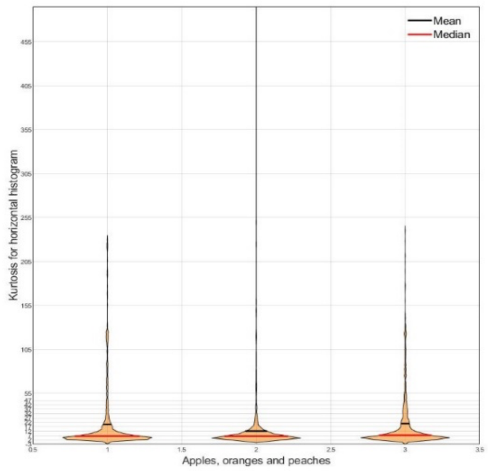
(b1)



(b2)



(c1)



(c2)

Fig. 3. Violin graph for analysing of density edges; (a1) standard deviation for vertical edges; (a2) standard deviation for horizontal edges; (b1) skewness for vertical edges; (b2) skewness deviation for horizontal edges; (c1) kurtosis for vertical edges; (c2) kurtosis deviation for horizontal edges.

VI. Convolutional Neural Newtork and Transfer Learning

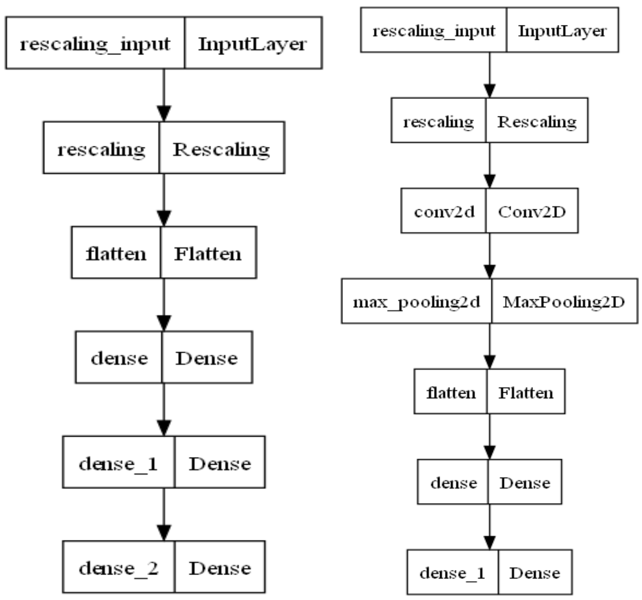
In recent years, convolutional neural networks (CNNs) have demonstrated outstanding performance on a wide range of image categorization tasks. Nonetheless, the architecture of CNNs has a significant influence on their performance. The architectures of the most advanced CNNs are frequently handcrafted by experts in both CNNs and the problems under investigation. As a result, designing the best CNN designs for unique image classification issues might be challenging for users lacking extensive experience with CNNs. Two deep learning algorithms are used, a DNN without convolutional layers and a CNN with convolutional layers, both are shown in Figure 4.

Optimizing and training the model is a difficult and time-consuming process. A robust graphics processing unit (GPU) and millions of training examples are needed for the training. All of the issues are resolved, though, by transfer learning, which is used in DL. The transfer learning by AlexNet and GoogLeNet, was assured. Pre-trained CNN model AlexNet has demonstrated excellent performance over the last few years, Early on in the creation of CNN architectures, AlexNet was suggested, the newest architecture contains eight layers five are convolutional layers, three are fully connected layers [15]. GoogLeNet CNN was proposed by research at Google, her architecture uses convolutions 1×1 size of kernel convolution in the middle of the architecture and global average pooling [16].

VII. Results and Discussion

Two aspects are corroborated in this section: the analysis of edge density and the accuracy classification provided by DL algorithms. Figura 3 shows the interpretation of the edges with violin graphs; from these, the values of the median are extracted, and the values are extracted in Table 2.

This section presents the performance of the proposed DL algorithm in terms of training accuracy, training loss, and test accuracy. The proposed models were trained for 10 and 20 epochs. Using training and testing techniques for apples vs. oranges and apples vs. peaches, a model's performance is quantified and confirmed. As seen in Tables 3–6, the confusion matrix has been utilized to calculate the fruit class performance parameter. Data augmentation techniques like random rotation are used to extend the training samples in order to ensure the diversity of sample images and prevent the overfitting issue. All images were resized to 80 x 80 pixels for the model fit and a minimum 100 images. The training and test sets of 70:30 were divided.



