GOCE DELCEV UNIVERSITY - STIP
FACULTY OF COMPUTER SCIENCE

The journal is indexed in
EBSCO

ISSN 2545-4803 on line
DOI: 10.46763/BJAMI

BALKAN JOURNAL OF APPLIED MATHEMATICS AND INFORMATICS (BJAMI)

BALKAN JOURNAL
OF APPLIED MATHEMATICS AND INFORMATICS

YEAR 2022

VOLUME V, Number 2

AIMS AND SCOPE:
BJAMI publishes original research articles in the areas of applied mathematics and informatics.

Topics:
1. Computer science;
2. Computer and software engineering;
3. Information technology;
4. Computer security;
5. Electrical engineering;
6. Telecommunication;
7. Mathematics and its applications;
8. Articles of interdisciplinary of computer and information sciences with education, economics, environmental, health, and engineering.

Address of the editorial office
Goce Delcev University – Stip
Faculty of philology
Krste Misirkov 10-A
PO box 201, 2000 Štip,
Republic of North Macedonia

BALKAN JOURNAL
OF APPLIED MATHEMATICS AND INFORMATICS (BJAMI), Vol 5

ISSN 2545-4803 on line
Vol. 5, No. 2, Year 2022

EDITORIAL BOARD

CONTENT

Balkan Journal of Applied Mathematics and Informatics
Volume V Number 2 Year 2022
Online ISSN 2545-4803
UDC: 632.03:634.8.074]:[004.891:004.932.72’1

GRAPE LEAVES DISEASE RECOGNITION USING AMAZON SAGE MAKER

ANGELA TOCKOVA, ZORAN ZLATEV, SASO KOCESKI

Abstract. Agriculture has faced multiple challenges throughout 20th and 21st century. Population growth means more mouths to feed, but, unfortunately, smaller rural labor force, which stimulated people to develop more efficient and sustainable production methods in order to adapt plants to different climate changes and make them more resistible to diseases. Accurate and rapid diagnosis of plant diseases has shown to play an important role in improving the quality and quantity of agriculture crops. In this article we are proposing an automated disease detection and classification service for grape leaves using traditional image processing and AI techniques using Amazon Sage Maker service. We are focusing our work on detecting three different diseases of grape leaves: black rot, black measles, and leaf blight disease. The results of our study are presented and discussed in the following.

1. Introduction

Grapes are one of the highest demanded table fruits in Europe after bananas. In 2019 Europe produced over 24.2 million tons of grapes with Italy and France at the top of grape production, but only 1.7 million were used as table grapes for fresh, raw consumption.

Moreover, the development of wine industry increased the demand for the production of high-quality grapes. Today, wine industry is a significant business sector having wine markets around the whole world. Europe is marked as the highest per capita wine consumption market with over 35 liters per person per year. The second biggest wine consumption market is Australia with 23.9 liters per person, and the US has a significantly lower consumption with 9.9 liters and China with just 3.5 liters per person [1]. Research shows that in the last three decades, wine industry has grown tremendously not only with the wine consumption, but also with the growth of wine tourism. Vineyards in combination with wineries have become a very popular tourist attraction especially in the 00’s. Studies from US showed that wineries and wine festivals drew over 16.3 million visitors in 2001, who spent over $2 billion [2]. Wine tourism and wine industry have a very big impact on the local economy of many countries in the world such as Italy, Spain, France, USA, China etc.

The economy of North Macedonia is highly dependent on agriculture and its productivity. In 2020, the share of agriculture in North Macedonia's gross domestic product (GDP) was around 10% percent. The wine and viticulture sectors are one of the most important sectors in the agriculture in the country contributing with 17-20% in the agriculture GDP. Around 100.000 people are directly engaged within this sector like wineries, winemakers, viticulturists, and employees of the companies which are producing the materials for the respective industry.

However, like many other plants, grapes are very sensitive to climate, bacteria, fungi and viruses. Grape diseases cause significant crop losses, reduce the quality of berries with impact on sugar content and pigmentation and can also shorten the longevity of one grape plant. We now know about 65 viruses that can attack grape plants; not all of them cause big economic damages, but there are some types that seriously affect the global grape industry. Some of the most well-known grape diseases are leafroll disease, black rot, measles etc. [3].

