Early Prediction of Plant Diseases using CNN and GANs

Ahmed Ali Gomaa[1]
Management Information Systems
Madina Academy High Institute for Management and Technology, Giza, Egypt

Yasser M. Abd El-Latif[2]
College of Computing and Information Technology
Arab Academy for Science
Technology and Maritime Transport
Faculty of Science, Ain Shams University
Cairo, Egypt

***Abstract*—Plant diseases enormously affect the agricultural crop production and quality with huge economic losses to the farmers and the country. This in turn increases the market price of crops and food, which increase the purchase burden of customers. Therefore, early identification and diagnosis of plant diseases at every stage of plant life cycle is a very critical approach to protect and increase the crop yield. In this paper using a deep-learning model, we present a classification system based on real-time images for early identification of plant infection prior of onset of severe disease symptoms at different life stages of a tomato plant infected with Tomato Mosaic Virus (TMV). The proposed classification was applied on each stage of the plant separately to obtain the largest data set and manifestation of each disease stage. The plant stages named in relation to disease stage as healthy (uninfected), early infection, and diseased (late infection). Classification was designed using the Convolutional Neural Network (CNN) model and the accuracy rate was 97%. Using Generative Adversarial Networks (GANs) to increase the number of real-time images and then apply CNN on these new images and the accuracy rate was 98%.**

Keywords—Plants diseases; deep learning; early detection; convolutional neural network; generative adversarial networks

I. Introduction

Quality crop production is an essential feature of any country's economic growth. The agricultural sector provides jobs for many people; in addition, it accounts for a large part of the Gross Domestic Product (GDP) in many countries around the world [1]. For example, it is clear that there is a rapid and wide agricultural development and land reclamation in Egypt, with increased application of technological advances. There has been remarkable increase made in agricultural sector, such as the 1.5 million Acres Project, the El-Alamein project, and the lake and Fayoum projects. The use of modern methods and technologies in agriculture, significantly increase crop production and yield, and increase protection of plants from insect pest infestation and disease infections at all stages of planting, harvesting, and post-harvesting till successful marketing [2]. In a fast-growing world population, although there are many improvements in large production and access to food, food security is threatened by a different set of factors such as a decreased fertility of soil and lands, decreased plant pollination efficiency, insect and other arthropod pests, and plant diseases. Plant diseases are classified as follows: bacterial diseases (bacterial speck, bacterial spot, bacterial canker) [3], fungal diseases (early blight, late blight, sectorial leaf spot, and Anthracnose fruit rot) [4], and viral diseases Tomato Mosaic Virus (TMV) [5]. The early detection and accurate diagnosis of plant diseases at every stage of the plant life cycle and the extent of infection until reaching the most infectious stage or appearnce of severe disease symptoms, and easily classifying them is very important, as shown in Fig. 1.

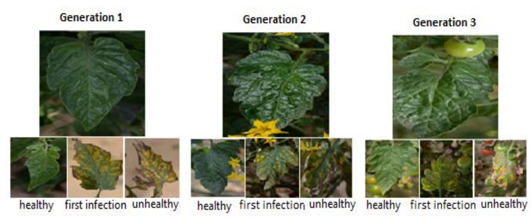

Fig. 1. The Age Stages of the Plant and the Stages of Infection.

Our Deep Learning (DL) model uses leaf images to detect diseases in plants by CNN to extract features from images, such as horizontal edges, vertical edges, Red Green Blue values, etc. A plant disease diagnosis system that uses machine learning techniques can correctly identify diseases plants healthy or unhealthy only [6]. Automatic detection of plant diseases at every age is an important research, analytical, and applied topic because it can help in monitoring large fields of crops in short time with high accuracy. Therefore, disease detection can discover symptoms visually and mechanically in the earliest time they appear on the leaves or other parts of the plant [6], as reported in numerous research publications and reports. CNN is believed to be the best DL neural network for extracting visual features [7]. The CNN-based network can be trained to discover diseases in plants by providing a large number of real-time images. In the case of lacking enough and good quality data or number of images, other techniques such as GANs can be used to generate the needed data for analysis and comparison with real-time data collected from the field. Both healthy and diseased plants and a future training model can be used to predict diseases in plants using plant leaf images [8].

II. Related Work

Recent advances in agricultural technology have led to a demand for a new set of automated, non-destructive methods for detecting plant diseases. Hence, several methods have turned to computer visual and machine learning (ML) techniques to create a rapid method for detecting plant diseases when symptoms appear [9]. Classifying plant diseases can be a very complex task because it depends mainly on published and used classification systems and also by experience of farmers and researchers.