(a) (b)

Fig. 4. Architectures of deep learning algorithms: (a) DNN; (b) CNN

After analyzing the violin graph in accordance with the statistical method and clustering data around the median, in the following table for the vertical and horizontal histograms, the data are stored.

TABLE II. Statistical method for edges analyzing in context of median

Statistical method	Fruit classes	Vertical histogram	Horizontal histogram
Standard deviation	*Apples*	**5.75**	5.92
	Oranges	5.11	5.54
	Peaches	5.22	5.86
Skewness	*Apples*	1.65	1.53
	Oranges	**1.86**	1.59
	Peaches	1.71	1.82
Kurtosis	*Apples*	7.82	7.55
	Oranges	**8.21**	7.64
	Peaches	7.88	6.77

When examining the distribution of numerical data, it was proposed that violin graphs be utilized. This is particularly helpful for comparing distributions across apple, orange, and peach groups. It is possible to examine the peaks, valleys, mean, median, and tails of each group's density curve to determine how distinct or similar the groups are.

The distribution of the edges is different for all fruits, and for all statistical methods, a large difference is between apples and oranges, for a vertical histogram when the standard deviation is computed. This metric measures how spread the edges are, and in this case, the vertical edges have a large distribution for apples (5.57) and a small distribution for oranges. The quantity of edges is more asymmetric for vertical edges that belong to oranges (1.86), and more outliers are detected for oranges (8.31).

TABLE III. EXPERIMENTAL RESULTS OF THE DL ALGORITHMS

Fruit classes	**DNN**			
	Apples vs. orange		***Apples vs. peaches***	
Epoch number	10	20	10	20
Test accuracy	0.844	0.848	0.694	0.746
Confusion matrices	$\begin{bmatrix}233 & 33\\74 & 201\end{bmatrix}$	$\begin{bmatrix}228 & 33\\74 & 208\end{bmatrix}$	$\begin{bmatrix}173 & 93\\42 & 134\end{bmatrix}$	$\begin{bmatrix}201 & 65\\47 & 129\end{bmatrix}$

TABLE IV. EXPERIMENTAL RESULTS OF THE CNN ALGORITMS

Fruit classes	**CNN**			
	Apples vs. orange		***Apples vs. peaches***	
Epoch number	10	20	10	20
Test accuracy	0.912	0.918	0.816	0.823
Confusion matrices	$\begin{bmatrix}234 & 32\\13 & 235\end{bmatrix}$	$\begin{bmatrix}245 & 21\\21 & 227\end{bmatrix}$	$\begin{bmatrix}205 & 61\\20 & 126\end{bmatrix}$	$\begin{bmatrix}213 & 53\\25 & 151\end{bmatrix}$

TABLE V. EXPERIMENTAL RESULTS OF THE ALEXNET ALGORITMS

Fruit classes	**AlexNet**			
	Apples vs. orange		***Apples vs. peaches***	
Epoch number	10	20	10	20
Test accuracy	0.795	0.931	0.794	0.744
Confusion matrices	$\begin{bmatrix}237 & 29\\76 & 172\end{bmatrix}$	$\begin{bmatrix}251 & 15\\74 & 228\end{bmatrix}$	$\begin{bmatrix}188 & 78\\13 & 163\end{bmatrix}$	$\begin{bmatrix}176 & 90\\23 & 153\end{bmatrix}$

TABLE VI. EXPERIMENTAL RESULTS OF THE GOOGLENET ALGORITMS

	GoogLeNet			
Fruit classes	***Apples vs. orange***		***Apples vs. peaches***	
Epoch number	5	10	5	10
Test accuracy	0.911	0.922	0.816	0.861
Confusion matrices	$\begin{bmatrix}259 & 7\\39 & 209\end{bmatrix}$	$\begin{bmatrix}257 & 9\\31 & 217\end{bmatrix}$	$\begin{bmatrix}204 & 62\\19 & 157\end{bmatrix}$	$\begin{bmatrix}247 & 19\\42 & 134\end{bmatrix}$

All models were trained with 10 and 20 epochs, and the performance of classification in terms of accuracy was followed. The highest accuracy is obtained for simple CNN for apples vs. oranges. The pre-train CNN AlexNet does not give good results for a reduced number of epochs, these being trained on the ImageNet dataset do not learn enough when the rate of learning is small. We can't say the same thing about GoogLeNet CNN, it has the same output as simple CNN, and we also note that the results are obtained from the same combination. For a DL network without convolutional layers, the results are weaker.

From Table 2, we notice that the higher difference in density edges is apples vs. oranges, and from Tables 4 and 6, we observe that the best accuracy is obtained from the same combination.

