Data Visualization Classification Using Simple Convolutional Neural Network Model

Original Scientific Paper

Filip Bajić
University Computing Centre (SRCE),
Josipa Marohnića 5, 10000 Zagreb, Croatia
filip.bajic@srce.hr

Josip Job
J. J. Strossmayer University of Osijek,
Faculty of Electrical Engineering, Computer Science and Information Technology
Kneza Trpimira 2B, 31000 Osijek, Croatia
josip.job@ferit.hr

Krešimir Nenadić
J. J. Strossmayer University of Osijek,
Faculty of Electrical Engineering, Computer Science and Information Technology
Kneza Trpimira 2B, 31000 Osijek, Croatia
kresimir.nenadic@ferit.hr

Abstract *– Data visualization is developed from the need to display a vast quantity of information more transparently. Data visualization often incorporates important information that is not listed anywhere in the document and enables the reader to discover significant data and save it in longer-term memory. On the other hand, Internet search engines have difficulty processing data visualization and connecting visualization and the request submitted by the user. With the use of data visualization, all blind individuals and individuals with impaired vision are left out. This article utilizes machine learning to classify data visualizations into 10 classes. Tested model is trained four times on the dataset which is preprocessed through four stages. Achieved accuracy of 89 % is comparable to other methods' results. It is showed that image processing can impact results, i.e. increasing or decreasing level of details in image impacts on average classification accuracy significantly.*

Keywords *– data visualization, chart image classification, convolutional neural networks, computational modeling, chart recognition*

1. INTRODUCTION

The era in which we live can undoubtedly be declared as the age of the data. Data science is a multidisciplinary field that uses a vast spectrum of skills to extract knowledge and information from data. Data science field employs mathematics, statistics, analytics and programming skills with techniques like artificial intelligence, machine learning, data mining and data visualization. Every day, an increasing amount of raw data is generated. Data scientist help turn huge data tables into valuable information that are easier to read and interpret. For the end-user, only summation data obtained by statistical data analysis is important. These statistics are most often depicted by figures that are sometimes difficult to read, and of which significant data is difficult to notice. Various graphical presentations are used to highlight essential information, which aims to be as simple as possible. From the need to show as much data as possible in the most readable way, to enable easier transfer and understanding of information, to show data links and to simplify decision-making, data visualization is created [1].

Data visualizations (bars, lines, pies, etc.) contain key data that is not listed anywhere in the text and enable the reader to get useful information and store it in long-term memory. Data visualizations also help to find trends in data, eases finding areas that need improvement, bring out correlations and key details from data, help make analysis reports and make everything visually appealing. There is a substantial amount of information available on the Internet. Internet search engines rely on image metadata instead of image content. The metadata is any auxiliary information stored within a file, which mostly does

not contain enough information about the image they represent, so the search engines "do not see" many valuable results of a user query. As tested on our image dataset, metadata (title and subject or tags or comments) is included in only 39 of 2702 collected images from Google image search which is 1.44% of the total dataset, Table 1.

Table 1. Metadata analysis of collected dataset

TYPE	Title	Subject	Tags	Comments
Area	2	2	0	0
Bar	6	4	2	1
Line	5	0	1	4
Map	14	9	6	6
Pareto	3	3	1	0
Pie	4	4	0	0
Radar	2	2	0	0
Scatter	0	0	0	0
Table	1	1	0	0
Venn	2	2	2	2
	39	**27**	**10**	**11**

We have noticed that title and subject attribute contain the same information which is also a title displayed on the image (if such exists).

Internet search engines are not the only ones faced by these problems. People with impaired vision and all blind individuals have even more problems in accessing data. To access the information or to navigate the document these people must use various screen readers. When a screen reader encounters a data visualization, it can at best read the title that is most commonly displayed under the visualization itself (more advanced screen readers can access the tags in a document). It is suggested that authors add descriptive text to any visualization that screen readers can effectively interpret, but these guidelines are not standardized and are generally ignored by authors. Screen readers use standard Optical Character Readers (OCR) that cannot obtain visualization information. The text below the visualization is not enough for the reader to guess what the visualization should be. It rarely contains details such as the type of data visualization or relationship between values [2, 3].

This work is created to enable Internet search engines and character readers to accurately label each data visualization with its corresponding type. Without this information, any advanced (automatic) image data interpretation is not possible, since classifying images is the first step. Classification is limited to ten most common data visualizations, namely: area charts, bar charts, line charts, maps, Pareto charts, pie charts, radar charts, scatter plots, tables, and Venn diagrams. Data visualizations that contain subcategories (e.g. horizontal, vertical, stacked or grouped bar chart) are treated as belonging to one (main) category (bar chart).

