International Journal of Engineering Trends and Technology
ISSN: 2231 – 5381 /doi:10.14445/22315381/IJETT-V69I7P207
Volume 69 Issue 7, 51-55, July, 2021

Improving the Prediction of Rotten Fruit Using Convolutional Neural Network

Sumitra Nuanmeesri[1], Lap Poomhiran[2], Kunalai Ploydanai[3]

[1]*Faculty of Science and Technology, Suan Sunandha Rajabhat University, Bangkok 10300, Thailand*

[2]*Faculty of Information Technology and Digital Innovation, King Mongkut's University of Technology North Bangkok, Bangkok 10800, Thailand*

[3]*Faculty of Agro-Industry, Kasetsart University, Bangkok 10900, Thailand*

[1]sumitra.nu@ssru.ac.th, [2]lap.p@kmutnb.ac.th, [3]kunalai.p@ku.ac.th

Abstract *— Separating rotten fruit is a necessary process that helps build trust and credibility before sending fresh products to consumers. This work proposed improving the model for predicting the rotten fruit that applied deep learning technique. The developed models were built using the VGG16 Convolutional Neural Network architecture with 3,000 images of fresh and rotten fruits for training, validating, and testing. The result showed that the developed model from the concatenated images dataset, which applied the RGB, Laplacian o Gaussian, GrabCut algorithm, and Hue Saturation Value with Adaptive Gaussian Thresholding, gave the higher model efficiency than the model that made from the RGB images dataset. This model has a validation accuracy of 89.97% and the testing accuracy of 91.33% for predicting the fresh or rotten red fruit.*

Keywords — *Concatenated images, Convolutional Neural Network, Rotten fruit*

I. INTRODUCTION

In this era, fruits continue to be popular and in demand in the market, especially people who want to take care of their health. Because fruits are the dietary supplement that contains nutrients such as protein, fat, vitamins, minerals, and fibers, which are suitable for health. Some people eat some fruit instead of the main meals, such as dinner, to help with weight control. Also, the fruits can be eaten immediately without any cooking process, making them a favorite for consumers.

Each fruit is originated or cultivated in different locations worldwide according to the species and environment favorable to yield. In the same way, the demand for consumption is also around the world as the population continues to rise. This resulted in the transportation of many varieties of fruit spread all over the world. Sometimes it takes several days to transport and causes fruit to spoil. Although nowadays, transportation technology is modern and fast. Nevertheless, some environments, such as temperature and humidity, might result in fruit spoiling faster than usual. Additionally, if one fruit begins to spoil, it can become a breeding ground for bacteria, spreading and eventually causing other fruits to spoil. Therefore, separating the rotten fruit from packaging or fruit crate is essential first before consumers consume it.

Many decades ago, workers from the fruit farm carried out the fruit classification and sorting process. Also, the market vendors further classified even the products in the distribution area or marketplace. The process is based on the human operator for the fruit classification, which takes time in the selecting process and is slow. It also increases production and distribution costs, which might result in higher fruit prices. Additionally, fruit separation by the manual and naked eye from humans, which are the traditional methods, can be inaccurate due to physical exhaustion during the working day. Thus, in the past few years, automatic fruit classification processes are gaining great attention and playing an important role, especially in the food industry, which requires massive production and reducing human labor costs.

The automatic fruit classification processes were starting from the fruit harvesting stage. The automated system will analyze and verify the suitability of the fruit to be consumed until the process of sizing and sorting fruit into packaging. The most automated system mentioned relies on machine learning principles that allow the machine to be trained until the recognition is done or accurate. This technique requires a large amount of visual input or images for training and testing purposes.

Currently, much research has been conducted on fruit classification and recognition. For example, image processing is applied to detect and classify diseases on fruit [1], while Balci et al. [2] assessed the Napoleon cherry number and dimension. Some studies applied Convolution Neural Network (CNN) for fruit detection [3][4] and fruit classification [5][6][7], including real-time classification by applied the image processing [8]. Thus, there are several studies focused on fruit classification and disease recognition by using image processing techniques, such as K-Means [9], Median Filter [10], Watershed [11], GrabCut algorithm [12], and Gray-level Co-occurrence Matrix [13].