Diagnosing a disease on plants can be difficult because farmers should be able to protect the plant before any disease becomes visually recognizable. Farmers usually first observe the visual symptoms of a disease in plants, but at that point it can already be too late to save the plant. On the other hand, experts can easily diagnose diseases with diagnostic tests made in laboratories, but the most common practice is detection with naked eye, which is what most farmers use. This practice can very often lead to wrong predictions and diagnosis; that is why there is a need of automatic disease detection which can detect the disease in its early stage without the help of laboratory tests and expert observations.

However, in the last decade, the concept of automation in agriculture is permanently advancing with a fast pace mainly due to the development of Machine Learning and Artificial Intelligence (AI). The growing popularity of the Artificial Intelligence (AI) and its application in various fields [3], starting from tourism [4] through medicine [5-8], biology [9], education [10], robotics [11-14], and also economy [15], is mainly due to the apparatus, i.e., the models and techniques used to mimic human reasoning, learning and improving during time.

In this article we are proposing an automated disease detection and classification service for grape leaves using traditional image processing and AI techniques using the Amazon Sage Maker service [16]. We are focusing our work on detecting three different diseases on grape leaves: black rot, black measles and leaf blight disease. The results of our study are presented and discussed as follows.

2. Related research

Since the beginning of grapevines plantation, people made tremendous efforts to improve the production and quality of the grape. Of course, science took a great part in fighting the battle between farmers and plant diseases. Scientists know that the key is in early detecting and recognizing the disease that affects the plant, so a lot of research is focused on how the disease can be detected before it is too late.

On the other hand, with the rising of artificial intelligence, scientists were longing to implement AI in everyday life. Automated identification and classification of plant leaf disease was a breakthrough idea for the reduction of economic losses and the conservation of specific species. In 2012, Li et al. [17] proposed the method based on K_means clustering with designing a special Support vector machine (SVM) used as classifier. It used thirty-one selected features to recognize powdery and downy mildew disease with very good recognition rate of 93.33% and 90% accordingly. Later in 2016, Waghmare [18] also used multiclass SVM to detect diseased parts using a different technique through leaf texture analysis and pattern recognition. One leaf was used as an input, on which background removal was performed and then it was put into segmentation. Then, high pass filter analyzes the segmented leaf to detect the affected (sick) part. In the end, the extracted texture pattern is given to the multiclass SVM to make the recognition.

With a modern approach, Miaomiao et al. [19] tried to use Deep Learning techniques with united convolutional neural networks (CNN) based on an integrated method. This UnitedModel was pretrained to distinct several grape plant diseases such as esca, isariopsis and black rot compared to healthy leaves, using complementary discriminative features. This model compared to other CNN models achieved better results with the test accuracy of 98.57%.

In our study we are proposing an algorithm based on ResNet neural network. Moreover, we are trying to integrate the proposed algorithm in the form of a cloud service leveraging on AWS infrastructure.

3. Working methodology and environment

With cloud technology these days, e.g., AWS SageMaker, we can find efficient and practical solutions to agricultural problems [20]. As part of Amazon Web Services, SageMaker was introduced as a service that enables scientists to build, train and deploy machine learning models in the cloud. Being a cloud service, and working with almost 100% GUI makes every step of ML development easier to learn and use. Starting from data labeling, preparation of data, feature engineering, auto-ML, training, automatic model tuning, deployment, hosting etc., it offers a whole process to build a machine learning solution for our problem.