We have organized the rest of this paper in the following way. Section II presents the current research and brief information on the most significant scientific papers. Section III shows the Convolutional Neural Network (CNN) model that is used in the experiment. Section IV gives information about the used image dataset and the results of the research. Finally, Section V shows the final remarks on this experiment and instructions for improvements.

2. RELATED WORK

Several articles describe retrieving data from images, a process called reverse engineering. By decreasing the scope of research exclusively to retrieve visualization information and to obtain a summary or data table, there are 47 scientific publications available at the moment of writing. All articles are publicly available and can be found in numerous online libraries such as IEEE Xplore, ACM, Semantic Scholar, ResearchGate, etc. Scientific papers dealing with the research of the above-mentioned issue also deal with the classification of the input visualization image, since they must first determine which category belongs to the loaded visualization to allow any descriptive function of the visualization. The obtained articles can be divided into three categories:

- Authors that are using Hough transformation, Hidden Markov model, vectorization, the histogram of gray color or any other method for obtaining data visualization type [4, 5, 6]
- Authors that are using Bayesian Network (BN) for obtaining data visualization type [2, 7, 8, 9]
- Authors that are using Neural Networks (NN) for obtaining data visualization type

Some publications stand out in the literature for proposing techniques and methodologies for visualization type classification. The mentioned papers are sorted chronologically from the old to the newer.

Huang et al. [6] presented a system capable of classifying the type of visualization and interpreting data. The proposed model is using raster-to-vector conversion, which can detect lines and arcs. Using vectorized lines and arcs, the model can classify input data visualization into four categories. Authors are also using a feature extraction which consists of text and graphics separation and edge detection. OCR is used over the image that contains textual information. Over the image that contains graphical information, edge detection and vectorization are applied. In the next phase, arcs and lines are represented by a set of vectors. By checking the relationships between the lines, authors can identify specific objects with which they can determine data visualization type.

Ferres et al. [10] presented The iGraph-lite System, an application that helps blind individuals and individuals with impaired vision interacting with data visualizations. iGraph-Lite system also generates textual description and a summary of visualization, and its main goal is lan-

guage-based interactivity with a user. The system uses messages (templates) which give information about the data visualization. These templates consist of slots (variables) that are filled when the visualization is processed.

Elzer et al. [2] focused work on understanding simple bar charts. Authors identify the communicative signals that appear in simple bar chart visualization and supporting caption, and present BN methodology for reasoning about these signals and hypothesizing visualization intended message. The proposed system architecture consists of a visual extraction module for analyzing the image, tagging module for extraction information from the caption and a module for recognizing the intended message.

Savva et al. [11] presented ReVision, a tool whose main goal is to create a new data visualization out of the existing visualization. ReVision can classify up to 10 types of visualizations and is considered a "state-of-the-art" tool. The tool uses computer vision and machine learning techniques for the input image classification and achieves the classification of the input visualization whose average rating is about 80%. The input visualization goes through three different stages. The classification of the input visualization is done in the first stage. Machine learning and computer vision are used to study the features of data visualization. After the classification, it is possible to find graphic marks (arrows, lines, and points), link them to the corresponding text (values on the axes) and export the data to the table. The third stage uses a data table from the previous stage to generate a new type of data visualization.

Jung et al. [3] presented ChartSense, an interactive system for classifying and extracting data from images. The system adopts pipeline for data extraction proposed in ReVision. ChartSense uses CNN for classifying input data visualization and then extracts underlying data using semi-automatic, interactive extraction algorithms. The authors assessed the efficacy of ChartSense by contrasting the accuracy of its classification with ReVision. The used dataset was as the one used in ReVision and then expanded with additional images collected with Google. The average classification accuracy is about 90%.

Poco and Heer [12] utilized CNN for the input image classification and achieved a total average rating of 94%. Their main input is a text assessment pipeline that recognizes text objects in the data visualization process (with bounding boxes), reads text content using OCR and classifies their role in the chart. The authors are using their own set of images as well as the set of images used by ReVision. The CNN was trained on half a million images. Filters are applied over input images for easier separation of text and highlighting graphics which contribute a more precise classification. The outcome of the image (graphic) classification of the input is contrasted with the result of the OCR, which also impacts the accuracy of the classification. The authors also compare their results with ReVision and ChartSense.

As shown, authors are using different methods for different end goals. The choice of method has a high impact on data visualization classification accuracy. It determines the complexity of the application, required computer power, number of images, time to produce a result and a number of classes in which input image can be classified. All the above-mentioned articles have one thing in common – classification of the input visualization. The NNs (CNNs) stand out in classifying input data visualization and in extracting information [13, 14, 15, 16]. Since the visualization images can have a lot of noise and distortions, CNN seems to be the best solution for the aforementioned problem.