Therefore, this research aims to improve and develop models for predicting freshness and rotten fruit based on a combination of image processing and CNN. The details of the development method are in Section 2, Section 3 explains the experiment results, and then summarized in Section 4.

II. METHODOLOGY

Developing the improvement model for predicting and classifying fresh fruit and rotten fruit in this research consisted of 1) images dataset collection, 2) images processing, 3) CNN modeling, and 4) model evaluation. These processes were described as follows.

A. Images Dataset Collection

In this study, the authors collected photographic data of three red fruits: red apple, rose apple, and rambutan. Each fruit contains 500 images of fresh fruit and 500 rotten fruits. Thus, there are 3,000 images in total were applied to generate the CNN model. The researchers took each image with varying widths and height, ranging from 640 pixels up to 1,920 pixels in multiple angles through the camera on any smartphone, no matter the model or brand.

B. Images Processing

Before starting to build the CNN model, the researchers processed two sets of images for use in modeling as follows:

a) Red Green Blue (RGB) images: All 3,000 photos of fresh and rotten fruit in RGB color were cropped into a square to center the fruit. Then resize the image size to only 224x224 pixels, as shown in Fig. 1.

Fig. 1 The examples of the RGB images dataset

b) Concatenate images: The RGB images dataset was used to produce a concatenated image by applying three image processing techniques, including Laplacian of Gaussian (LoG), grayscale without background, and Hue Saturation Value (HSV) with Adaptive Gaussian Thresholding (AGT) as follows:

First, the LoG image processing this technique applied the Laplacian derivatives of the coordinate x and y pixels in (1) [14] with a filter dimension of 3x3 pixels in Hessian matrix [15], as in (2) [16]. The output image was illustrated in Fig. 2.

$$LoG(x,y) = \frac{\partial^2(x,y)}{\partial x^2} + \frac{\partial^2(x,y)}{\partial y^2} \qquad (1)$$

$$Filter_{3\times3} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \qquad (2)$$

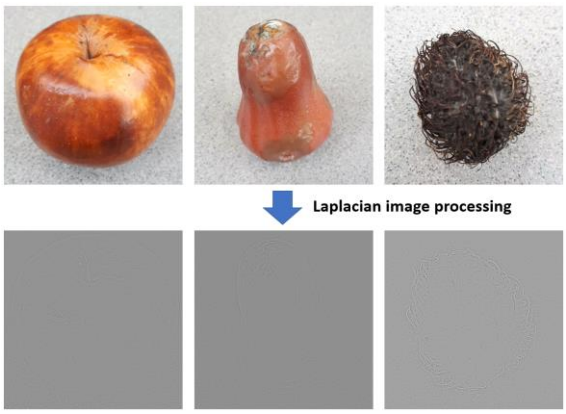

Fig. 2 The output of the LoG image processing

Second, the grayscale without background image was applied using the GrabCut algorithm [17]. This technique conducted the Gaussian Mixture Model (GMM) to learn and generate the new pixel distributed in a graph for separating the foreground and background pixels. The weights of the graph were used to determine until the final pixel of the background was classified. Therefore, the RGB image has eliminated the background information then converted it to a grayscale image, as in Fig. 3.

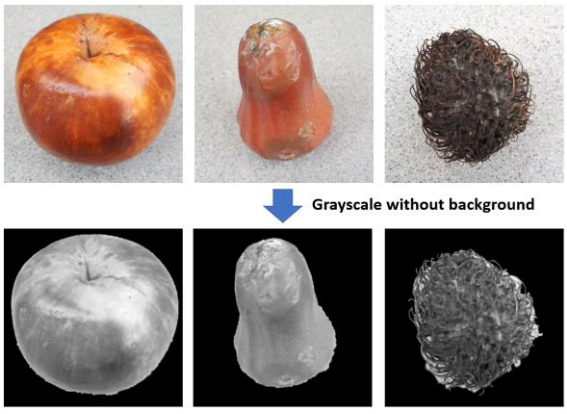

Fig. 3 The output of the grayscale image without background information

Last, HSV with AGT image processing was conducted to find the region of interest (ROI) for the rotten area on fruit and reduce the glare and incident light on the fruit surface. At the starting process, the authors conducted the RGB without a background image that was generated in the previous state to control the red color on the fruit surface in HSV color by bitwise the pixels range between 61 to 255 degrees. Then convert the pixels to the grayscale mode for processing with AGT. The AGT or Adaptive Gaussian Thresholding [18] was applied for eliminated the noises [19] and illumination in different surface areas. In this process, the grayscale image with HSV processed was blur by using the Gaussian blur technique and adjusted the

thresholding value to 255 degrees. Thus, only pixels whose color differs from the main surface of the fruit and does not glare will be left. The example of the output image was shown in Fig. 4.