Developing a reliable system that can be applied to many plant classes is a difficult task. To date, most automatic plant disease classification methods have depended on ML algorithms and basic feature engineering. These methods usually focus on specific environments and are suitable for a smaller number of categories, as some small changes in the system can lead to a severe drop in resolution. Recently, CNNs have shown impressive results in many image classification tasks that have allowed researchers to improve the classification of agriculture and plant diseases [10]. CNN is a technology that mixes artificial neural networks (ANNs) and up to date DL strategies [11].

In deep learning, CNN is at the center of spectacular advances. This ANN has been applied to several image recognition tasks for decades and has attracted the eye of the researchers of many countries in recent years; as CNN has shown promising performances in several computer visual and ML tasks [12]. This paper describes the underlying architecture and different applications of the CNN.

In Y. Kawasaki, ET. al. [13], the authors introduce a novel plant disease detection system based on CNN. Using only training images, CNN can automatically acquire the requisite features for classification and achieve high classification performance. A total of 800 cucumber leaf images are used to train CNN using the proposed techniques. Under the 4-fold cross-validation strategy, the proposed CNN-based system (which also extends the training dataset by generating additional images) achieves an average accuracy of 94.9 % in classifying cucumbers into two typical disease classes and a non-diseased class. In this study, the authors proposed a novel plant viral disease detection system using CNN and confirmed its effectiveness. They also asserted that the strategy for training CNN has significantly improved the accuracy of its classification. This work will free system users from paying extra attention to the details of plant shooting conditions.

In Y. Kawasaki, et. al. [13], future the system makes a large contribution in the agricultural field. Data augmentation is an essential part of the training process applied to DL models. The motivation is that a robust training process for DL models depends on large annotated datasets, which are expensive to be acquired, stored and processed. Therefore, a reasonable alternative is to be able to automatically generate new annotated training samples using a process known as data augmentation [14]. A GAN model consists of two important factors: the discriminator (D), and the generator (G). The generator and discriminator have opposite objectives during training. The discriminator is trained toward distinguishing between synthesized and real-time data while the generator is trained to fool the discriminator with synthesized data, as shown in Fig. 2.

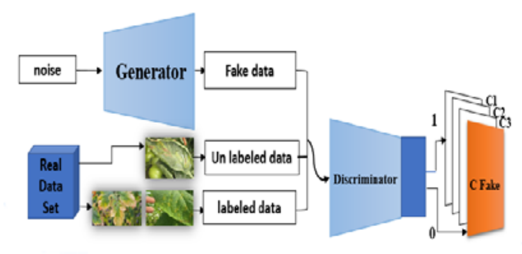

Fig. 2. GANs Architecture.

In D. Farm. [15], the authors propose a synthetic sampling solution is presented at the data level to identify them from small and unbalanced data sets using GANs. The reason for using GANs is the challenges in different fields as they deal with small data sets and volatile amounts of samples per category [16]. As a result, GANs offer an approach that can improve learning regarding data distributions, reduce bias resulting from class imbalance, and change classification. Resolution limits towards more accurate results. The method of [16] was trained on a small dataset of 2789 images of highly perishable tomato plant diseases with a class imbalance in 9 disease categories. Moreover, they evaluated their results in terms of different measures and compared the quality of these results for stratified excellence. GANs are an exciting and quickly changing field, delivering on the deal of generative models in their capacity to generate realistic examples across a range of problem domains. In 2014, conditional GANs was extended to a conditional model if both the generator and discriminator are conditioned on some extra data. They can perform the conditioning by data feeding into both the discriminator and generator as additional input layer [17].

In 2016, the Auxiliary Classifier GAN (AC-GAN) has received much interest due to easy and extensibility to different applications. AC-GAN integrates the conditional information (label) by training the GAN discriminator with an additional classification loss. AC-GAN is able to generate high-quality images and has been extended to different learning problems. However, the difference between the generated samples by AC-GAN going to decrease as the number of classes increases; hence limiting its power on large-scale data [18].

In 2016, the Information Maximizing GAN (Info-GAN) integrated the output of the generator to a component of its input called the hidden codes. Uncovering some successful and unsuccessful configurations for generating images using Info-GAN [19] are shown in Fig. 3.