3. THE MODEL

In this section, basic information about VGG (name after Oxford's team Visual Geometry Group) model and detailed information on the layer configuration of the model used in this research is provided. We also show the process of choosing the NN architecture for our research.

Since the number of NN architectures is growing each day, we had to choose the correct architecture for our research. In the pool of currently available NN architectures, we have set our own list of requirements for NN. The NN architecture should comply with:

- Image classification on ImageNet dataset - ImageNet is a project that aims to manually tag and classify images into more than 20000 classes to be used for computer vision research. When ImageNet is mentioned in the context of machine learning and CNNs, the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) is discussed. This classification aims to produce a model that effectively classifies the input image in 1000 distinct classes. Because of this competition image classification error has been reduced to less than 8%, and each year the new "state-of-the-art" architectures are presented.
- Generalization - The same architecture can be used in different applications or with different data types. The network cannot be limited to one specific goal or one data type and should support process called transfer learning. Transfer learning utilizes knowledge from previously learned tasks and applies them to newer (pre-trained models).
- Scalability - The architectures with homogenous topology represent the best choice. The number of layers can be easily changed. Changing the number of layers in a network changes the depth of network which is in correlation with the total number of parameters. This tuning allows us to get the best results with the network as simple as possible.
- Simple implementation - The architecture and the network must be publicly available and free of charge. This allows us to compare results with

other teams or use pre-trained models. The network can be implemented without knowing advanced programming languages and should have written documentation.

- Low computing power - Network should be able to run on a normal (home) computer due to our limiting computing power and training of the network should not last more than an hour.

Simonyan and Zisserma [17] proposed the architecture that fulfills all requirements for image classification. The VGG is modular in layers (in [17] there are six configurations, the depth of the network increases from left to right: A, A-LRN, B, C, D and E), has homogenous topology and is simple [18]. Although, VGG was not at the top place of the 2014 ILSVRC competition it showed good results in image classification. The main limitation associated with VGG is high computing power. Even though VGG uses small 3 x 3 filter layers the use of about 140 million parameters requires high computing power and long training time (the simplified model has only ~ 13 million parameters, Table 2).

The NN consists of many layers that can be of different types. The number of layers and the type of layers best used is free of choice. The VGG16 network is designed to function with a very large quantity of data and to classify the input image into several hundred classes. The network is considered a "deep network" because it contains 16 weight layers. The weight layers are only the convolutional layers and the fully connected layers because they contain the parameters that can be learned. Deep NNs also require a large amount of computing power. The purpose of this research idea is not to use the VGG16 network in its entirety, but to use the scaled-down implementation to see the performance of a simplified model. The model from Table 2 consists of 7 weight layers (5 convolutional layers and 2 fully connected layers). As noted in [17] the simplest model (A) contains 11 weight layers and the total number of parameters is around 133 million.

Detailed setup of the model is shown in the Table 2 [19]. As noted in [11] the best classification accuracy is achieved with images of 128 x 128 pixels. The input model utilizes a static 128 x 128 image containing RGB colors. The first convolution layer contains 32 filters with a 3 x 3 kernel. Rectified Linear Unit (ReLU) activation is used followed by a Normalization layer. The Pooling layer uses a 3 x 3 size window for a quick spatial reduction of the input image from 128 x 128 to 42 x 42. Due to the small quantity of data, it is essential to use an extra Dropout layer that prevents network overfit. Additional convolutional layers are added without a Pooling layer. A bigger amount of convolution layers in a row without a compression layer makes it possible to learn a bigger set of features. The operation is performed by incorporating a few of the layers mentioned above, the only distinction being the size of the filter. A fully connected layer that uses ReLU activation and normalization is located at the end of the model.

The proposed model from Table 2 is trained four times from scratch. Input dataset is preprocessed through four stages, if image resize is not considered as separate stage. At each stage obtained dataset is used for model training, i.e. the same dataset was used as input but with different image processing applied to it at specific stage. In testing of the models, the same input dataset is used but it is also preprocessed according to the current stage image transformations so the datasets used for training, validation and testing are matched according to applied processing.