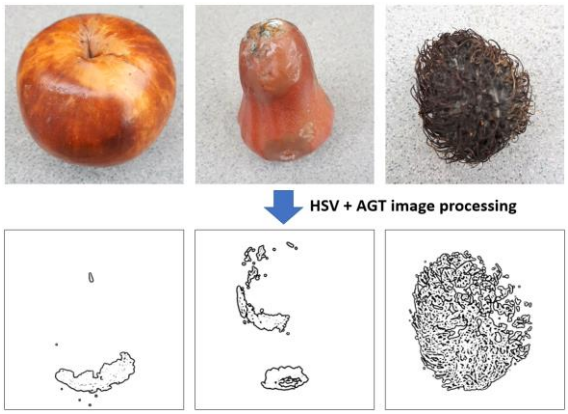

Fig. 4 The output of HSV with AGT image processing

All three processed images were combined with RGB images to create a single image using a concatenated images technique in a square shape. Then resize these images to 224x224 pixels for use in creating the CNN modeling process. The final output of concatenated image is shown in Fig. 5.

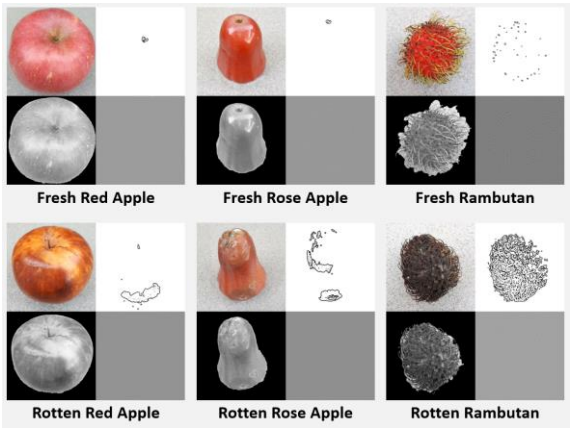

Fig. 5 The example of concatenated image dataset

C. CNN Modeling

At this stage, two datasets were used to build the CNN models. Each of the two datasets was split into 70% (2,100 images), 10% (300 images), and 20% (600 images) for training, validating, and testing the models, respectively. The authors applied the VGG16 model, which was presented by Simonyan and Zisserman [20], for the CNN modeling. The developed model consisting of thirteen convolutional layers with Rectified Linear Unit (ReLU) activation function, five max-pooling layers, a single flatten layer, and three dense layers. The last dense layer includes the softmax function for classifying the class output possible to one of six classes. The total trainable parameter was 134,285,126 paramters. The developed model architecture was described in Fig. 6.

```
Model: "Rotten Fruits CNN VGG-16"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
rescale (Rescaling)          (None, 224, 224, 3)       0
_________________________________________________________________
conv2d_1_1 (Conv2D)          (None, 224, 224, 64)      1792
_________________________________________________________________
conv2d_1_2 (Conv2D)          (None, 224, 224, 64)      36928
_________________________________________________________________
maxpooling1 (MaxPooling2D)   (None, 112, 112, 64)      0
_________________________________________________________________
conv2d_2_1 (Conv2D)          (None, 112, 112, 128)     73856
_________________________________________________________________
conv2d_2_2 (Conv2D)          (None, 112, 112, 128)     147584
_________________________________________________________________
maxpooling2 (MaxPooling2D)   (None, 56, 56, 128)       0
_________________________________________________________________
conv2d_3_1 (Conv2D)          (None, 56, 56, 256)       295168
_________________________________________________________________
conv2d_3_2 (Conv2D)          (None, 56, 56, 256)       590080
_________________________________________________________________
conv2d_3_3 (Conv2D)          (None, 56, 56, 256)       590080
_________________________________________________________________
maxpooling3 (MaxPooling2D)   (None, 28, 28, 256)       0
_________________________________________________________________
conv2d_4_1 (Conv2D)          (None, 28, 28, 512)       1180160
_________________________________________________________________
conv2d_4_2 (Conv2D)          (None, 28, 28, 512)       2359808
_________________________________________________________________
conv2d_4_3 (Conv2D)          (None, 28, 28, 512)       2359808
_________________________________________________________________
maxpooling4 (MaxPooling2D)   (None, 14, 14, 512)       0
_________________________________________________________________
conv2d_5_1 (Conv2D)          (None, 14, 14, 512)       2359808
_________________________________________________________________
conv2d_5_2 (Conv2D)          (None, 14, 14, 512)       2359808
_________________________________________________________________
conv2d_5_3 (Conv2D)          (None, 14, 14, 512)       2359808
_________________________________________________________________
maxpooling5 (MaxPooling2D)   (None, 7, 7, 512)         0
_________________________________________________________________
flatten (Flatten)            (None, 25088)             0
_________________________________________________________________
fc_1 (Dense)                 (None, 4096)              102764544
_________________________________________________________________
fc_2 (Dense)                 (None, 4096)              16781312
_________________________________________________________________
softmax (Dense)              (None, 6)                 24582
=================================================================
Total params: 134,285,126
Trainable params: 134,285,126
Non-trainable params: 0
_________________________________________________________________
```