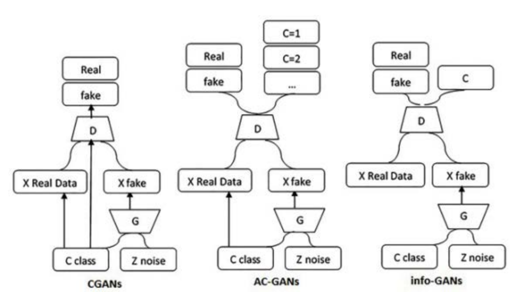

Fig. 3. Types of GANs.

III. Methodology

This methodology works on three steps:

First step: divide the plants into three class's generation (G1–G2–G3) with respect to the age stage.

Second step: each generation contains three Phases (Pi) of plants according to its pre-symptoms (uninfected P1 - early infection P2 – late infection P3).

Third step: Prediction and early detection of diseases to apply it to each generation, as shown in Fig. 4.

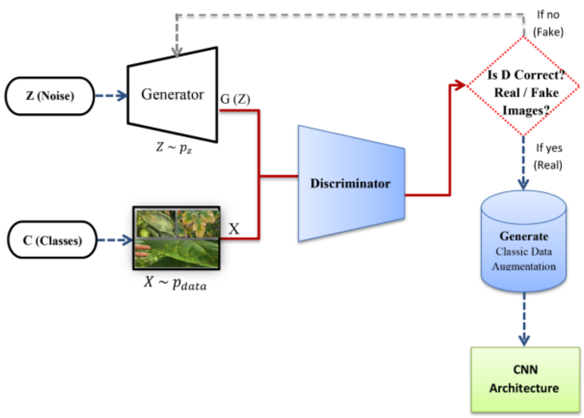

Fig. 4. The Model of Plants Generation/Phases.

Description of the model of plant generation/phases:

- p_{data} Represents all data set.
- Sample a noise set and a real-data set that includes classes (G1, G2, G3), each with size m.
- X represents the real sample belonging to the distribution $X \sim p_{data}$.
- Z denotes a random series belonging to the distributionp_z, which obeys a normal distribution. D and G represent the discriminator and generator respectively.
- The output of the generator is *X* fake (Data).
- The generator to make synthetic samples G (z) extremely approach to the distributionp_{data}.
- To increase the data set apply the discriminator on this data and extract the global polling and hidden layer to get extra real data.
- Global polling layer is inserted in front of the discriminator network's output layer to extract representative features with 512 dimensions.
- The discriminator input is (*X* and G (z)) and they are compared until get the output discriminator.
- The final output of the discriminator is real/fake images.
- When data is fake, the discriminator and generator are trained alternatively. For the training process of the generator, the synthetic samples G (z) is taken into the discriminator and the produced loss value loss G is transmitted back to the generator one more time.
- When the data is real, it is entered into the data augment repository and then after that it is transferred to the CNN structure.
- The final model output is the data augmentation as a shape class to using in CNN mode.

The following results were collected from the use of these images on the CNN, as shown in Table I.

IV. Experiments And Results

In this section, the study concludes the Experimental setup for our synthetic task generates data to detect diseases early:

1) Dataset: These images were collected from agricultural lands and it is a real data set that was used in this work to prove the growth stages of the plant and also increase the data from the original data and determine the stages of plant disease, there is a total of 5400 real images of diseased and healthy plants. These images covered all growth stages of plants and the extent of disease infection.

2) CNNs are proposed to reduce the number of parameters used and adapt the network architecture exactly to visual tasks. CNNs are usually composed of a set of layers that can be grouped by their functionalities; a CNN is typically composed of four types of layers: Convolution Layer, ReLu and sigmoid functions, Pooling, and Fully Connected Layer [20, 21, 22].

TABLE I. Apply CNN on the Groups of Images According to their Generation/Phases

Generation	Phases	Number of images	Loss	Vall_Loss	Accuracy	Vall_Accu	Max_Accu
G1	P1	600	0.0936	2.9398	0.9726	0.4200	0.9726
	P2	600	0.0725	2.8965	0.9701	0.3811	
	P3	600	0.0664	2.8356	0.9689	0.2147	
G2	P1	600	0.0861	2.7649	0.9601	0.2641	0.9754
	P2	600	0.0845	2.8527	0.9754	0.3979	
	P3	600	0.0689	2.8950	0.9742	0.4215	
G3	P1	600	0.0621	2.6691	0.9597	0.2954	0.9748
	P2	600	0.0859	2.9184	0.9726	0.4021	
	P3	600	0.0746	2.8302	0.9748	0.4593	

Table I shows that the highest percentage in healthy cases is in the first age early stage (uninfected) of plant growth. The highest percentage in cases of the first virus infection (early infection) is in the second age stage. The highest percentage in diseased cases (late infection) is in the third age stage.