Table 2. Network model with number of parameters for each layer

Layer (type)	Output shape	Number of parameters
Convolution	None, 128, 128, 32	896
Activation	None, 128, 128, 32	0
Normalization	None, 128, 128, 32	128
Pooling	None, 42, 42, 32	0
Dropout	None, 42, 42, 32	0
Convolution	None, 42, 42, 64	18496
Activation	None, 42, 42, 64	0
Normalization	None, 42, 42, 64	256
Convolution	None, 42, 42, 64	36928
Activation	None, 42, 42, 64	0
Normalization	None, 42, 42, 64	256
Pooling	None, 21, 21, 64	0
Dropout	None, 21, 21, 64	0
Convolution	None, 21, 21, 128	73856
Activation	None, 21, 21, 128	0
Normalization	None, 21, 21, 128	512
Convolution	None, 21, 21, 128	147584
Activation	None, 21, 21, 128	0
Normalization	None, 21, 21, 128	512
Pooling	None, 10, 10, 128	0
Dropout	None, 10, 10, 128	0
Flatten	None, 12800	0
Dense	None, 1024	13198224
Activation	None, 1024	0
Normalization	None, 1024	4096
Dropout	None, 1024	0
Dense	None, 10	10250
Activation	None, 10	0

Total number of parameters: 13401994
Number of trainable parameters: 13299114
Number of non-trainable parametres: 2880