Fig. 6 The proposed CNN model architecture of the rotten fruit prediction

For the processing of CNN modeling, it was set the batch size of 32 with a learning rate of 0.001. A training epoch was defined at 200 epochs while the early stopping was activated to solve the overfitting during the training process.

D. Model Evaluation

The developed models were evaluated during the model testing processing in terms of accuracy, precision, recall, and F-measure, which were calculated in (3), (4), (5), and (6) [21][22][23], respectively.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (3)$$

$$Precision = \frac{TP}{TP + FP} \quad (4)$$

$$Recall = \frac{TP}{TP + FN} \quad (5)$$

$$F - measure = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (6)$$

Where:
TP refers to predicted positive, and it is true;
TN refers to predicted negative, and it is true;
FP refers to predicted positive, and it is false;

FN refers to predicted negative, and it is false.

Further, the models were evaluated using the normalized confusion matrix and cross-entropy function during the validating processing. The cross-entropy was calculated in (7) [24][25][26].

$$Cross-entropy = -\sum_{i=1}^{m}\sum_{j=1}^{n} y_{i,j} \log\left(p_{i,j}\right) \quad (7)$$

Where:

m represents the number of inputs;

n represents the number of classes;

$y_{i,j}$ represents the input i on class j;

$p_{i,j}$ represents the prediction of the probability of input i on class j.

II. RESULTS

Two models were developed base on different datasets and were evaluated in term of cross-entropy validation, the model testing effectiveness, and normalized confusion matrix as follows:

A. The Result of Cross-Entropy Validation

The model with the RGB images dataset was stopped training at the end of the 34th epochs. The validation accuracy was 77.21%, and the validation loss was 8.13%. For the model with the concatenated images dataset, this model was stopped after training through the 75th epochs. The model gave the validation accuracy of 89.97% and the validation loss of 4.98%. The comparison results of the validation accuracy and validation loss of the developed model with the RGB images dataset and the concatenated images dataset was shown in Fig. 7.

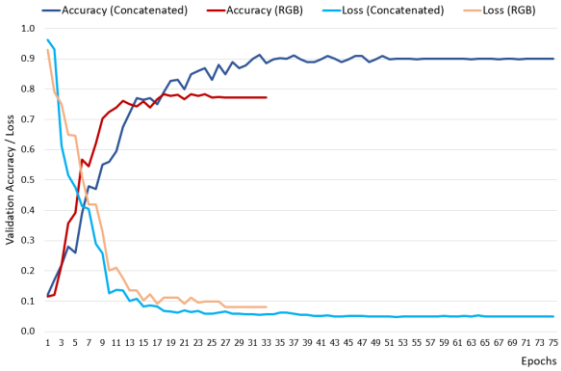

Fig. 7 The comparison of the validating models

B. The Result of the Model Testing Effectiveness

Both models were tested with 600 images in terms of accuracy, precision, recall, and F-measure. The result of the model comparison was shown in Table I. The model developed with the concatenated images dataset gave the accuracy was 91.33%, the precision was 91.44%, the recall was 90.82%, and the F-measure was 91.13%. While the model which was built from the RGB images dataset gave the model effectiveness values were 79.67%, 81.03%, 77.81%, and 79.39% for the accuracy, precision, recall, and F-measure, respectively.

