DEÜ FMD 24(71), 631-642, 2022

Dokuz Eylül Üniversitesi Mühendislik Fakültesi Fen ve Mühendislik Dergisi
Dokuz Eylul University Faculty of Engineering Journal of Science and Engineering

Basılı/Printed ISSN: 1302-9304. Elektronik/Online ISSN: 2547-958X

Classification of Scatter Plot Images Using Deep Learning

Derin Öğrenme Kullanarak Dağılım Grafiği Görüntülerinin Sınıflandırılması

Derya Birant [1*], Aydanur Akca [2], Buse Bozkurt [3], Mehtap Baglan [4]

[1,2,3,4] Dokuz Eylul University Engineering Faculty Department of Computer Engineering, Izmir, Turkey
Sorumlu Yazar / Corresponding Author *: derya.birant@deu.edu.tr

Geliş Tarihi / Received: 08.08.2021 *Araştırma Makalesi/Research Article*
Kabul Tarihi / Accepted: 16.01.2022 *DOI:10.21205/deufmd.2022247126*
Atıf şekli/ How to cite: BIRANT,D., AKCA, A., BOZKURT, B., BAGLAN, M. (2022). Classification of Scatter Plot Images Using Deep Learning. DEUFMD, 24(71), 631-642.

Abstract

Scatter plot is one of the well-known charts and is frequently embedded in different types of documents such as articles, books, and dissertations. However, the information given in the scatter plots can't be directly noticed by visually impaired individuals, because they are usually in an image format, and so they are not naturally readable by machines. To solve this problem, this paper proposes a system that can extract visual properties from scatter plot images using deep learning and image processing techniques. It is the first study that automatically classifies scatter plots in terms of two aspects: *degree of correlation* (strong or weak) and *types of correlation* (positive, negative, or neutral). In the experimental studies, alternative convolutional neural network (CNN) architectures were compared on both synthetic and real-world datasets in terms of accuracy, including Residual Networks (ResNet), Alex Networks (AlexNet), and Visual Geometry Group (VGG) Networks. The experimental results showed that the proposed system successfully (93.90%) classified scatter plot images to help visually impaired users understand the information given in the graph.

Keywords: *Scatter plots, Image classification, Deep learning, Machine learning*

Öz

Dağılım grafiği, iyi bilinen grafiklerden biridir ve makaleler, kitaplar, raporlar gibi birçok farklı türdeki dokümanlarda sıklıkla kullanılmaktadır. Ancak, dağılım grafikleri genellikle görüntü biçiminde olduğu için grafiklerde verilen bilgiler görme engelli kişiler tarafından fark edilemez, yani esasen makine tarafından okunabilir değillerdir. Bu sorunu çözmek için, bu makale, derin öğrenme ve görüntü işleme tekniklerini kullanarak, dağılım grafiği görüntülerinden görsel özellikleri çıkartabilen bir sistem önermektedir. Dağılım grafiklerini iki açıdan otomatik olarak sınıflandıran ilk çalışmadır: *korelasyon derecesi* (güçlü veya zayıf) ve *korelasyon türleri* (pozitif, negatif veya nötr). Deneysel çalışmalarda, Artık Ağlar (ResNet), Alex Ağları (AlexNet) ve Görsel Geometri Grubu (VGG) Ağları gibi alternatif evrişimsel sinir ağı (CNN) mimarileri hem sentetik hem de gerçek dünya veri setlerinde doğruluk açısından karşılaştırılmıştır. Deneysel sonuçlar, önerilen sistemin başarılı bir şekilde (%93,90) dağılım grafiği görüntülerini sınıflandırarak görme engelli kullanıcıların grafikte verilen bilgileri anlamalarına yardımcı olduğunu göstermiştir.

Anahtar Kelimeler: *Dağılım grafikleri, Görüntü sınıflandırma, Derin öğrenme, Makine öğrenmesi*

1. Introduction

Charts are frequently embedded elements in different types of documents such as articles, books, reports, newspapers, magazines, presentation slides, and blog posts. Charts are preferable and popular tools to present numerical data in digital sources. Chart representations have many benefits over text-based representations such as quickly illustrating comparisons, representing ideas better, and long retention time in human memory.

A *scatter plot* is one of the well-known charts that is used to represent the correlation between two variables. Although a scatter diagram cannot describe the cause of such a correlation, it can show whether or not a correlation exists, and if so, just how strong it is. If the values corresponding to two variables lie along a line in the diagram, it means that the correlation between them gets stronger. Scatter plots can be interpreted in five different ways: strong positive correlations, weak positive correlations, non-correlation, strong negative correlations, and weak negative correlations.

Scatter plots are generally generated to be understandable by humans and they aren't essentially readable by machines because they are usually in an image form. Nevertheless, recently, there is an increasing need to automatically classify scatter plot images to use this information for further processes such as for providing recommendations in applications, developing better search engines, investigating regression models in scatter plot data [1], tagging the scatter plot images for a machine learning system to analyze later, redesigning scatter plots, placing new objects on a scatter plot for augmented reality, and enabling decision trend analysis for a dataset (i.e., wine, wages, and cars data) [2]. Especially, scatter plot classification is needed to give information about the relationship between two variables [3] such as evaluating the correlation between temperature and vegetation (T-V) for soil moisture estimation in the field of remote sensing [4]. Another example is the grouping of scatter plots according to their types of physical or chemical interactions between microorganisms and humans in the field of biology [5].

The novelty and main contributions of this work can be summarized as follows. It proposes a machine learning-based model that can extract graphical properties from scatter plot graphic images. It is the first study that can automatically predict the class label of a scatter plot in terms of two aspects: *degree of correlation* (strong or weak) and *types of correlation* (positive, negative, or neutral). The study is also original in that it compares alternative deep learning architectures in classifying scatter diagrams in terms of accuracy.

Scatter plots aim to make a visual evaluation of the correlation between two scaled variables and are widely used in many fields. However, because they are visual pictures, they can't be noticed by visually impaired people. Therefore, visually impaired people have difficulty in accessing the information that they can be inferred from scatter plots. Visually impaired people can use document and screen readers for text-based data sources; however, these readers cannot read chart images, they can only read the headings given under charts. We developed a system that classifies scatter plot images using deep learning. Thereby, our solution can help visually impaired people understand the information given in scatter plots.