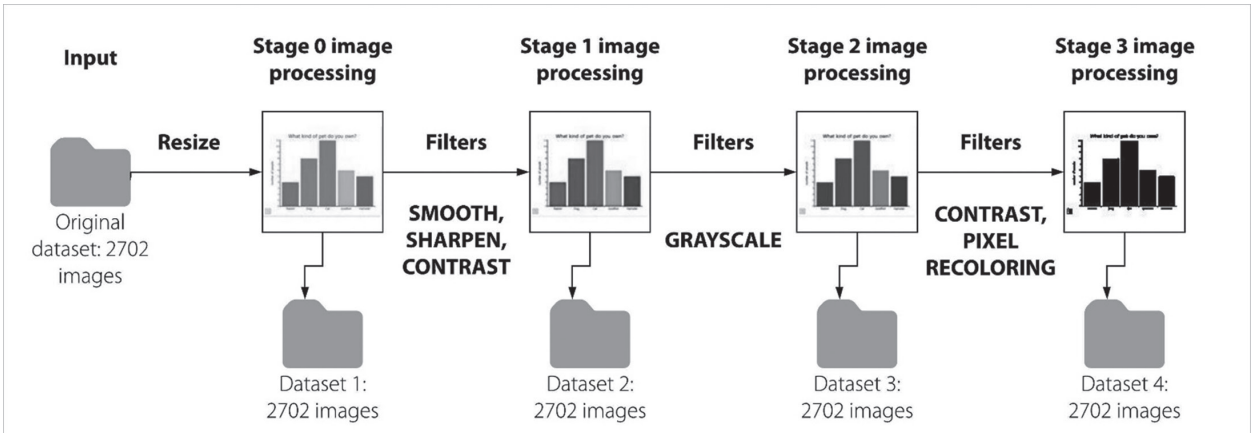

Fig. 1. The process of generating datasets

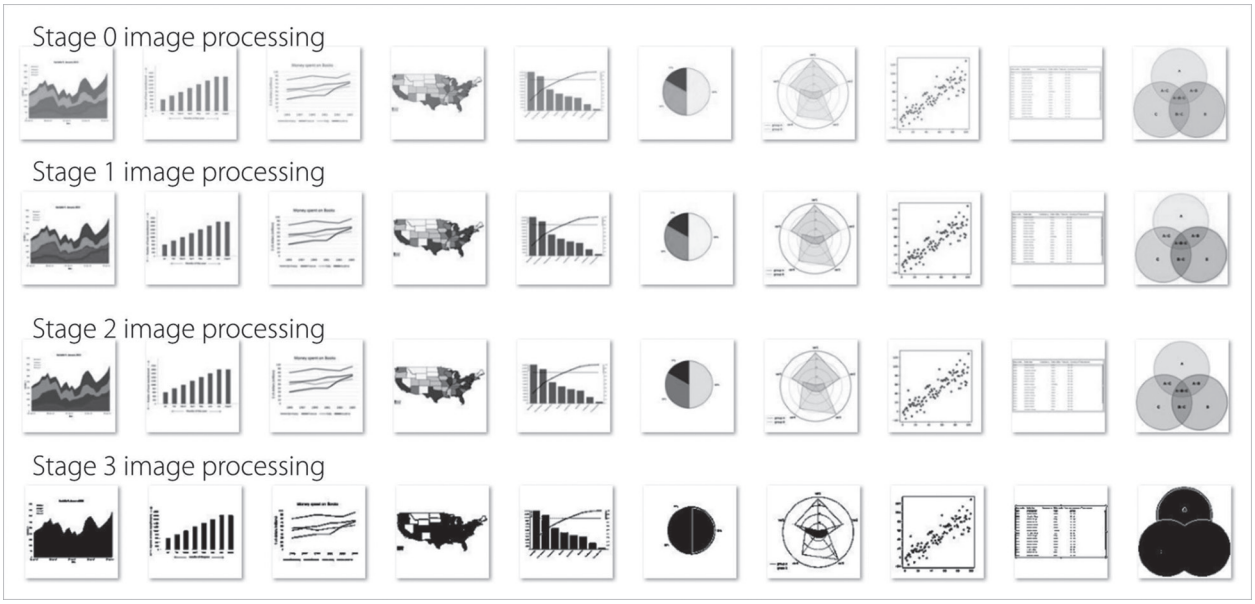

Fig. 2. Randomly selected visualization examples from Google Image search, from left to right: area, bar, line, map, Pareto, pie, radar, scatter, table and Venn. Input images, after resize are in the first row. The rest of the rows are images created using advanced image processing, Fig. 1.

4. EXPERIMENTAL RESULTS

This section provides the configuration used in the classification of data visualization, all datasets used in the method, and the process how to create them. The comparison between each dataset is shown in Table 4. We also show how our research compares with other researches in this field.

4.1. EXPERIMENTS SETUP

For CNN implementation, TensorFlow [20] and Keras [21] are used. The network was trained on laptop with Intel i5-8250U and 24GB DDR4 RAM. The laptop runs Microsoft Windows 10 64-bit operating system.

4.2. THE DATASET

CNN requires three datasets. The biggest data set is used to train the network. The second dataset is used for validation purposes. The best practice is to divide the training set into two batches. First, the biggest batch includes 80% and is used to train the network, and the second batch is used to validate the network.

The third dataset may be of arbitrary size and no image may be contained in any of the two previous datasets. This dataset is used to validate the entire network, i.e. the performance of the model.

Two major datasets of images used for CNN training, testing and validation process are as follows:

- Images collected from the Google Image search engine
- Images used in existing ReVision system

Using Google Image search, we obtained 2702 distinctive images divided into 10 classes as shown in the Table 3.

The publicly available repository of ReVision dataset contains links to used images. The repository is not

maintained and vast number of links resolves in error or points to irrelevant images. During the period of this research the number of available images resulted with 610 images from the original dataset. Those images are also filtered and the dataset is reduced to 30 images per data visualization type.

All images have been manually classified and divided into respective groups. Images with the following features:

- More than one data visualizations on the same image
- Partially showing a data visualization
- Watermarks
- Resolution was below 500 x 500
- Transparent background
- 3D data visualizations
- Image format is not .jpg/.jpeg

were manually excluded from the dataset.

All collected images we processed four times and created four datasets (dataset 1 - 4), Fig. 1. Since all collected images were in different resolutions and the input to the model expects an image size of 128 x 128 pixels, we manually scaled-down all collected images, dataset 1. This is the only preprocessing that was used on the image dataset 1. In order to achieve the best possible results, all images are scaled down with a preserved aspect ratio. White padding is added for the images that had lower resolution than 128 x 128 pixels, Fig. 3. By applying filters (smooth, sharpen and contrast) we created the dataset 2. Dataset 3 is created by transferring the second dataset into grayscale. To create the dataset 4 we applied high contrast and pixel recoloring on the previous dataset. Total number of images in each dataset is 2702. The process from Fig. 1 is also used on ReVision dataset.

Fig. 3. Left image – scaled down map visualization without preserved aspect ratio. Right image – scaled down map visualization with preserved aspect ratio and added white padding on the bottom.

For manual validation of the entire NN, we have chosen 30 images per class from the ReVision dataset (300 in total for each dataset). None of the images contained within the ReVision dataset were used during the network training and testing process.

Table 3. Image dataset overview (each of the datasets contains the same number of data visualizations)

TYPE	Google Image search			ReVision
	Collected	Training	Testing	Validation
Area	278	222	56	30
Bar	291	233	58	30
Line	257	206	51	30
Map	116	93	23	30
Pareto	198	158	40	30
Pie	412	330	82	30
Radar	371	297	74	30
Scatter	258	206	52	30
Table	373	298	75	30
Venn	148	118	30	30
	2702	**2161**	**541**	**300**

4.3. THE RESULTS

The model from Table 2 was trained from scratch on each dataset for 30 epoch. The performance of the model trained on dataset 4 is shown in Fig 4 and Fig. 5.