TABLE I
THE MODEL TESTING EFFECTIVENESS RESULTS

Dataset	Accuracy (%)	Precision (%)	Recall (%)	F-measure (%)
RGB images	79.67	81.03	77.81	79.39
Concatenated images	91.33	91.44	90.82	91.13

C. The Result of Normalized Confusion Matrix

The results of the models tested and normalized to the confusion matrix were shown in Fig. 8 and Fig. 9 for the RGB images dataset and concatenated image dataset, respectively.

Fig. 8 The model testing result of the RGB image dataset

Fig. 9 The model testing result of the concatenated image dataset

According to Fig. 8 and Fig. 9, the model generated from the concatenated images dataset gave the accuracy higher than the model from the RGB images dataset.

III. CONCLUSION

There is a wide range of research and innovations in fruit inspection and sorting, ranging from cultivation, harvesting, packaging, and transportation from farm to store to distribute products to consumers. Nevertheless, sometimes, the unsuitable environment results in fruit spoilage before delivery to customers. This research developed two models for fruit recognition and predicting

the freshness or spoilage of fruit using the CNN approach based on VGG16 architecture. The first model was developed from the RGB images dataset, while the second model has developed from the concatenated images dataset, including RGB image, LoG image, grayscale without background image, and HSV with AGT image.

The research found that the red fruit consisting of red apple, rose apple, and rambutan could be determined freshness or rotten fruit using the CNN model effectively because it could be considered the different conditions on the fruit surface and spoiled area. Furthermore, the result showed that the developed model generated from the concatenated images dataset is suitable for predicting the rotten fruit in this work. This model gave the validation accuracy of 89.97% and the validation loss of 4.98% during the validation processing. In the model testing processing, the accuracy was 91.33%, the precision was 91.44%, the recall was 90.82%, and the F-measure was 91.13% for this model, which was higher than the model effectiveness from the RGB images dataset. It could be said that the concatenated images could improve the prediction of rotten red fruit using the convolutional neural network.

For further work, the image augmentation would be conducted to a new dataset and build several model architectures for comparison the model efficiency. A suitable developed model would be applied for hardware development on the Internet of Things-based and implement for the food industry.

ACKNOWLEDGMENT

The authors are grateful to the Institute for Research and Development, the Faculty of Science at Suan Sunandha Rajabhat University, the Faculty of Information Technology and Digital Innovation at King Mongkut's University of Technology North Bangkok, and the Faculty of Agro-Industry at Kasetsart University, who supported and gave this research opportunity.

REFERENCES

[1] S. R. Dubey and A. S. Jalal., Application of image processing in fruit and vegetable analysis: A review, Journal of Intelligent Systems, 24(4) (2015) 405–424.

[2] M. Balci, A. A. Altun, and S. Taşdemir., Görüntü leme teknikleri kullanlarak napolyon tipi kirazlarn snflandrlmas, SelçukTeknik Dergisi, 15(3) (2016) 221–237.

[3] I. Sa, Z. Ge, F. Dayoub, B. Upcroft, T. Perez, and C. McCool., DeepFruits: A fruit detection system using deep neural networks, Sensors, 16(8) (2016) 1222.

[4] Y. Yu, K. Zhang, L. Yang, and D. Zhang., Fruit detection for strawberry harvesting robot in non-structural environment based on Mask-RCNN, Computers and Electronics in Agriculture, 163 (2019) 104846.

[5] L. Hou, Q. Wu, Q. Sun, H. Yang, and P. Li., Fruit recognition based on convolution neural network, in Proc. 12th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery, (2016) 18–22.

[6] S. H. Wang and Y. Chen., Fruit category classification via an eight-layer convolutional neural network with parametric rectified linear unit and dropout technique, Springer, 79 (2020) 15117–15133.