The remainder of the paper is generally structured as follows. Section 2 explains the previous studies about chart classification. Section 3 describes the proposed approach and explains the details of the CNN models. Section 4 describes the dataset and presents the performances of these models. Finally, the conclusion and possible future works are given in Section 5.

2. Related Works

Though text-based information is the main data source, there is an increasing tendency to present the graphic forms by rendering the information visual. Charts are commonly used to present and examine the data, show the relationship between variables, and highlight the main points available in the data. They are not naturally readable by machines because they are usually in picture form. Therefore, automated methods such as computer-based analysis are required to extract information from charts.

While some studies [6, 7] in the literature focused on extracting data from charts, some others [8, 9] performed the classification of charts. While the former ones [10, 11] use image processing and/or optical character recognition

(OCR) techniques to extract information from chart images, the latter ones [12, 13] use image classification methods like deep learning to make a prediction based on patterns observed from charts.

Previous studies on chart classification usually built a model for determining the types of charts (i.e., line chart, pie chart, and bar chart) [8, 9, 12, 14]. On the other hand, some studies especially focused on a single type of chart and classify it according to its properties such as the classification of bar charts [15], the categorization of line charts [16], visual reasoning over pie charts [17], pattern recognition from control chart [13, 18, 19], object recognition over candlestick charts [20], and making predictions on price chart images [21].

In the literature, many machine learning studies on chart images have been focused on classification problem [22-24]; however, recently, some studies have focused on a clustering problem [25]. Researchers have used different machine learning techniques to classify chart images by type such as Convolutional Neural Networks (CNN) [12, 13, 18], Long Short-Term Memory (LSTM) [26], Adaptive Logistic Regression (LogitBoost) [27, 28], Random Forest [23], Support Vector Machines [22, 29, 30], Logistic Model Tree (LMT) [28], Decision Tree [24, 31], Reduced Error Pruning Tree (REPTree) [28], Naive Bayes [31], Bayes Network (BayesNet) [27, 28], K-Nearest Neighbors (KNN) [31], Linear Logistic Regression (SimpleLogistic) [28], Artificial Neural Networks (ANN) [32, 33], Multivariate Adaptive Regression Splines (MARS) [34], Extreme Learning Machine (ELM) [35], and Nearest Neighbor With Generalization (NNge) [27].

In the rest of this section, recently developed systems [6, 26, 30, 35-39] related to charts are explained briefly.

ChartFuse [30]: It is a method to classify chart images, which uses a feature extractor, called heterogeneity index (HI). The study considers the common and uncommon microstructural features in the images and interprets colors, texture, structural and illumination properties of the charts. The method consists of four steps: first, detection of the chroma effects; second, extraction of features related to these effects; third, combining all the separate micro-structures. and finally, fusing with local feature descriptor-based approaches.

Chart Decoder [35]: It is a system that takes a chart picture as input and then obtains the numeric and textual information from it as output by using computer vision, deep learning, and text recognition methods. It consists of four steps: chart classification (i.e., bar, pie, line, scatter, and radar), graphical component extraction, textual component extraction, and chart data recovery. GoogLeNet was chosen by considering the classification accuracy and the number of parameters related to the method.

Deep Chart [36]: It is a framework that was developed to classify charts by types, including bar, line, scatter, pie, and flow chart. This framework combines deep belief networks and deep convolutional networks (ConvNets) to achieve good performance for chart classification. It was demonstrated that more discriminative features could be obtained by deep ConvNets compared to hand-crafted feature extraction for chart classification.

ChartNet [26]: It is a question answering system developed for statistical charts such as pie and bar charts. It solves the problem of reasoning over chart images using CNN and LSTM models.

ChartSight [37]: It is an interactive and automated chart understanding system that extracts the chart images in the documents and then classifies them into different chart categories such as bar, pie, line, area, map, and radar charts. It involves a densely connected CNN architecture.

Chartem [6]: It is a data-embedding method to automatically encode a piece of knowledge into the background of a graph image. It also extracts information from a chart image to enable repurpose and reusability of chart images.

ReVision [38]: It is a system that consists of a three-stage pipeline: chart classification, data extraction, and redesign to improve graphical perception. It utilizes an OCR engine to extract text regions, and also uses machine learning and computer vision methods to identify ten different chart types such as pie, bar, radar, area, curve, and maps. It can be used to improve information retrieval systems with graphic metadata and to provide screen-reading of charts for visually impaired users.

ChartSense [39]: It is an automated chart data extraction tool. First, ChartSense determines the type of the chart given an image format using a deep learning technique and after that, it extracts the underlying information from the chart image by using interactive (semi-automatic) subtraction algorithms, which are optimized for each chart type. GoogLeNet was chosen from among alternative CNN variations such as AlexNet and LeNet-1, due to its higher accuracy.

Unlike the studies aforementioned here, our study especially focused on scatter plots. It is the first study that automatically classifies scatter plots in terms of two aspects: *degree of correlation* (strong or weak) and *types of correlation* (positive, negative, or neutral). Our study is also original in that it compares alternative deep learning architectures in classifying scatter plots in terms of accuracy.

3. Material and Methods

3.1. Proposed Approach

The scatter plot is used to understand and interpret the correlation between two variables. It is a preferred data visualization tool to show the data as a whole model and therefore to better understand the analyzed data. Most of the scatter plots currently presented in many fields have been created to be understandable by people. However, visually impaired individuals cannot classify and interpret these graphics when considered. In this study, we aim to offer a solution for visually impaired individuals who cannot make sense of scatter plots.

This study aims to build a machine learning model that describes the information given in the scatter diagrams. While building this model, it is aimed to classify the plots according to their correlation types (positive, negative) and correlation degrees (weak, none, strong). Our solution involves automatically classifying scatter plots by using deep learning. We developed a system that classifies scatter plot images for visually impaired individuals.

Figure 1 presents a general overview of the proposed approach. The scatter plot images are fed into the convolutional neural network. In the training phase, CNN extracts useful features from images and runs the learning algorithm that maps inputs to correct outputs. Once the predictive model is built, then it is used to classify an unseen scatter plot image.