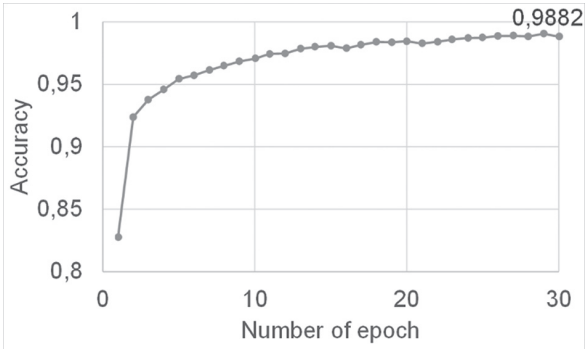

Fig. 4. Model performance after 30 epoch on training dataset (dataset 4)

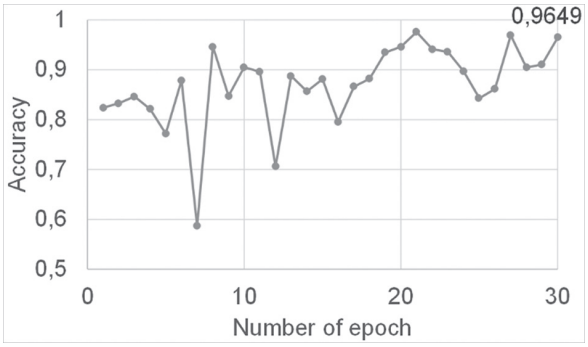

Fig. 5. Model performance after 30 epoch on testing dataset (dataset 4)

The maximum accuracy achieved on training dataset is 0.9882 or 98.82%. The maximum accuracy achieved on testing dataset is 0.9649 or 96.49%.

Each data visualization obtained is distinct in its form (colors, text, legend, orientation, position, labels, marks, etc.). There is also a difference between the Google Image set and ReVision set. The ReVision set consists of images that are fairly more complex and have an average resolution of 50 percent lower than the images collected from the Google Image search.

The Table 4 shows how the network evaluates on ReVision dataset. The images were selected from original dataset and then paired with corresponding copies from each dataset.

Table 4. Results of network evaluation on ReVision dataset

TYPE	Accuracy [%]			
	Stage 0	Stage 1	Stage 2	Stage 3
	Dataset 1	Dataset 2	Dataset 3	Dataset 4
Area	66.67	63.33	86.67	93.33
Bar	66.67	60.00	63.33	90.00
Line	76.67	60.00	63.33	86.67
Map	80.00	76.67	76.67	80.00
Pareto	60.00	56.67	63.33	83.33
Pie	93.33	93.33	96.67	100.00
Radar	80.00	66.67	76.67	86.67
Scatter	83.33	80.00	80.00	86.67
Table	90.00	90.00	90.00	83.33
Venn	86.67	86.67	93.33	100.00
	78.33	**73.33**	**79.00**	**89.00**

The Stage 0 dataset achieved decreased results compared to previous research (81.67 %) [22]. Previous network model was using 96 x 96 pixels input image that contains lower number of details. Stage 1 image processing achieved the worst results compared to other three columns. In this stage, details on image are increased and text is sharp and readable what can result in wrong classification, e.g. some data visualization with coordinate system could be misclassified as table because of emphasized lines. Transferring images into grayscale in Stage 2 improves classification of area, pie and Venn. Stage 3 image processing reduces number of details on image, removes color, and text is distorted and not readable. The image contains only the shape (footprint) of data visualization that is black. Stage 3 achieves the highest average accuracy. For further comparison, the results from Stage 3 will be used.

Training the same model with the same input dataset but with different image processing applied shows that performance of the model can be further increased or decreased without changing the depth of the NN or any of the parameters. It is important to understand which level of detail is required for specific task in computer vision.

4.4. THE COMPARISON

To see how our results compare with other scientific papers we analyzed their reported achievements. We compare our research only with papers that use the same 10 visualization data types as we do, Table 5 and Table 6. We also exclude papers that do not report the achieved classification accuracy by data visualization type. As far as our research goes, only three papers contain all the required data for comparison, Table 5.

Table 5. Analysis of relevant papers

Research	[12]	[3]	Proposed method	[11]
Method	CNN: AlexNet	CNN: GoogLeNet	CNN: Simplified VGG	Support Vector Machine (SVM)
Pretrained model	Yes	No	No	No
Total images	5125	5659	2702	2601
Training set [%]	75	80	80	80
Testing set [%]	25	20	20	20
Validation set	2000+	-	300	-
Average accuracy [%]	94	90	89	80
Image processing	Yes	Yes	Yes	Yes
OCR	Yes	Yes	No	Yes

As seen from Table 5, only two other types of research are using CNN for input image classification. All these CNNs are made of different layer composition and all of them have different depth and a different number of parameters. As noted in [23], the depth of the network is not crucial in achieving the best results. For splitting dataset into training and testing dataset, we are all using the best practice. Researchers that do not report the number of validation set data are noted as "-". All other researchers used advanced image processing such as color corrections, edge emphasizing, noise reduction and (or) text/graphics separation.