The plant in the first age stage is in the sterilization stage and healthy hybridization and not exposed to a large pesticides spraying, taking into account the appropriate weather for cultivation. After that, in later stages of growth, the plant is exposed to larger spraying with pesticides and exposed to different climate factors as well as poor workmanship (farming), its stage will be the highest complete unhealthy rate. The data was divided into 70% training and 30% testing, and determining the number of batches required for the model. The accuracy rate and the loss rate were deduced as shown in the Fig. 5.

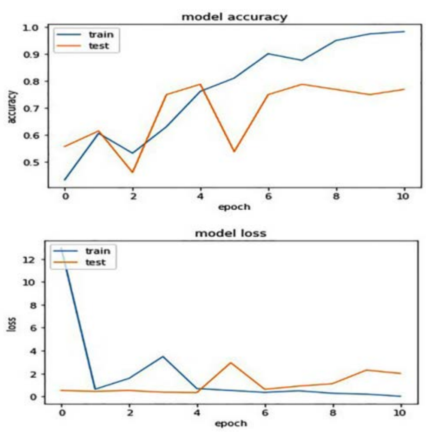

Fig. 5. Model (Accuracy and Loss).

The number of the expected data on the actual data was clear in the following table, and it was found that the second growth stage is the most vulnerable stage to viral infection through the distribution of data by 70% training and 30% testing, as shown in Table II.

['Gen2Phase1', 'Gen2Phase3', 'Gen3Phase3', 'Gen3Phase2', 'Gen1Phase2', 'Gen2Phase2', 'Gen1Phase3', 'Gen3Phase1', 'Gen1Phase1'].

Computed fusion matrix: Heterogeneous data sources can be collectively mined by data fusion. Fusion can focus on a specific target relation and exploit directly associated data together with data on the context or additional constraints [23], as shown in the Fig. 6.

TABLE II. TEST RESULT BY CNN WITH TRAINING AND TESTING

Prediction	G1	G2	G3	All
Actual				
P1	240	210	90	540
P2	120	300	120	540
P3	70	230	240	540
All	430	740	450	1620

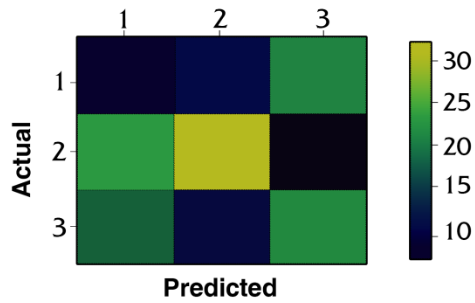

Fig. 6. Computed Fusion Matrix.

3) Generative Adversarial Networks (GANs): After the CNN stage of classification and prediction was completed, the stage of increasing data into by using the (GANs), and during this stage, the experiments were done on the real data and the steps of this stage were as follows:

- Prepare Dataset:
  - o import numpy as np
  - o import pandas as pd
  - o import os
  - o print(os.listdir("TomatoDB"))
- Generator and Discriminator for Dataset:
- Image Samples:
  - o SAMPLES_TO_SHOW = 8
  - o Input: (60, 64, 64, 3), as shown in the Fig. 7.

Fig. 7. Images Sample Complete Description.

- Code implementation steps, as shown in Table III:

TABLE III. RESULT ACCURACY FOR GENERATOR AND DISCRIMINATOR MODEL: "FUNCTIONAL_1"

Layer (type)	Output Shape	Parameter
input_1 (InputLayer)	[(None, 64, 64, 3)]	0
conv1 (Conv2D)	(None, 32, 32, 32)	2432
batch_norm1 (BatchNormalization)	(None, 32, 32, 32)	128
conv1_out (LeakyReLU)	(None, 32, 32, 32)	0
conv2 (Conv2D)	(None, 16, 16, 64)	51264
batch_norm2 (BatchNormalization)	(None, 16, 16, 64)	256
conv2_out (LeakyReLU)	(None, 16, 16, 64)	0
conv3 (Conv2D)	(None, 8, 8, 128)	204928
batch_norm3 (BatchNormalization)	(None, 8, 8, 128)	512
conv3_out (LeakyReLU)	(None, 8, 8, 128)	0

conv4 (Conv2D)	(None, 8, 8, 256)	819456
batch_norm4 (BatchNormalization)	(None, 8, 8, 256)	1024
conv4_out (LeakyReLU)	(None, 8, 8, 256)	0
conv5 (Conv2D)	(None, 4, 4, 512)	3277312
batch_norm5 (BatchNormalization)	(None, 4, 4, 512)	2048
conv5_out (LeakyReLU)	(None, 4, 4, 512)	0
flatten (Flatten)	(None, 8192)	0
ligit (Dense)	(None, 1)	8193
Total params	Trainable params	Non-trainable params
4,367,553	4,365,569	1,984