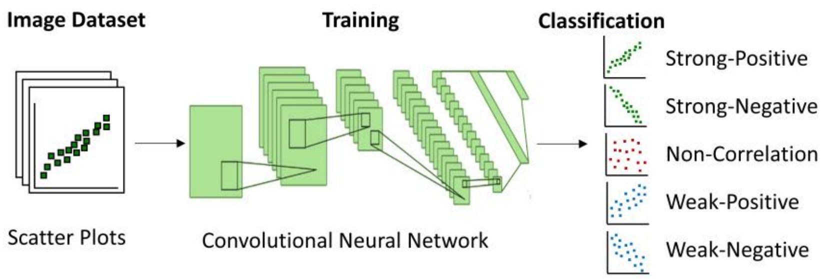


Figure 1. An overview of the proposed approach

Figure 2 shows a screenshot from the web application. We designed a user-friendly and easy-to-use interface since it is dedicated to visually impaired users. After a scatter plot image is selected, it is automatically classified by the deep learning model, and the classification result is displayed on the screen in text format. After that, the application converts text into sound using an audio tool. In this way, a visually impaired user can listen and learn the type of scatter plot. In the development of the system, an API was used to access the model created in the backend and was stored on a server. Requests sent by the end-user to the system via the frontend are met by the API and forwarded to the backend. The response from the backend comes to the frontend and meets with the end-user via API.

In this study, we used Streamlit which is an open source Python framework for building web applications for Machine Learning and Data Science. Using Streamlit, it is possible to instantly develop web applications with Python and deploy them easily. It can be seamlessly integrated with other Python libraries such as NumPy, Pandas, Matplotlib, and much more. In the development of the system, the ResNet50 model was registered since it achieved the best performance among alternative models. The saved model is accessed via Streamlit and predictions are made on the inputs from the user. Streamlit makes it seamless to work on the interactive coding cycle and displays the results in the web application, so there is no distinction between frontend and backend while creating the web page.

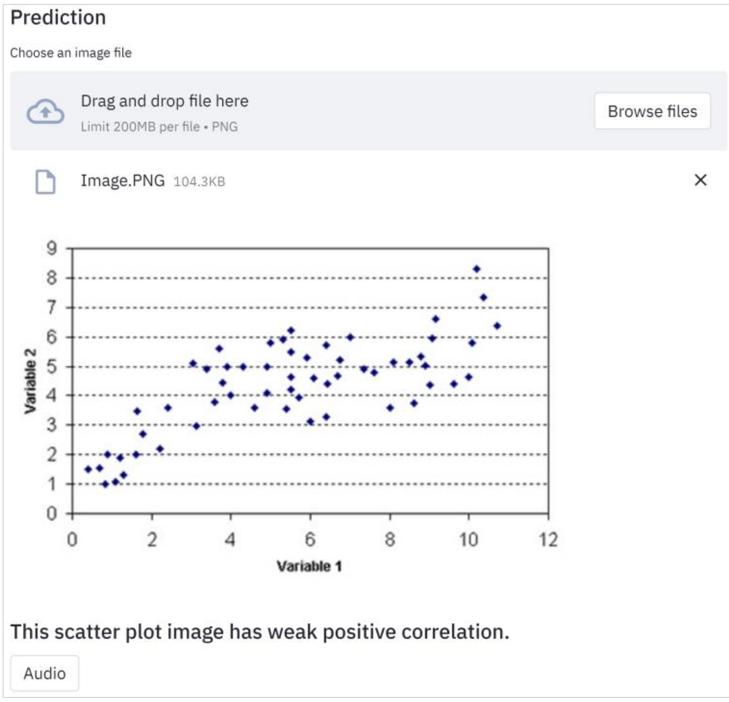


Figure 2. A screenshot from the web application

3.2. Deep Learning Architectures

Convolutional Neural Network (CNN) is one of the most popular deep learning models and has been widely-used in computer vision tasks, especially in image classification. A CNN consists of a series of layers such as convolutional layers, pooling layer, dropout layer, fully-connected layer, which can be arranged one after another to form a specific network architecture. It automatically extracts useful knowledge from the dataset without involving manual preprocessing or feature engineering procedures.

In this study, four different CNN models were tested for scatter plot image classification, including Alex Networks (AlexNet), OpenCV-CNN, Visual Geometry Group (VGG) Networks, and Residual Networks (ResNet). Each deep learning model has a specific architecture, learning procedure, and properties. The rest of this section describes each of them briefly.

Alex Networks (AlexNet): AlexNet [40] is one of CNN's ready-made architectures that consists of eight layers such that five convolutional layers are followed by three fully-connected layers. Some convolutional layers are followed by *Maximum Pooling* layers. The AlexNet model is defined using the sequential function in the Keras library. All layers except the output layer are formed as three fully-connected layers using Rectified Linear Unit (*ReLU*) as the activation function. In the final (output) layer, the *Softmax* activation function is used to make a multi-class classification. In addition, the *Adam* optimizer was successfully used to minimize the loss. In this study, the parameters of the network were kept as specified in the original model, so our model consists of five convolution layers with 11×11, 5×5, 3×3, 3×3, and 3×3 filters, respectively.

OpenCV-CNN: Two different models were built, the first one with grayscale images and the other one with RGB images. Both of them include six convolution layers with a kernel size of 3 x 3. It was constructed as three fully-connected layers using *ReLU* as an activation function for the hidden layer. In the final layer, the *Softmax* activation function was used to make the multi-class classification. *Max Pooling* layers were added to reduce size in convolutional layers. *Dropout* was also added to prevent overfitting. Besides, the *Adam* optimizer was used to minimize the loss.

Visual Geometry Group (VGG): VGG19 [41] is a CNN model and has been trained on 224 x 224 RGB images. In this architecture, the average RGB values from each pixel are extracted as a preprocessing process. All convolution layers have a core size of 3×3 and the network has a depth of 19 layers. Of these layers, 13 are convolutional layers and 3 are fully-connected layers. Volume size reduction is done with *Maximum Pooling*. *ReLU* was used as an activation function in VGG19. Despite its simple nature, this model achieves competitive classification accuracy compared to more complex networks, i.e., *GoogLeNet*. However, it has two disadvantages. First, the training process is slow. Another disadvantage is that the network architecture weights are quite large.