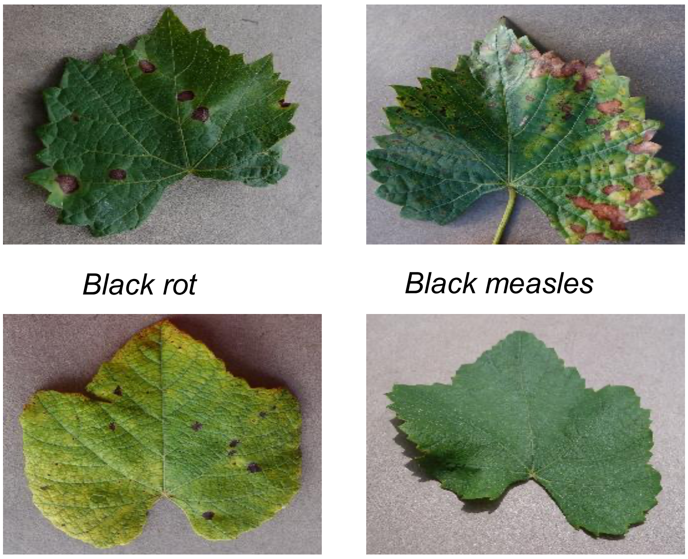
AWS SageMaker service is also very good for building scalable machine learning models, and it will provide incremental training as well as automatic hyperparameter optimization. In comparison with JVM-based model, this approach showed better performance on large datasets [21]. In our research, we will be using this service to develop and build our train machine model and then deploy the model for further use. Imagining a robot that will take pictures of plant's leaves, or maybe a case scenario when the farmer himself will want to know which disease he sees on his plant, the taken picture will be uploaded on the hosted environment where after examining the image, our software will give information about the disease of the plant with a percentage of accuracy. The software will also recognize healthy leaves where there is no disease visible.

4. Data Source

For developing the machine learning service, images are gathered from the public data source Plant Village Dataset [22]. We used raw, colored images for: Black rot, Black measles, Leaf blight and also images for healthy leaves. We had two hundred images of each disease and two hundred images of healthy leaves. We choose the same number of images per disease in order to have a balanced dataset which is crucial for making a good machine learning model. Imbalanced datasets can lead to false prediction and bad metric results.

Every image was in *.jpg* format and had the same size of 256 × 256 pixels. Images were classified within folders which were named after the disease the images represented, e.g.

- *input-images/black-rot*;
- *input-images/black-measles*;
- *input-images/leaf-blight;*
- *input-images/healthy*



Black rot *Black measles*

Leaf blight *Healthy*

Figure 1. Image examples of each category

After the preparation of example data, the process can continue within the AWS SageMaker. For the purpose of training, we created an Amazon SageMaker notebook instance which is a machine learning compute instance running the Jupyter Notebook App. Inside our notebook instance, we first uploaded our training data in the Files directory. The training data is being uploaded as we previously classified it in named folders, two hundred images for every type of disease, plus two hundred images in the Healthy leaves folder. After uploading all data, we continued the process by creating a Python script within the Jupyter Notebook which has a preinstalled environment for working with Python 3. This script is used to define the training and validation set, percentage of images which should be allocated for testing and validation and generating the training and validation data. In this script, we defined the split ratio for the validation set to be 20%. That means that 80% of the images will be used for training (160 images from each disease folder and 160 from the healthy leaves folder), and 20% (40 images from each disease folder and 40 from the healthy leaves folder) will be used to validate if the model did a good job in learning the image's features.

For the use of the training algorithm, we generated a list file in which every row represents one image and has three explanation columns: index, label (image class) and file name. This list file is stored in the S3 AWS storage. With creating the list file, we finished the first process of fetching, cleaning and preparing data. Now the second process starts, which will be in the hands of the AWS Residual neural network (ResNet) integration.

5. System architecture

In 2015 ResNet became the most groundbreaking work in the computer vision and deep learning. Its success was largely based on the fact that training on large number of layers was now successful without failing on the *"vanishing gradient"* problem. This vanishing gradient problem was an issue in every network architecture built with thousands of convolutional layers. To understand this problem, which was very usual then, we must explain the backpropagation process.

Backpropagation process is an algorithm for training neural networks based on gradient descent. On every iteration the algorithm calculates the error of how far the output from the actual output is, then it checks whether this error is minimized, if it is not, then the algorithm must update the weights and repeat the process until it finds the minimum error. This process is being done on every layer, and when we have a network with too many layers, the gradient becomes smaller and smaller until it "disappears".