MODEL: "FUNCTIONAL_3"

Layer (type)	Output Shape	Parameter
input_2 (InputLayer)	[(None, 100)]	0
Dense (Dense)	(None, 8192)	827392
leaky_re_lu (LeakyReLU)	(None, 8192)	0
reshape (Reshape)	(None, 4, 4, 512)	0
trans_conv1 (Conv2DTranspose)	(None, 8, 8, 512)	6554112
batch_trans_conv1 (BatchNormalization)	(None, 8, 8, 512)	2048
trans_conv1_out (LeakyReLU)	(None, 8, 8, 512)	0
trans_conv2 (Conv2DTranspose	(None, 16, 16, 256)	3277056
batch_trans_conv2 (BatchNormalization)	(None, 16, 16, 256)	1024
trans_conv2_out (LeakyReLU)	(None, 16, 16, 256)	0
trans_conv3 (Conv2DTranspose)	(None, 32, 32, 128)	819328
batch_trans_conv3 (BatchNormalization)	(None, 32, 32, 128)	512
trans_conv3_out (LeakyReLU)	(None, 32, 32, 128)	0
trans_conv4 (Conv2DTranspose)	(None, 64, 64, 64)	204864
batch_trans_conv4 (BatchNormalization)	(None, 64, 64, 64)	256
trans_conv4_out (LeakyReLU)	(None, 64, 64, 64)	0
logits (Conv2DTranspose)	(None, 64, 64, 3)	4803
out (Activation)	(None, 64, 64, 3)	0
Total params	Trainable params	Non-trainable params
11,691,395	11,689,475	1,920

- Result plot loss curve (Fig. 8):

The loss result it's almost 2%, and it's the complement the accuracy rate, and also the result rate of generator little some extent and the result rate of discriminator high some extent, this means that the capacity of the model is high, this is what is aimed to be achieved:

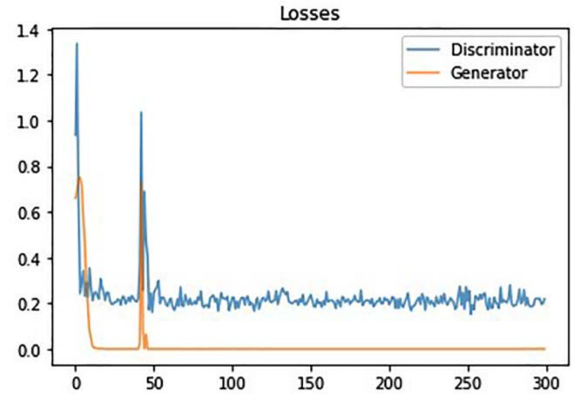

Fig. 8. Losses Result.

- Result Accuracy and Losses (Table IV):

TABLE IV. GANS RESULT

	Losses	Accuracy
GANs:	2.1%	97.9%

V. CONCLUSION AND FUTURE WORK

To summarize, DL was used in early prediction to detect diseases in different plant growth stages using the CNN algorithm for classification and prediction. Here, using the tomato infected with TMV as a model, the accuracy rate of TMV infection was 97%. The GANs used to increase the size of data and prediction accuracy rate by 98% when compared to the original data. For each plant growth phase, it became clear that the most growth stage group is vulnerable to viral infection is the second group. Therefore that determining the growth stages in this paper helped at obtaining results that prove the age group most susceptible to Unhealthy by determining the stages of Unhealthy also (healthy - first infection - Unhealthy), Thus, the study has concluded the previous results by applying to a set of real data that was collected manually from one of the farms in Egypt. Future work will include several DL models for early detection and classification of plant diseases due to using the rapid progress and improvements in DL models, transfer learning techniques, and CNN frameworks. Larger real-time dataset of TMV-infected tomato plants, and other important plant-disease system will be tested for attaining highest prediction accuracy. . Building a robust and accurate digital and computer-based plant pest-infestations and microbial disease-infections early-detection and warning system, will significantly help plant protection in early stages, with increased yield, quality, local marketing, and international exporting competitiveness.