Residual Networks (ResNet): ResNet [42] is a deep learning model that makes it possible to train up to hundreds or even thousands of layers and it has achieved very high accuracy, especially in many computer vision applications. It is considered as one of the breakthroughs in deep learning in recent years. ResNet is created by adding residual values to residual blocks in the next layers. This feature distinguishes ResNet from other models. As the mesh depth increases, the accuracy of the model saturates but then tends to decline rapidly. To solve this problem, ResNet has some shortcuts which are added between layers. With these shortcuts, the corruption that occurs as the network gets deeper can be prevented. Recently, some improvements have been made related to the ResNet architecture and some different versions are currently available. Resnet50, which can be used with transfer learning, is a deep network of 50 layers trained on an image dataset. The layers consist of 1×1, 3×3, and 7×7 kernel sizes. *Max Pooling* and *Average Pooling* processes are applied to reduce the size. In the ResNet50 model, *ReLU* is used as the activation function for the hidden layers of the network, and *Softmax* is utilized as the activation function for the final layer.

In this study, transfer learning was performed in model training with the ResNet50 and VGG19 architectures. *Transfer learning* is a kind of machine learning method that harnesses the power of a pre-trained model which solves another problem. Thanks to this method, rapid progress is achieved when modeling the second task and therefore the performance is improved compared to the initial performance of a traditional learner. Transfer learning is a popular approach in deep learning and it usually gives successful results on small datasets.

4. Experimental Studies

The aim of this experiment is to build a deep learning model that will help visually impaired people understand the information given in a chart. We especially focused on scatter plots among the charts. Alternative convolutional neural network (CNN) architectures were compared to determine the best model in terms of accuracy, including AlexNet, OpenCV-CNN, VGG19, and ResNet50. This study was also aimed to reduce the training times of CNNs by exploiting the parallelism in the algorithms using Graphics Processing Units (GPU).

The Python programming language was chosen to train and build deep learning models. The following libraries, packages, and tools were used in this study during the data manipulation and training phases: Numpy, Pandas, Matplotlib, OpenCV, Scikit-learn, Keras, and Tensorflow. In addition, the Streamlit and Pyttsx3 libraries were used during the web development phase.

The performances of the models were evaluated in terms of accuracy, recall, precision, f-measure, and confusion matrix. The values of all these metrics should be as high as possible. *Accuracy* is calculated as the ratio of the values, which are predicted correctly by the model, to the total size of the set. It refers to how well the model estimates previously unseen input data. The test accuracy is measured on the test set, while the training accuracy is the classification accuracy of the current predictive model on the training set. Test loss and train loss are the recognition loss rates on the testing and training sets, respectively. Model accuracy alone is not sufficient, especially in unbalanced datasets that are not evenly distributed. For this reason, we also evaluated other metrics. The *Precision* measure indicates how many of the values estimated as positive are actually positive. It is very important especially when the cost of false-positive prediction is high. On the other hand, *Recall* is a measure that states how much of the instances that are needed to estimate as positive, we estimated as positive. Recall is also a significant assessment measure that helps us in situations where the cost of estimating as false-negative is high. *F-measure* is the harmonic mean of the recall and precision values. The reason why it is a harmonic mean, instead of a simple mean, is that there is generally a trade-off between recall and precision. The details of all these measures are given in Equations (1) through (4).

$$Accuracy = \frac{true\ positives + true\ negatives}{total\ data} \quad (1)$$

$$Precision = \frac{true\ positives}{true\ positives + false\ positives} \quad (2)$$

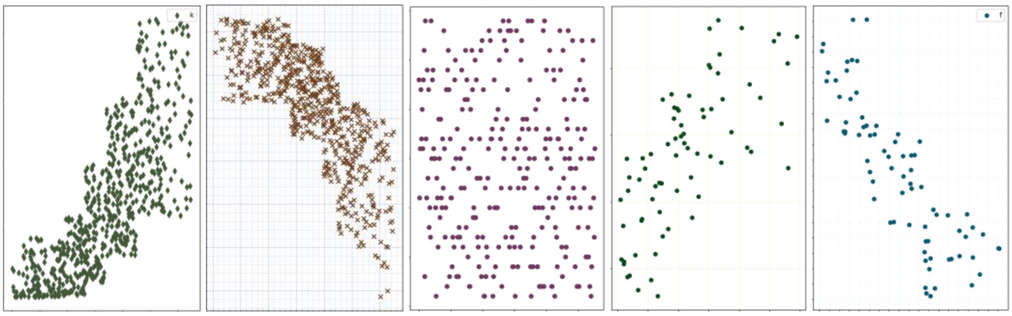
$$Recall = \frac{true\ positives}{true\ positives + false\ negatives} \quad (3)$$

$$Fmeasure = \frac{2 \times precision \times recall}{precision + recall} \quad (4)$$

4.1. Dataset Description

Scatter plots are used to determine the relation between two different variables. Although the reason for the relationship is not understood with this diagram, it can be understood whether there is a relationship or how strong the relationship is. In this study, scatter plots are divided into five general classes by considering the correlation types (positive, negative) and correlation degrees (weak, none, strong). These are strong-positive, weak-positive, strong-negative, weak-negative, and non-correlation. Figure 3 shows a sample scatter plot image from each class.

- *Strong Positive Correlation:* As the value of x increases, the value of y also increases, and the points are clustered close to each other.
- *Weak Positive Correlation:* The values of x and y increase at the same time, but the points are scattered.
- *Strong Negative Correlation:* As the value of x increases, the value of y decreases, and the points are clustered close to each other.



(a) strong-positive (b) strong-negative (c) non-correlation (d) weak-positive (e) weak-negative

Figure 3. Types of scatter plots

- *Weak Negative Correlation:* As the value of x increases, the value of y decreases, but the points are scattered.
- *Non-Correlation:* There is no relationship between two variables (x and y).