Before ResNet, there were several ways invented to solve this vanishing gradient issue, but unfortunately none of them succeeded in solving the problem. Then a new theory was introduced by the creators [23] of ResNet in which they successfully managed to solve the problem by adding an "identity shortcut connection" which skips one or more layers. This idea is shown in the following figure:

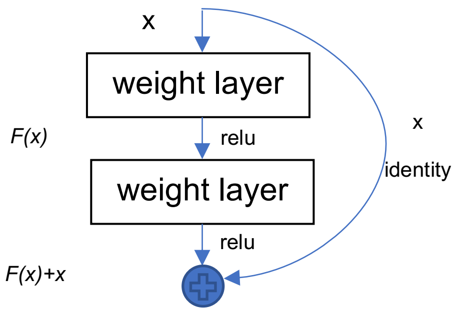


Figure 2. ResNet architecture

$$
\begin{aligned}
a^{L} &= x \\
z^{L+1} &= w^{L+1} \cdot a^{L} + b^{L+1} \\
a^{L+1} &= Relu(Z^{L+1}) \\
z^{L+2} &= w^{L+2} \cdot a^{L+1} + b^{L+2} \\
a^{L+2} &= z^{L+2} + a^{L} \leftrightarrow f(x) + x
\end{aligned}
$$

With stacking identity mappings or layers that actually do not do anything, the network would not degrade its performance and deeper model should not produce a higher training error. ResNet made its breakthrough when winning the prestigious CVRP 2016 Best Paper Award, and after that it took the first place on three tasks in the ImageNet competition. ResNet showed to reduce the top-1 error by 3.5 % compared with plain ImageNet. Over the past few years, there have been many variations of the network and the latest improvement comes from Amazon researchers who implemented this neural network in Amazon machine learning services.

6. Model training

To start training our model with the already prepared data, we first used the Estimator API. We create an estimator class in which we declare our number of training instances, their type and the output path. For this purpose, we created one instance with type ***ml.p2.xlarge*** which has 1 GPUs, 4 vCPUs, 61 GiB of RAM and high network bandwidth.

Table 1. Five best training jobs

#	Value	Name	Time	Learning Rate
18	0.979290	plant-disease-002-fe1389d3	392s	0.030373
3	0.978198	plant-disease-017-9b271b42	408s	0.006945
4	0.975783	plant-disease-016-ff8b5c57	413s	0.000215
1	0.974702	plant-disease-019-7db57420	409s	0.005196
11	0.973373	plant-disease-009-201f8054	466s	0.027701

After creating the estimator, we declare our training parameters, for the training we will use 18 layers and 20 parallel training jobs. Our image shape is already defined and it is 256 x 256 pixels. The number of training samples is the number of images we will fed to the model which in our case is 700 images, the number of classes is 4 as we train our model to find three types of diseases and also healthy leaves. We also define the minimum and maximum batch size, we set the minimum to 16 and the maximum to 64, and we set epoch number to 10. Batch size is the number of training examples utilized in one iteration, where epoch is the number of passes to the entire training dataset to the machine learning algorithm. Next, we set our minimum learning rate to 0.0001 and maximum learning rate to 1.

The next thing is configuring Hyperparameter Tuning Job using the previously defined parameters. This feature chooses a random combination of values from within the ranges that we specify for hyperparameters for each training job. The purpose of Hyperparameter tuning is to allow multiple training jobs with different values to be tried out seeking for the best training job with the best validation accuracy. After this, we run the hyperparameter tuning job. It took 8 hours, 8 min and 24 seconds to finish the training job.
After the job was finished, we printed the best 5 training jobs.

From this table we can see that the best training job is with number 18 and the value of 0.979290. After finding the best training job, we deploy the model.

7. Deploying the model

After finishing with training, we want to deploy the model to an endpoint. The endpoint will be a separate AWS instance which will contain our trained model and will be programmed to recognize every image that has been uploaded to the test-images folder within the Jupiter Notebook. For creating the model, we call the *create_model* function from the SageMaker API with the model name **plantdisease-training**. When the training model is created, we create the endpoint with the *create_endpoint* function and host the model on it. For endpoint configuration we choose one instance with the type ***ml.t2.medium*** and we name the endpoint **plant-disease-ep**. When the endpoint is created, we start doing tests to see how our trained model will respond to given images for recognition.

8. Experimental results and discussion

From the process of training the model, we could see very good results on every disease, and on the healthy leaves also. Our five best training jobs were with more than 97% accuracy, especially good on leaves with black measles and black rot with 99% accuracy. The training dataset of images contained images with good quality and good representation of the diseases. The quality and quantity of the dataset itself made big impact on the accuracy of the training model.