REFERENCES

[1] M. Saleem, J. Potgieter, “Plant Disease Detection and Classification by Deep Learning” MDPI journals, 2019 Oct 31.

[2] P. Sobiczewski, “Bacterial Diseases of Plant: Epidemiology, Diagnostics, and Control” ISSN journals, 1392-3196, No. 3 2008.

[3] D. Singh, S. Teotia, “Fungal Disease Management in Plants” DOI: 10.1007/978-81-322-1620-9_19, October 2014.

[4] I. Ferriol, L. Galipienso, L. Rubio, “Detection of Plant Viruses and Disease Management: Relevance of Genetic Diversity and Evolution” Frontiers in plant science, 17 July 2020.

[5] V. Sharma, A. Verma, “Classification Techniques for Plant Disease Detection” International Journal of Recent Technology and Engineering (IJRTE), March 2020.

[6] S. Kaur, S. Pandey, S.Goel, “Plants Disease Identification and Classification through Leaf Images: A Survey” Journal research gate, 2018.

[7] L. Kawaguchi, B. Saitama, S. Chikusa, N. Nagoya, “How Convolutional Neural Networks Diagnose Plant Disease” 2019.

[8] T. Salimans, I. Goodfellow, A. Radford, “Improve technique for training GANs” SAIL journal, 2016.

[9] J. Amara, B. Bouaziz, A. Algergawy, “A Deep Learning-based Approach for Banana Leaf Diseases Classification” Lecture Notes in Informatics (LNI), 2017.

[10] M. Brahimi, M. Arsenovic, S. Laraba, K. Boukhalfa, “Deep Learning for Plant Diseases: Detection and Saliency Map Visualisation” Part of the Human-Computer Interaction Series book series (HCIS), springer, June 2018.

[11] R. Yamashita, M. Nishio, R. Gian Do, K. Togashi, “Convolutional neural networks: an overview and application in radiology” Springer, 2018.

[12] S. Sakib, N. Ahmed, A. Kabir, H. Ahmed, “An Overview of Convolutional Neural Network: Its Architecture and Applications” Journal research gate, November 2018.

[13] Y. Kawasaki, H. Uga, H. Iyatomi, S. Kagiwada, “Basic Study of Automated Diagnosis of Viral Plant Diseases Using Convolutional Neural Networks” Springer international journal, 18 December 2015.

[14] T. Tran, T. Pham, G. Carneiro, L. Palmer, I. Reid, “A Bayesian Data Augmentation Approach for Learning Deep Models” Research gate, October 2017.

[15] D. Farm, “Image-to-Image Translation with GAN for Synthetic Data Augmentation in Plant Disease Datasets” Journal research gate, July 2019.

[16] A. Odena, C. Olah, J. Shlens, “Conditional Image Synthesis with Auxiliary Classifier GANs” arXiv: 1610.09585v4, 20 Jul 2017.

[17] S. Osindero, M. Montreal, “Conditional Generative Adversarial Nets” ArXiv: 1411.1784v1 cs.LG, 6 Nov 2014.

[18] M. Gong, Y. Xu, C. Li, K. Zhang, K. Batmanghelich, “Twin Auxiliary Classifiers GAN” NIPS journal, 2019.

[19] K. Evtimova, A. Drozdov, “Understanding Mutual Information and its Use in InfoGAN” Semantic scholar, 2016.

[20] A. Khan, A. Sohail, U. Zahoora, A. Saeed, “A Survey of the Recent Architectures of Deep Convolutional Neural Networks” Springer, Published: 21 April 2020.

[21] M. Pashaei, H. Kamangir, M. Starek, P. Tissot, “Review and Evaluation of Deep Learning Architectures for Efficient Land Cover Mapping with UAS Hyper-Spatial Imagery: A Case Study over a Wetland” Multidisciplinary Digital Publishing Institute (MDPI) journal, Published: 16 March 2020.

[22] K. Shridhar, J. Lee, H. Hayashi, P. Mehta, B. Iwana, S. Kang, S. Uchida, S. Ahmed, A. Dengel, “ProbAct: A Probabilistic Activation Function for Deep Neural Networks” arxiv journal, 16 Jun 2020.

[23] M. Zitnik, B. Zupan, “Data Fusion by Matrix Factorization” IEEE Transactions on Pattern Analysis and Machine Intelligence, July 2013.