Since collecting a large number of chart images is a difficult task, both real-world and synthetic data were used to assess the efficiency of the different CNN models. The data was shuffled randomly and then split into training, validating, and test sets. First, the model was trained with 3000 scatter plot images, 600 from each class. After that, the model was validated with 1000 images and finally tested with 1000 images, 200 from each class. In total, the dataset contains 5000 different scatter plot images for training, validation, and testing.

The real-world data was collected through a search engine by searching the related keywords such as "scatter plot", "scatter diagram", and "scatter graph", and "scatter chart". We eliminated the duplicate charts obtained from different search keywords by comparing each other. The collected chart images were then manually labeled by three different persons to serve as ground truth. Collected real-world scatter plot images have different properties, features, and design styles such as different color schemes, coordinate values, scales, and orientations, with/without legends and text regions. Furthermore, in the images, dependent and independent variables are correlated by different functional relationships such as linear, logarithmic, and exponential.

The synthetic data was generated in Python. We generated an arbitrarily large dataset in which the charts visually vary. In the synthetic scatter plot data generation process, the following workflow was defined and applied. First, we investigated the variation in real-world scatter plot visualizations to synthesize data as realistically as possible. After that, different equations were defined for each correlation type (strong correlation, weak correlation, and non-correlation). Linearly, logarithmically, and exponentially increasing and decreasing plots were created to provide diversity. Some random points were generated for the values on the x-axis, and then the values on the y-axis were created using the values on the x-axis according to the correlation and function types by using some mathematical operations such as square root, delta, and power. For weakly-correlated plots, the region was divided into three parts and ten to thirty points were randomly generated for each filled part separately by using the function. For strongly-correlated plots, the region is divided into five parts and fifty to two hundred points were randomly generated by the function for each filled part to make it denser. To make correlation positive (rising), the pattern of points slopes from lower left to upper right. On the other hand, if it suggests a negative correlation between the variables being studied, the pattern of points slopes from upper left to lower right. To construct a scatter plot belonging to the non-correlation class, points were randomly generated without any direction within the maximum and minimum x-axis and y-axis ranges. Chart styles (i.e., color, the number of points) were also designed randomly. As a result of data generation, the charts in the dataset have different design styles, color schemes, legends, and text regions. , and other irrelevant elements.

The core synthetic data was then augmented by using the "*ImageDataGenerator*" function in the Keras library in Python. In the *data augmentation* process, the direction of the images was transformed with the *shear_range* parameter, and zoomed in and out with the *zoom_range* parameter. Finally, some images were slightly rotated with the *height_shift_range* parameter with a very small random probability (0.08) to prevent overfitting. Image rotation is one of the recommended methods applied to augment datasets when training with a convolution neural network [43]. Some discussions on how image rotation improves accuracy can be found in the literature [44-46].

4.2. Experimental Results

Table 1 shows the accuracy values obtained by each CNN model. All test accuracy results range between 87.8% and 93.9%, so it is possible to say that CNN models have the ability to reach high accuracy in scatter plot classification. The best result was obtained by ResNet50. It can be seen that ResNet50 outperformed the rest with an accuracy of 93.9%. This is probably because of the fact that it is deeper than others and it has the incorporation of skip connections that supports solving the vanishing gradient descent problem. ResNet50 is followed by the AlexNet and VGG19 with the test accuracy values of 91.4% and 91.1%, respectively.

Table 1. Comparison of different deep learning architectures.

Architecture	Train Accuracy (%)	Validation Accuracy (%)	Test Accuracy (%)	Epoch	Transfer Learning	Color
AlexNet	87.60	90.50	91.40	13	No	RGB
OpenCV-CNN	85.40	85.90	87.80	15	No	RGB
OpenCV-CNN	85.13	86.60	89.30	24	No	Grayscale
VGG19	94.07	94.07	91.10	12	Yes	RGB
ResNet50	97.00	93.80	93.90	16	Yes	RGB

Figure 4 shows the comparison of models in terms of recall, precision, and f-measure. The values of these measures are in the range of 0 and 1, being near to 1 the best. From this perspective, ResNet50 is the best model, since its precision, recall and f-measure values are closer to 1 than the other models. This indicates that ResNet50 usually yields better classification results than the rest. For example, ResNet50 (0.9379) outperformed AlexNet (0.9141), VGG19 (0.9098), OpenCV-CNN-GrayScale (0.8950), and OpenCV-CNN-RGB (0.8782) in terms of f-measure. According to the results, it can be concluded that the ResNet50 model is better in scatter plot classification compared to the rest.

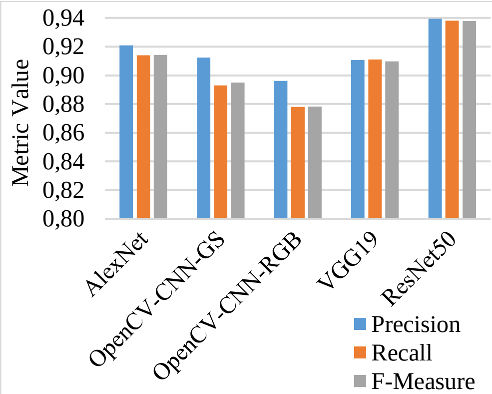


Figure 4. Comparison of the models in terms of precision, recall, and f-measure

Figure 5 shows the confusion matrix of the best model (ResNet50) to present the performance of the method on each scatter plot type separately. Class labels in the matrix are represented by their initials, which are strong-negative (sn), strong-positive (sp), weak-negative (wn), weak-positive (wp), and non-correlation (nc), respectively. It can be seen from the figure that the predictive model usually had no difficulty in identifying chart images. For example, 194 out of 200 instances belonging to the "strong-negative correlation" type were correctly predicted; however, only 6 of them were misclassified by the model. Though scatter plot types were distinguished well with high accuracy values, "weak-negative correlation" and "weak-positive correlation" classes were slightly confused by the model during prediction. As can be seen from the confusion matrix of the best model was achieved on the "non-correlation" type.