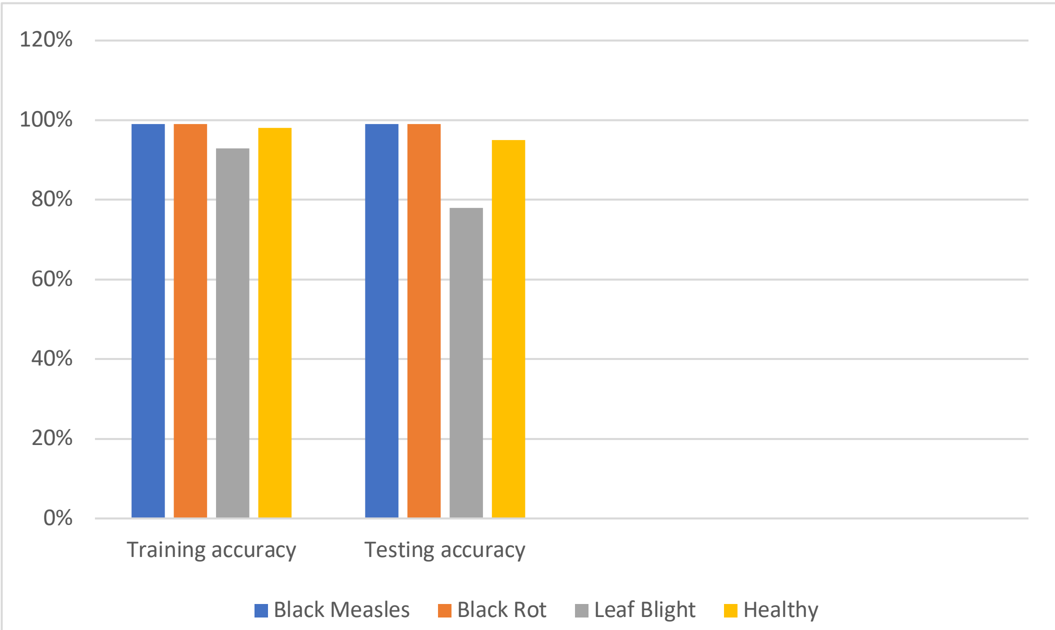


Figure 3. Training and testing accuracy

To finalize our research about how AWS SageMaker works with image recognition, we ran several tests using images with different size, different lighting, and low-quality images. We used random images of grape leaves downloaded from the internet and uploaded them to our test-images folder. From testing on random images, we can say that our model successfully recognized every time we uploaded an image with black rot disease, or black measles disease.

We also realized that the model sometimes gave wrong outputs when uploading an image where the disease was not very easily noticeable, so the result was a healthy leaf when in fact the leaf had a leaf-blight condition. We also noticed that when black rot disease is widespread on a bigger surface of the leaf, it is sometimes confused with the black-measles disease. Overall, we can conclude that our model, beside some wrongful outputs, proved to be accurate in most of our tested images.

```
File name: grape-leaf1.jpg Result: label - black-measles with confidence
0.9995668530464172
```

Figure 4. Results from testing

```
File name: grape-leaf2.jpg Result: label - black-rot with confidence
0.9996704520463182
```

Figure 5. Results from testing

```
File name: grape-leaf3.jpg Result: label - healthy with confidence
0.7883404620361192
```

Figure 6. Results from testing

In addition to the accuracy of the model, very important classification metrics are **precision** and **recall** and **F1-Score**. These other metrics are very important because they represent a more insightful picture which is not the performance of the model, but also which classes are predicted correctly and incorrectly. To calculate these metrics, we need four classification numbers: TP (true positive), FP (false positive), FN (false negative), FP (false positive) [24].