The main objective of this was accomplished, and four DL algorithms were tested on three fruit classes.

CONCLUSION

This paper presents an investigation into the DL and transfer learning of fruit classification. The study was carried out using three datasets: apples, oranges, and peaches. The density of edges in images can influence the classification of images. We used transfer learning with GoogLeNet and AlexNet CNNs to develop a robust model for assessing the classification of fruits. After changing the number of epochs, enhanced performance of DL algorithms was observed. We achieved a higher accuracy of 0.918 with simple CNN, and GoogLeNet transfer learning model produced an accuracy of 0.922. In the future, we aim to increase the variety of class objects so that the best accuracy can be detected for classification. Furthermore, we also aim to tune CNNs and investigate the effects of different parameters on CNNs.

REFERENCES

[1] S. Moldovanu and F. Batog, “The monitoring of burning buildings with convolutional neural network”, System Theory, Control and Computing Journal, vol. 3(2), pp. 1-8, (2023):

[2] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature, vol. 521 pp. 436–444 (2015) 436.

[3] C.T Cheng, T.Y. Ho, T.Y. Lee, C.C. Chang, C.C Chou, C.C. Chen, I. Chung and C.H. Liao, “Application of a deep learning algorithm for detection and visualization of hip fractures on plain pelvic radiographs”. Eur. Radiol. vol. 295, pp.469–5477, 2019.

[4] R. Dandavate and V. Patodkar, “CNN and data augmentation-based fruit classification model”, 4th Int. Conf. on I-SMAC (IoT in Social, Mobile, Analytics and Cloud), pp. 784–787, 2020.

[5] A. Kausar, M. Sharif, J. Park and D.R. Shin, “Pure-CNN: A framework for fruit images classification”, Int. Conf. on Computational Science and Computational Intelligence (CSCI) (2018), pp. 404-408

[6] S.S.S. Palakodati, V.R Chirra, Y. Dasari and S. Bulla, “Fresh and Rotten Fruits Classification Using CNN and Transfer Learning”. Rev. d’Intelligence Artif. vol. 34, pp. 617–622, 2020.

[7] M. Momeny, A. Jahanbakhshi, K. Jafarnezhad and Y.D. Zhang, “Accurate classification of cherry fruit using deep CNN based on hybrid pooling approach”. Postharvest Biol. Technol. vol. 166, pp. 111204, 2020.

[8] U. Amin, M. I. Shahzad, A. Shahzad, M. Shahzad, U. Khan et al., “Automatic fruits freshness classification using CNN and transfer learning,” Appl. Sci., vol. 13(14), pp. 1–17, 2023.

[9] H. Kang, and C. Chen, “Fruit detection, segmentation and 3D visualisation of environments in apple orchards”. Comput. Electron. Agric. vol. 171, pp. 105302, 2020,

[10] M Miron, S Moldovanu, B.I. Ștefănescu, M. Culea, and S.M. Pavel, Culea-Florescu, A.L. “A new approach in detectability of microcalcifications in the placenta during pregnancy using textural features and k-nearest neighbors algorithm”. J. Imaging, vol. 8, pp. 81, 2022

[11] T. Dewi, R. Rusdianasari, RD. Kusumanto, and S. Siproni, “Image Processing Application on Automatic Fruit Detection for Agriculture Industry,” Atlantis Highlights in Engineering, vol. 9, pp. 47-53, 2022.

[12] A.R. Ali, J. Li and S.J. O'Shea, "Towards the automatic detection of skin lesion shape asymmetry, color variegation and diameter in dermoscopic images". PLoS One, vol. 15, e0234352, 2020.

[13] S. Moldovanu, L. Moraru and D. Bibicu "Characterization of myocardium muscle biostructure using first order features", Dig. J. Nanomater Bios., vol. 6, pp. 1357-1365, 2011.

[14] F.A. Damian, S. Moldovanu, N. Dey, A.S. Ashour and L. Moraru, "Feature Selection of Non-Dermoscopic Skin Lesion Images for Nevus and Melanoma Classification". Computation.vol. 8, pp. 41, 2020.

[15] T.Shanthi, and R.S. Sabeenian, "Modified Alexnet architecture for classification of diabetic retinopathy images". Comput. Electr. Eng. vol. 76, pp. 56–64, 2019

[16] Y. Cui, F. Zhou, J. Wang, X. Liu, Y. Lin and S. Belongie, "Kernel pooling for convolutional neural networks", IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2921-2930, 2001.