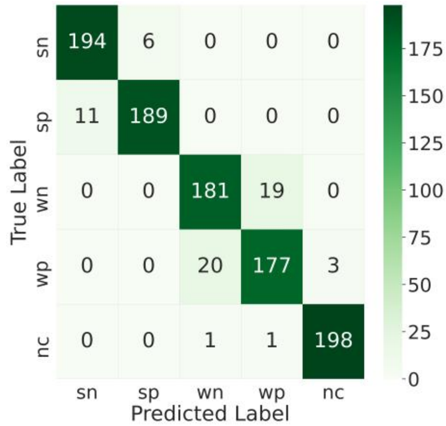


Figure 5. Confusion matrix of the ResNet50 model

Accuracy results of the ResNet50 model are given in Figure 6. It is seen that the train and test accuracy values are close to each other. Besides, as the iteration epoch increases, the train and test accuracy values increase in parallel, which means that there is no overfitting problem. In addition, there are no sharp ups and downs in the plot, which states that the model is well-trained.

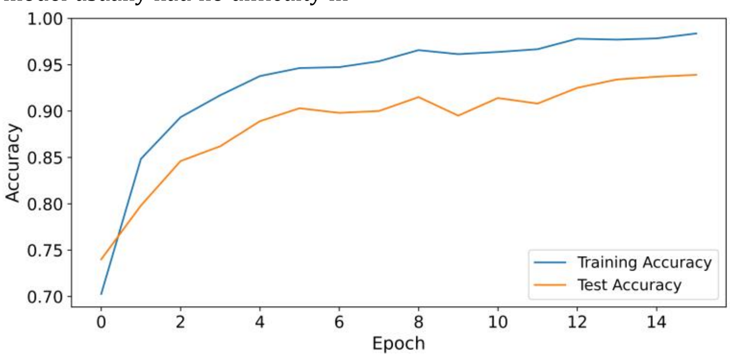


Figure 6. Train and test accuracy values with the ResNet50 model

Figure 7 shows the training times (in seconds) of deep learning models individually. As can be seen, OpenCV-CNN-GS model was constructed faster than the others, since the learning algorithm was applied to the grayscale images. It is followed by AlexNet with a training time of 210 seconds. VGG19 may not be a good choice in terms of execution time.

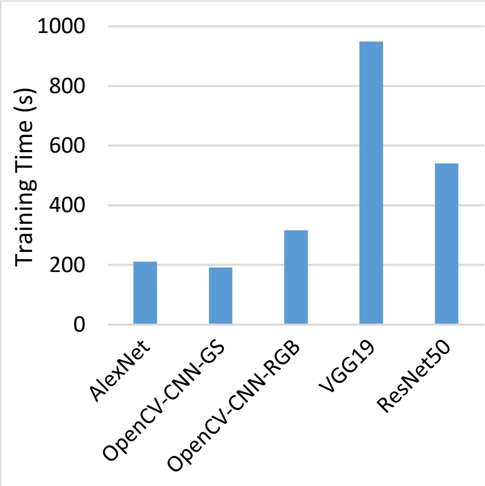


Figure 7. Training times of the deep learning models

It is possible to discuss the validity of the research in four aspects: (i) construct validity, (ii) internal validity, (iii) external validity, and (iv) conclusion validity. *Threats to construct validity* were addressed by using four different evaluation metrics: accuracy, precision, recall, and f-measure. These commonly-used performance metrics were specially selected to overcome the threat of measure selection. According to the results, the predictive validity of the constructed model was proven since all the results (93.90%, 93.94%, 93.80%, and 93.79%, respectively) are higher than the acceptable level (>80%). *Threats to internal validity* related to data collection were addressed by selecting proper search keywords relevant to the topic of interest such as "scatter plot", "scatter diagram", and "scatter graph", and "scatter chart". In addition, to reduce internal validity threats, the collected chart images were manually labeled by three different persons to serve as ground truth. The *external validity threats* were addressed by running all the experiments in the same environment under the same conditions. The validity of the research was increased by performing automatic feature extraction with a deep convolutional neural network and using Adam optimizer to minimize the loss. The *conclusion validity* was addressed by using a statistical test technique. We used the Friedman Aligned Ranks test to ensure that the differences in model performances are statistically significant. Since the p-value obtained from all-vs-all pairwise comparisons (0.04448) is smaller than the significance level (0.05), it is possible to say that the results are statistically significant.

5. Conclusion

Scatter plots aim to make a visual representation of the correlation between two scaled variables and are widely used in many fields. However, since they are in the form of images, they cannot be interpreted by visually impaired users. Therefore, visually impaired people fall behind in accessing this information. In this study, our motivation is to help visually impaired people make sense of scatter diagrams without the help of another person. Towards this purpose, this paper proposes an intelligent system that can interpret a scatter plot for visually impaired people. It is the first study that describes a deep learning-based approach for automatic detection of five classes of scatter plots: strong-positive, weak-positive, non-correlation, strong-negative, and weak-negative correlation.

In the experimental studies, alternative convolutional neural network architectures were compared on both real-world and synthetic datasets in terms of accuracy, including AlexNet, OpenCV-CNN, VGG19, and ResNet50. The experimental results showed that the proposed system could be successfully (93.90%) used by visually impaired users to classify scatter plot images.

As future work, the web application can be further improved in several aspects. In addition to visual features, more information about scatter plot images can also be given to visually impaired people by obtaining textual features such as the title of the plot, the names on the horizontal and vertical axes in the plot, the ranges, figure caption, legends, figure citation in the text, and similar properties. Furthermore, similar studies can be carried out on other chart types according to the needs of visually impaired people and added to this system. Chart options can be diversified in this way.