Since data visualization title is often part of the image itself, OCR can be used to further increase classification accuracy by comparing OCR results with data visualization classification result. Table 6 shows a detailed comparison by data visualization type as reported by researches. In a total of four related types of research, we rank ourselves in third place by average classification accuracy. Without OCR, we achieved the best results in pie and Venn classification.

Table 6. Comparison of achieved data classification accuracy by type

TYPE	Accuracy [%]			
	[12]	[3]	Proposed method	[11]
Area	95.00	67.00	93.33	88.00
Bar	97.00	93.00	90.00	78.00
Line	94.00	78.00	86.67	73.00
Map	96.00	88.00	80.00	84.00
Pareto	89.00	85.00	83.33	85.00
Pie	98.00	92.00	100.00	79.00
Radar	93.00	86.00	86.67	88.00
Scatter	92.00	86.00	86.67	79.00
Table	98.00	94.00	83.33	86.00
Venn	91.00	67.00	100.00	75.00
	94.00	**90.00**	**89.00**	**80.00**

5. CONCLUSION

With the rapid development of the Internet and exponential growth in data, there is an increasing number of data visualizations that are not only a problem for blind people or people with impaired vision but also Internet search engines. As seen from our dataset, only 1.44% of collected images contain metadata which is the primary source of information when an embedded image does not contain descriptive tags. Modern research is focused on the use of NN to classify the presented data visualization correctly and generate a text description. The problem with this approach is the need for a large variety of different data visualization images to enable the NN to identify efficiently what it requires. The need for greater quality and quantity of data also means the need for ever-increasing computing power, and these are important parameters that have an impact on the overall performance of the model. There is no layer-wise guidance for NN, which parameters should be used or how deeply the network should grow. Everything is reduced to the use of the success and fail method that requires a considerable amount of time. Changing only one parameter may have a "hit" result on the network in the context that the classification is significantly improved or deteriorated. The best practice is to stick to the well-known classification models or to create a new model based on the existing ones. There is no "fine-tuning" on the model used in this paper. A standard model is used, which can achieve a classification accuracy of 70% on various datasets. Due to the small dataset used to train a network, it is always necessary to use Dropout layers to stop the network from overfitting. In higher network training stages, Dropout layer does not prevent the network from overfitting too quickly, and a negative effect occurs. Image processing and reducing the number of details in image can have a major impact in increasing classification accuracy. We switch RGB colors in image with black, as color is rarely indicative of data visualization type. Since we are not using OCR, the text fields in image present unnecessary level of detail. With such decreased level of detail, we achieve average classification accuracy of 89% across 10 visualization data types.

It is showed that images filtering can boost performance of the model significantly, therefore it is necessary to notice that the lack of unique, publicly available, dataset may impact the results presented in various scholar papers so the comparison of different methods could be considered as comparing apples to oranges if the used dataset is not similar in size and features. One of the aims of the future research will be to make publicly available dataset which could be used for the testing purposes to make comparison of various methods for data visualization classification viable.

In the future, we plan to create an image dataset that will consist of 1000 unique images per data visualization type. We also plan to increase the number of data visualization types to 20 which will include a circle pack, sunburst diagram, heat map, gauge chart, funnel chart, box plot, tree diagram, word cloud, matrix and node diagram. Stage 2 image processing increases readability of all text fields and marks that can be paired with OCR and further increase average classification accuracy. Pairing Stage 2 and improved Stage 4 image processing text/graphics separation can be achieved and additional features could be extracted. Other NN models could be tested for comparison purposes. When finding the best stock model, the fine-tuning will be applied to achieve the best performance in this specific task.

6. ACKNOWLEDGEMENTS

The current archival periodical article is based on the conference presentation [22].

7. REFERENCES

[1] D. Chester, S. Elzer., "Getting Computers to See Information Graphics so Users Do Not Have to", Proceedings of the International Symposium on Methodologies for Intelligent Systems, Saratoga Springs, NY, USA, 25-28 May 2005, pp. 660-668.

[2] S. Elzer, E. Schwartz, S. Carberry, D. Chester, S. Demir, P. Wu. "A browser extension for providing visually impaired users access to the content of bar charts on the web". Proceedings of the 3rd International Conference on Web Information Systems and Technology, Barcelona, Spain, 3-6 March 2007, pp. 59-66.