Black measles

$$precission = \frac{TP}{TP + FP} = \frac{19}{19 + 2} = \frac{19}{21} = 0.90 \ \ => 90\%$$

$$recall = \frac{TP}{TP + FN} = \frac{19}{19 + 0} = \frac{19}{19} = 1 => 100\%$$

$$F1 = 2 * \frac{precision * recall}{recision + recall} = 2 * \frac{0.90 * 1}{0.90 + 1} = 2 * \frac{0.90}{1.90} = 2 * 0.47 = 0.94$$

Black root

$$precission = \frac{TP}{TP + FP} = \frac{19}{19 + 1} = \frac{19}{20} = 0.95 \ \ => 95\%$$

$$recall = \frac{TP}{TP + FN} = \frac{19}{19 + 0} = \frac{19}{20} = 1 => 100\%$$

$$F1 = 2 * \frac{precision * recall}{recision + recall} = 2 * \frac{0.95 * 1}{0.95 + 1} = 2 * \frac{0.95}{1.95} = 2 * 0.49 = 0.98$$

Leaf blight

$$precission = \frac{TP}{TP + FP} = \frac{15}{15 + 4} = \frac{15}{19} = 0.78 \ \ => 78\%$$

$$recall = \frac{TP}{TP + FN} = \frac{15}{15 + 2} = \frac{15}{17} = 0.88 => 88\%$$

$$F1 = 2 * \frac{precision * recall}{recision + recall} = 2 * \frac{0.78 * 1}{0.78 + 1} = 2 * \frac{0.78}{1.78} = 2 * 0.44 = 0.88$$

9. Conclusion

In this work, we managed to develop an effective and “modern” approach to automatic grape diseases identification. What makes our approach effective is the use of the ResNet architecture which made a breakthrough with image recognition in the past years. With the ResNet networks the vanishing gradient issue is completely solved and there is no problem in building deeper models. The training error is reduced by 3.5% percent in comparison with ImageNet which is also shown in our training jobs with more than 97% accuracy.

On the other hand, what gives a modern feature to our work is that all the developing and testing is completely based on the cloud. Amazon Web Services give us almost unlimited resources of storage, computing power and speed to solve even the hardest deep learning problems with insignificant money and time. Using the SageMaker service we have built a scalable machine which will have no problems to work with large datasets. Also, we do not have to worry about the latest technology improvements because these services are maintained constantly by one of the best developers in the world.

Being on the cloud, our project is remotely accessible from everywhere, which is one of the key features in developing now. In addition, it can always be extended to other plant disease identification tasks and similar machine learning problems.

References

[1] García-Cortijo, María Carmen, Emiliano C. Villanueva, Juan Sebastián Castillo-Valero, and Yuanbo Li. "Wine consumption in China: Profiling the 21st Century Chinese wine consumer." Ciência e Técnica Vitivinícola 34, no. 2 (2019): 71-83.

[2] Foltz C.J, Woodall S., Wandschneider R. P., Garth T.R. (2007), The Contribution of Grape and Wine Industry to Idaho Economy: Agribusiness and Tourism Impacts

[3] Marcos F.B., Thor V.M.F, Pasquale S. (2017), Grapevine virus diseases: Economic impact and current advances in viral prospection and management, Revista Brasileira de Fruticultura 39. 10.1590/0100-29452017411.

[4] Loshkovska, Suzana, and Saso Koceski, eds. ICT innovations 2015: Emerging technologies for better living. Vol. 399. Springer, 2015.

[5] Koceski, Saso, and Biljana Petrevska. "Empirical evidence of contribution to e-tourism by application of personalized tourism recommendation system." Annals of the Alexandru Ioan Cuza University-Economics 59, no. 1 (2012): 363-374.

[6] Trajkovik, Vladimir, Elena Vlahu-Gjorgievska, Saso Koceski, and Igor Kulev. "General assisted living system architecture model." In International Conference on Mobile Networks and Management, pp. 329-343. Springer, Cham, 2014.

[7] Stojanov, Done, and Saso Koceski. "Topological MRI prostate segmentation method." In Computer Science and Information Systems (FedCSIS), 2014 Federated Conference on, pp. 219-225. IEEE, 2014

[8] Koceski, Saso, and Natasa Koceska. "Evaluation of an assistive telepresence robot for elderly healthcare." Journal of medical systems 40, no. 5 (2016): 121.

[9] Stojanov, Done, Aleksandra Mileva, and Sašo Koceski. "A new, space-efficient local pairwise alignment methodology."Advanced Studies in Biology 4, no. 2 (2012): 85-93.