References

[1] Shao, L., Mahajan, A., Schreck, T., Lehmann, D.J. 2017. Interactive Regression Lens for Exploring Scatter Plots, Computer Graphics Forum, Volume. 36, p. 157-166. DOI: 10.1111/cgf.13176

[2] Wang, W.B., Huang, M.L., Nguyen, Q.V., Huang, W., Zhang, K., Huang, T.H. 2016. Enabling Decision Trend

Analysis with Interactive Scatter Plot Matrices Visualization, Journal of Visual Languages & Computing, Volume. 33, p. 13-23. DOI: 10.1016/j.jvlc.2015.11.002

[3] Sainani, K.L. 2016. The Value of Scatter Plots, Physical Medicine and Rehabilitation (PM&R), Volume. 8, p. 1213-1217. DOI: 10.1016/j.pmrj.2016.10.018

[4] Mohseni, F., Mokhtarzade, M. 2021. The Synergistic Use of Microwave Coarse-scale Measurements and Two Adopted High-resolution Indices Driven from Long-term T-V Scatter Plot for Fine-scale Soil Moisture Estimation, GIScience & Remote Sensing, Volume. 58, p. 455-482. DOI: 10.1080/15481603.2021.1906056

[5] Zhang, Z., Cui, X., Jeske, D.R., Borneman, J. 2013. Biclustering Scatter Plots Using Data Depth Measures, Statistical Analysis and Data Mining, Volume. 6, p. 102-115. DOI: 10.1002/sam.11166

[6] Fu, J., Zhu, B., Cui, W., Ge, S., Wang, Y., Zhang, H., Huang, H., Tang, Y., Zhang, D., Ma, X. 2020. Chartem: Reviving Chart Images with Data Embedding, IEEE Transactions on Visualization and Computer Graphics, Volume. 27, p. 337-346. DOI: 10.1109/TVCG.2020.3030351

[7] Al-Zaidy, R.A., Giles, C.L. 2015. Automatic Extraction of Data from Bar Charts. 8th International Conference on Knowledge Capture, 07-10 October, Palisades, USA, 1-4. DOI: 10.1145/2815833.2816956

[8] Bajic, F., Job, J., Nenadic, K. 2019. Chart Classification Using Simplified VGG Model, International Conference on Systems, Signals and Image Processing, 5–7 June, Osijek, Croatia, 229-233. DOI: 10.1109/IWSSIP.2019.8787299

[9] Araujo, T., Chagas, P., Alves, J., Santos, C., Santos, B., Meiguins, B. 2020. A Real-World Approach on the Problem of Chart Recognition Using Classification, Detection and Perspective Correction, Sensors, Volume. 20, p. 1-21. DOI: 10.3390/s20164370

[10] Deepa, R., Tamilselvan, E. 2020. Processing Of Evaluation Chart Uses Optical Character Recognition, International Journal of Scientific & Technology Research, Volume. 9, p. 4770-4773.

[11] Vassilieva, N., Fomina, Y. 2013. Text Detection in Chart Images, Pattern Recognition and Image Analysis, Volume. 23, p. 139-144. DOI: 10.1134/S1054661813010112

[12] Chagas, P., Freitas, A., Daisuke, R., Miranda, B., De Araújo, T.D.O., Santos, C., Meiguins, B., De Morais, J.M. 2017. Architecture Proposal for Data Extraction of Chart Images Using Convolutional Neural Network, 21st International Conference on Information Visualisation, 11–14 July, London, UK, 318-323. DOI: 10.1109/iV.2017.37

[13] Zan, T., Liu, Z., Wang, H., Wang, M., Gao, X. 2020. Control Chart Pattern Recognition Using the Convolutional Neural Network, Journal of Intelligent Manufacturing, Volume. 31, p. 703-716. DOI: 10.1007/s10845-019-01473-0

[14] Mishra, P., Kumar, S., Chaube, M.K. 2020. Dissimilarity Based Regularized Deep Learning Model for Information Charts, 9th International Conference on Informatics, Electronics & Vision, 26-29 August, Kitakyushu, Japan, 1-6. DOI: 10.1109/ICIEVicIVPR48672.2020.9306660

[15] Zhou, F., Zhao, Y., Chen, W., Tan, Y., Xu, Y., Chen, Y., Liu, C., Zhao, Y. 2021. Reverse-engineering Bar Charts Using Neural Networks. Journal of Visualization, Volume. 24, p. 419-435. DOI: 10.1007/s12650-020-00702-6

[16] Sohn, C., Choi, H., Kim, K., Park, J., Noh, J. 2021. Line Chart Understanding with Convolutional Neural Network, Electronics, Volume. 10, p. 1-17. DOI: 10.3390/electronics10060749

[17] De, P. 2018. Automatic Data Extraction from 2D and 3D Pie Chart Images, IEEE 8th International Advance Computing Conference, 14–15 December, India, 20-25. DOI: 10.1109/IADCC.2018.8692104

[18] Fuqua, D., Razzaghi, T. 2020. A Cost-sensitive Convolution Neural Network Learning for Control Chart Pattern Recognition, Expert Systems with Applications, Volume. 150, p. 1-17. DOI: 10.1016/j.eswa.2020.113275

[19] Zan, T., Su, Z., Liu, Z., Chen, D., Wang, M., Gao, X. 2020. Pattern Recognition of Different Window Size Control Charts Based on Convolutional Neural Network and Information Fusion, Symmetry, Volume. 12, p. 1-13. DOI: 10.3390/sym13010110

[20] Birogul, S., Temür, G., Kose, U. 2020. YOLO Object Recognition Algorithm and "Buy-Sell Decision" Model Over 2D Candlestick Charts, IEEE Access, Volume. 8, p. 91894-91915. DOI: 10.1109/ACCESS.2020.2994282

[21] Siper, M., Makinen, K., Kanan, R. 2021. TABot – A Distributed Deep Learning Framework for Classifying Price Chart Images, Advanced Computing, Volume. 1367. p. 1-15. DOI: 10.1007/978-981-16-0401-0_37

[22] Ünlü, R. 2021. A Robust Data Simulation Technique to Improve Early Detection Performance of a Classifier in Control Chart Pattern Recognition Systems, Information Sciences, Volume. 548, p. 18-36. DOI: 10.1016/j.ins.2020.09.059

[23] Kaur, P., Kiesel, D. 2020. Combining Image and Caption Analysis for Classifying Charts in Biodiversity Texts, 15th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications, 27-29 February, Valletta, Malta, 157-168. DOI: 10.5220/0008946701570168

[24] Zaman, M., Hassan, A. 2021. Fuzzy Heuristics and Decision Tree for Classification of Statistical Feature-Based Control Chart Patterns, Symmetry, Volume. 13, p. 1-12. DOI: 10.3390/sym13010110

[25] Mishra, P. & Kumar, S., Chaube, M.K. 2020. Interpretation and Segmentation of Chart Images Using h-Means Image Clustering Algorithm, International Conference on Data Management,