[3] D. Jung, W. Kim, H. Song, J. Hwang, B. Lee, B. Kim, J. Seo., "ChartSense: Interactive Data Extraction from Chart Images", Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems, New York, NY, USA, 2017, pp. 6706-6717.

[4] Y. P. Zhou., C. L. Tan, "Learning-based scientific chart recognition", Proceedings of the 4th IAPR International Workshop on Graphics Recognition, Kingston, ON, Canada, 7-8 September 2001, pp. 482-492.

[5] A. Telea, A. Maccari, C. Riva, "An open toolkit for prototyping reverse engineering visualizations", Proceedings of the Symposium on Data Visualisation, Aire-la-Ville, Switzerland, May 2002, pp. 241-249.

[6] W. Huang, C. L. Tan, W. K. Leow, "Model-Based Chart Image Recognition", Proceedings of the International Workshop on Graphics Recognition, Barcelona, Spain, 30-31 July 2003, pp. 87-99.

[7] S. Schwartz, S. Carberry, I. Zukerman, "The automated understanding of simple bar charts", Artificial Intelligence, Vol. 175, No. 2, 2011, pp. 526-555.

[8] S. Demir, S. Carberry, K. Mccoy, "Summarizing Information Graphics Textually", Computational Linguistics, Vol. 38, No. 3, 2012, pp. 527-574.

[9] S: Demir, S. Elzer Schwartz, R. Burns, S. Carberry, "What is being Measured in an Information Graphic?", Proceedings of the International Conference on Intelligent Text Processing and Computational Linguistics, Samos, Greece, 24-30 March 2013, pp. 501-512.

[10] L. Ferres, P. Verkhogliad, G. Lindgaard, L. Boucher, A. Chretien, M. Lachance, "Improving accessibility to statistical graphs: the iGraph-Lite system", Proceedings of the 9th international ACM SIGACCESS Conference on Computers and Accessibility, New York, NY, USA, October 2007, pp. 67-74.

[11] M. Savva, N. Kong, A. Chhajta, L. Fei-Fei, M. Agrawala, J. Heer., "ReVision: Automated classification, analysis and redesign of chart images", Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology, Santa Barbara, CA, USA, October 2011, pp. 393-402.

[12] J. Poco, J. Heer, "Reverse-Engineering Visualizations: Recovering Visual Encodings from Chart Images", Computer Graphics Forum, Vol. 36, No. 3, 2017, pp. 353-363.

[13] L. Battle, P. Duan, Z. Miranda, D. Mukusheva, R. Chang, M. Stonebraker. "Beagle: Automated Extraction and Interpretation of Visualizations from the Web", arXiv, No. 1711.05962, 2017.

[14] I. Kavasidis, S. Palazzo, C. Spampinato, C. Pino, D. Giordano, D. Giuffrida, P. Messina, "A Saliency-based Convolutional Neural Network for Table and Chart Detection in Digitized Documents", arXiv, No. 1804.06236, 2018.

[15] P. Chagas, R. Akiyama, A. Meiguins, C. Santos, F. Saraiva, B. Meiguins, J. Morais, "Evaluation of Convolutional Neural Network Architectures for Chart Image Classification", Proceedings of the International Joint Conference on Neural Networks, Rio de Janeiro, Brazil, 8-13 July 2018, pp. 1-8.

[16] M. Cliche, D. Rosenberg, D. Madeka, C. Yee, "Scatteract: Automated extraction of data from scatter plots", arXiv, No. 1704.06687, 2017.

[17] K. Simonyan, A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition", Proceedings of the 3rd International Conference on Learning Representation, San Diego, CA, USA, 7-9 May 2015.

[18] A. Khan, A. Sohail, U, Zahoora, A. S. Qureshi, "A Survey of the Recent Architectures of Deep Convolutional Neural Networks", arXiv, No. 1901.06032, 2019.

[19] A. Rosebrock, "Multi-label classification with Keras", https://www.pyimagesearch.com, (accessed: 2019)

[20] TensorFlow, https://www.tensorflow.org (accessed: 2019)

[21] Keras: The Python Deep Learning library, https://keras.io (accessed: 2019)

[22] F. Bajić, J. Job, K. Nenadić, "Chart Classification Using Simplified VGG Model", Proceedings of the 26th International Conference on Systems, Signals and Image Processing, Osijek, Croatia, 5-7 June 2019, pp. 229-233.

[23] A. O. Vorontsov, A. N. Averkin, "Comparison of different convolution neural network architectures for the solution of the problem of emotion recognition by facial expression", Proceedings of the 8th International Conference 'Distributed Computing and Grid-technologies in Science and Education', Dubna, Moscow region, Russian, 10-14 September 2018, pp. 342-345.