[10] Koceski, Saso, and Natasa Koceska. "Challenges of videoconferencing distance education-a student perspective." International Journal of Information, Business and Management 5, no. 2 (2013): 274.

[11] Koceski, Saso, Natasa Koceska, and Ivica Kocev. "Design and evaluation of cell phone pointing interface for robot control." International Journal of Advanced Robotic Systems 9, no. 4 (2012): 135.

[12] Koceski, Saso, Stojanche Panov, Natasa Koceska, Pierluigi Beomonte Zobel, and Francesco Durante. "A novel quad harmony search algorithm for grid-based path finding." International Journal of Advanced Robotic Systems 11, no. 9 (2014): 144.

[13] Koceska, Natasa, Saso Koceski, Francesco Durante, Pierluigi Beomonte Zobel, and Terenziano Raparelli. "Control architecture of a 10 DOF lower limbs exoskeleton for gait rehabilitation." International Journal of Advanced Robotic Systems 10, no. 1 (2013): 68.

[14] Serafimov, Kire, Dimitrija Angelkov, Natasa Koceska, and Saso Koceski. "Using mobile-phone accelerometer for gestural control of soccer robots." In Embedded Computing (MECO), 2012 Mediterranean Conference on, Bar, Montenegro, pp. 140- 143. 2012.

[15] Koceska, Natasa, and Saso Koceski. "Financial-Economic Time Series Modeling and Prediction Techniques–Review." Journal of Applied Economics and Business 2, no. 4 (2014): 28-33.

[16] Amazon SageMaker Service
https://aws.amazon.com/sagemaker/

[17] Guanlin L., Zhanhong M, Haiguang W., (2012), Image recognition of grape downy mildew and grape powdery mildew based on support vector machine, Proc international conference on computer and computing technologies in agriculture. Beijing, China

[18] Waghmare, H., Kokare, R., Dandawate, Y. (2016), Detection and classification of diseases of Grape plant using opposite colour Local Binary Pattern feature and machine learning for automated Decision Support System, (International Conference on Signal Processing and Integrated Networks), 513–518. doi: 10.1109/SPIN.2016.7566749

[19] Miaomiao Ji, Lei Zhang, Qiufeng Wu (2020), Automatic grape leaf diseases identification via UnitedModel based on multiple convolutional neural networks, Information Processing in Agriculture, Volume 7, Issue 3

[20] Khan A., Nawaz U., Ulhaq A. and Robinson R.W. (2020), Real-time plant health assessment via implementing cloud-based scalable transfer learning on AWS DeepLens. PLoS ONE 15(12): e0243243

[21] Liberty E., Karnin Z., Xiang B., Rouesnel L., Coskun B., Nallapati R., Delgado J., Sadoughi A., Astashonok Y., Das B P., Balioglu C., Chakravarty S., Jha M., Gautier P., Arpin D., Januschowski T., Flunkert V., Wang Y., Gasthaus J., Stella L., Rangapuram S., Salinas D., Schelter S., and Smola A. (2020), Elastic Machine Learning Algorithms in Amazon SageMaker. In Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data (SIGMOD '20). Association for Computing Machinery, New York, NY, USA, 731–737

[22] GitHub PlantVillage-Dataset
https://github.com/spMohanty/PlantVillage-Dataset (last accessed on: 27.10.2022)

[23] Du, Yixin, and Xin Li. "Recursive deep residual learning for single image dehazing." In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 730-737. 2018.

[24] Hossin, Mohammad, and Md Nasir Sulaiman. "A review on evaluation metrics for data classification evaluations." International journal of data mining & knowledge management process 5, no. 2 (2015).

Angela Tockova
Goce Delcev University of Stip,
Faculty of Computer Science
“Krste Misirkov” 10-A, Shtip, North Macedonia
angtockova@gmail.com

Zoran Zlatev
Goce Delcev University of Stip,
Faculty of Computer Science
“Krste Misirkov” 10-A, Shtip, North Macedonia
zoran.zlatev@ugd.edu.mk

Saso Koceski
Goce Delcev University of Stip,
Faculty of Computer Science
“Krste Misirkov” 10-A, Shtip, North Macedonia
saso.koceski@ugd.edu.mk