Analytics and Innovation, 17–19 January, New Delhi, India, 379-391. DOI: 10.1007/978-981-15-5616-6_27

[26] Sharma, M., Gupta, S., Chowdhury, A., Vig, L. 2019. ChartNet: Visual Reasoning Over Statistical Charts Using MAC-Networks, International Joint Conference on Neural Networks, 14-19 July, Budapest, Hungary, 1-7. DOI: 10.1109/IJCNN.2019.8852427

[27] Mishchenko, A., Vassilieva, N. 2011. Chart Image Understanding and Numerical Data Extraction, 6th International Conference on Digital Information Management, 26–28 September, Australia, 115-210. DOI: 10.1109/ICDIM.2011.6093320

[28] Mishchenko, A., Vassilieva, N. 2011. Model-based Chart Image Classification, International Symposium on Visual Computing, Advances in Visual Computing, Lecture Notes in Computer Science, Volume. 6939, p. 476-485. DOI: 10.1007/978-3-642-24031 -7_48

[29] Maboudou-Tchao, E.M. 2020. Change Detection Using Least Squares One-class Classification Control Chart, Quality Technology & Quantitative Management, Volume. 17, p. 609-626. DOI: 10.1080/16843703.2019.1711302

[30] Mishra, P., Kumar, S., Chaube, M.K. 2020. ChartFuse: A Novel Fusion Method for Chart Classification Using Heterogeneous Microstructures, Multimedia Tools and Applications, Volume. 80, p. 10417-10439. DOI: 10.1007/s11042-020-10186-z

[31] Direncioğlu Diren, D., Boran, S., Cil, I. 2020. Integration of Machine Learning Techniques and Control Charts in Multivariate Processes, Scientia Iranica, Volume. 27, p. 3233-3241, DOI: 10.24200/sci.2019.50377.1667

[32] Kalteh, A.A., Babouei, S. 2020. Control Chart Patterns Recognition Using ANFIS with New Training Algorithm and Intelligent Utilization of Shape and Statistical Features, ISA Transactions, Volume. 102, p. 12-22. DOI: 10.1016/j.isatra.2019.12.001

[33] Kadakadiyavar, S., Ramrao, N., Singh, M.K. 2019. Efficient Mixture Control Chart Pattern Recognition Using Adaptive RBF Neural Network, International Journal of Information Technology, Volume. 12, p. 1271-1280. DOI: 10.1007/s41870-019-00381-z

[34] Shao, Y.E., Hu, Y.T. 2020. Using Machine Learning Classifiers to Recognize the Mixture Control Chart Patterns for a Multiple-input Multiple-output Process, Mathematics, Volume. 8, p. 1-14. DOI: 10.3390/math8010102

[35] Dai, W., Wang, M., Niu, Z., Zhang, J. 2018. Chart Decoder: Generating Textual and Numeric Information from Chart Images Automatically, Journal of Visual Languages & Computing, Volume. 48, p. 101-109. DOI: 10.1016/j.jvlc.2018.08.005

[36] Tang, B., Liu, X., Lei, J., Song, M., Tao, D., Sun, S., Dong, F. 2016. DeepChart: Combining Deep Convolutional Networks and Deep Belief Networks in Chart Classification, Signal Process, Volume. 124, p. 156-161. DOI: 10.1016/j.sigpro.2015.09.027

[37] Singh, M., Goyal, P. 2021. ChartSight: An Automated Scheme for Assisting Visually Impaired in Understanding Scientific Charts, 16th International Joint Conference on Computer Vision, 8-10 February, Setubal, Portugal, 309-318. DOI: 10.5220/0010201203090318

[38] Savva, M., Kong, N., Chhajta, A., Li, F.F., Agrawala, M., Heer, J. 2011. ReVision: Automated Classification, Analysis and Redesign of Chart Images, 24th Annual ACM Symposium on User Interface Software and Technology, 16-19 October, California, USA, 393-402. DOI: 10.1145/20471 96.20472 47

[39] Jung, D., Kim, W., Song, H., Hwang, J., Lee, B., Kim, B., Seo, J. 2017. ChartSense: Interactive Data Extraction from Chart Images, Conference on Human Factors in Computing Systems, 6-11 May, Denver Colorado, USA, 6706-6717. DOI: 10.1145/30254 53.30259 57

[40] Krizhevsky, A., Sutskever, I., Hinton, G.E. 2012. ImageNet Classification with Deep Convolutional Neural Networks, International Conference on Neural Information Processing Systems, 3-6 December, Nevada, USA, 1097–1105.

[41] Simonyan, K., Zisserman, A. 2015. Very Deep Convolutional Networks for Large-scale Image Recognition, 3rd International Conference on Learning Representations, 7-9 May, San Diego, CA, USA, 1–14.

[42] He, K., Zhang, X., Ren, S., Sun, J. 2016. Deep Residual Learning for Image Recognition, IEEE Conference on Computer Vision and Pattern Recognition, 26 June-1 July, Las Vegas, USA, 770-778. DOI: 10.1109/CVPR.2016.90

[43] Jia, S., Wang, P., Jia, P., Hu, S. 2017. Research on Data Augmentation for Image Classification Based on Convolution Neural Networks. Chinese Automation Congress, 20-22 October, Jinan, China, 4165-4170. DOI: 10.1109/CAC.2017.8243510

[44] Wicaksono P. 2016. Improving the Accuracy of Multispectral-based Benthic Habitats Mapping Using Image Rotations: The Application of Principle Component Analysis and Independent Component Analysis, European Journal of Remote Sensing, Volume. 49, p. 433-463. DOI: 10.5721/EuJRS20164924

[45] Liang, G., Hong, H., Xie, W., Zheng L. 2018. Combining Convolutional Neural Network With Recursive Neural Network for Blood Cell Image Classification, IEEE Access, Volume. 6, p. 36188-36197. DOI: 10.1109/ACCESS.2018.2846685

[46] Somasundaram, D. 2019. Machine Learning Approach for Homolog Chromosome Classification, International Journal of Imaging Systems and Technology, Volume. 29, p. 161-167. DOI: 10.1002/ima.22287