[7] B. K. Savaş and Y. Becerikli., Real time driver fatigue detection system based on multi-task CNN, IEEE Access, 8 (2020) 12491–12498.

[8] I. B. Akinci., The real-time automation system for classifying fruits with image processing, Com. Eng. thesis, Fen Bilimleri Enstitüsü, Karabük Üniversitesi, Turkey, (2017).

[9] M. T. Habib, A. Majumder, A. Z. M. Jakaria, M. Akter, M. S. Uddin, and F. Ahmed., Machine vision based papaya disease recognition, Journal of King Saud University - Computer and Information Sciences, 32(3) (2020) 300–309.

[10] Y. H. S. Kumar and G. Suhas., Identification and classification of fruit diseases, in Santosh K., Hangarge M., Bevilacqua V., Negi A., (Eds) Recent Trends in Image Processing and Pattern Recognition, Communications in Computer and Information Science, 709 (2017) 382–390.

[11] A. Nosseir and S. E. A. Ahmed., Automatic identification and classifications for fruits using k-NN, in Proc. ACM International Conference on Software and Information Engineering, (2018) 62–67.

[12] F. Ye, X. Lou, and M. Han., Evolving support vector machine using modified fruit fly optimization algorithm and genetic algorithm for binary classification problem, in Proc. 13th International Computer Conference on Wavelet Active Media Technology and Information Processing, (2016) 38–46.

[13] S. Jana, S. Basak, and R. Parekh., Automatic fruit recognition from natural images using color and texture features, in Proc.Devices for Integrated Circuit, (2017) 620–624.

[14] G.E. Sotak, K.L. Boyer., The Laplacian-of-Gaussian kernel: A formal analysis and design procedure for fast, accurate convolution and full-frame output, Computer Vision, Graphics, and Image Processing, 48(2) (1989) 147–189.

[15] B. A. Pearlmutter., Fast exact multiplication by the Hessian, Neural Computation, 6(1) (1993) 147–160.

[16] (2021) OpenCV: Image Gradients. [Online]. Available: https://docs.opencv.org/master/d5/d0f/tutorial_py_gradients.html

[17] G. Rother, V. Kolmogorov, and A. Blake., GrabCut: Interactive foreground extraction using iterated graph cuts, in Proc. ACM SIGGRAPH Papers, (2004) 309–314.

[18] J. W. Long, X. J. Shen, H. Zang, and H. P. Chen., An adaptive thresholding algorithm by background estimation in Gaussian scale space, Zidonghua Xuebao/Acta Automatica Sinica, 40(8) (2014) 1773–1782.

[19] D. N. H. Thanh, U. Erkan, V. B. S. Prasath, V. Kumar, and N. N. Hien., A skin lesion segmentation method for dermoscopic images based on adaptive thresholding with normalization of color models, in Proc. 6th International Conference on Electrical and Electronics Engineering, (2019) 116–120.

[20] K. Simonyan and A. Zisserman., Very deep convolutional networks for large-scale image recognition, arXiv preprint, arXiv:1409.1556, (2015).

[21] S. Nuanmeesri., Development of community tourism enhancement in emerging cities using gamification and adaptive tourism recommendation, Journal of King Saud University - Computer and Information Sciences, (in press), (2021).

[22] S. Nuanmeesri and W. Sriurai.,Thai water buffalo disease analysis with the application of feature selection technique and Multi-Layer Perceptron Neural Network, Engineering, Technology & Applied Science Research, 11(2) (2021) 6907–6911.

[23] S. Nuanmeesri, W. Sriurai, and N. Lamsamut., Stroke patients classification using resampling techniques and decision tree learning, International Journal of Engineering Trends and Technology, 69(6) (2021) 115–120.

[24] K. Janocha and W. M. Czarnecki., On loss functions for deep neural networks in classification, Schedae Informaticae, 25 (2016) 49–59.

[25] Z. Zhang and M. R. Sabuncu., Generalized cross entropy loss for training deep neural networks with noisy labels, in Proc. the 32nd Conference on Neural Information Processing Systems, (2018).

[26] Q. Zhu, Z. He, T. Zhang, and W. Cui., Improving classification performance of softmax loss function based on scalable batch normalization, Applied Sciences, 10(8) (2020) 29–50.