International Journal on Document Analysis and Recognition (IJDAR) (2024) 27:19–44
https://doi.org/10.1007/s10032-023-00443-w

ORIGINAL PAPER

Chart classification: a survey and benchmarking of different state-of-the-art methods

Jennil Thiyam[1] · Sanasam Ranbir Singh[1] · Prabin Kumar Bora[1]

Received: 5 June 2022 / Revised: 5 May 2023 / Accepted: 19 May 2023 / Published online: 20 June 2023

Abstract
With the increase in the number of documents with various types of charts available on the internet, automatic chart classification has become an essential task for various downstream applications such as chart data recovery, chart replenishment. This paper presents a comprehensive survey of the studies reported in the literature since 2001 from the perspective of the corpus, pre-processing techniques, feature extraction, and methodologies. Considering that the majority of the existing studies use small datasets with a smaller number of chart types and also reported varying performances, this paper implements and evaluates 44 different machine learning-based chart classification models. The evaluation is done over a large dataset curated locally and benchmarks the performances of these 44 different models over a common experimental framework. It also performs a comprehensive error analysis, identifying two core challenging issues (noise in the charts and confusing chart pairs) that affect the chart classification performances. Compared with the existing survey papers, this paper presents a more comprehensive review and experimental analysis.

Keywords Chart survey · Chart image classification · Chart dataset · Chart classification error analysis · Chart's noise · Confusing chart pairs

1 Introduction

Data visualizations using charts are quite common in our everyday lives and are regularly available in newspapers, journals, online sites, books, etc. A well-designed chart can usually provide an intuitive comprehension of its underlying data in an effective way. Charts are powerful summarization tools that allow researchers to quickly navigate through results and comprehend them. As the number of scientific documents with various forms of charts available on the internet increases, automatic chart classification is becoming an essential task for multiple applications (such as information mining, redesigning) to better understand intrinsic information present in the documents.

Though studies on chart analysis can be traced back to the year 1991 [4], studies on chart classification have mostly started from the year 2001 [5]. The process of automatic chart classification generally consists of three steps; *corpus creation*, *pre-processing*, and *classification*, as shown in Fig. 1. In the literature, two types of corpus are generally used for chart classification; *real chart images* curated from the internet, journals, documents, etc., and images generated from *synthetic data*. Though generating a large synthetic dataset is inexpensive (sample size 202k in [6] and 144k in [7]), such datasets may not capture variational characteristics present in real charts. While a real dataset captures the actual variational characteristics of charts, creating a large-scale real chart dataset is an expensive operation. The pre-processing step is optional, depending on the nature of the samples and classifiers used. The pre-processing step generally refines the representation of the samples by employing the methods such as dimension reduction, noise removal, contrast altering, contrast pixel recoloring. Considering the recent trends, the classifiers used for the chart classification can be broadly grouped into *pre-deep learning* and *deep learning* era. The pre-deep learning era generally focuses on hand-crafted features and traditional classifiers such as SVM, KNN, Decision Tree, Bayesian. The deep learning era exploits methods like CNN, and its variants (VGGs, Xception, ResNets, Inceptions, MobileNet), Deep belief networks (DBN), etc. Though several studies on chart classification have been reported in the past, variations in the perfor-

✉ Jennil Thiyam
jenni176155101@iitg.ac.in

[1] IIT Guwahati, Guwahati, India

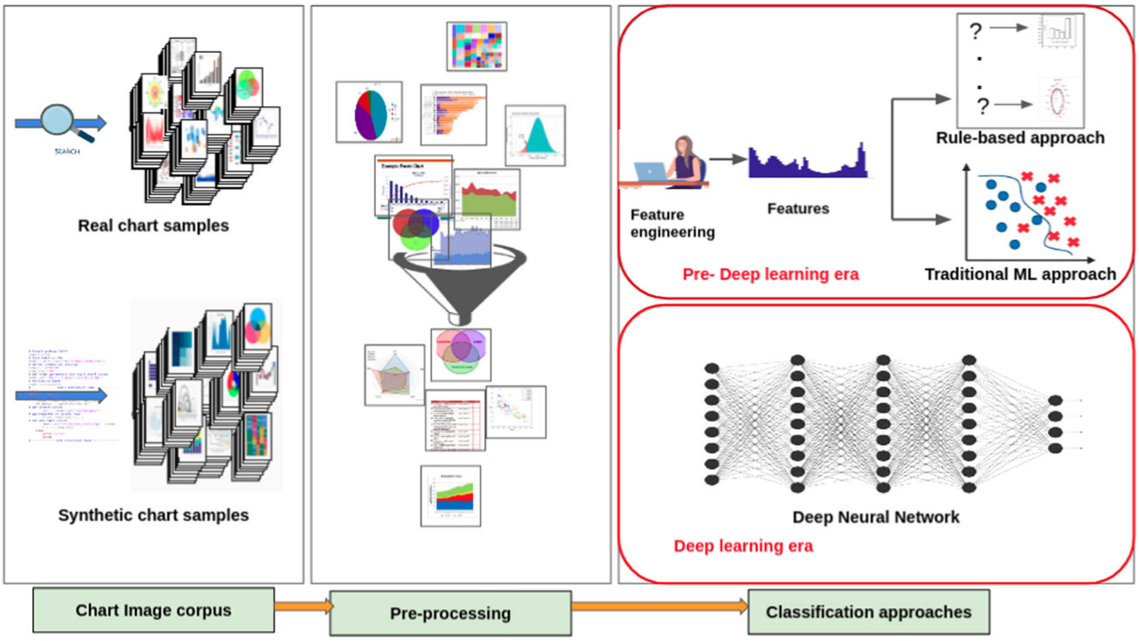


Fig. 1 General process for developing a chart classifier

Table 1 Comparison of three existing survey papers with ours with respect to the areas in which the **reviews** are done and the **contributions** made by each study

Study	Review on						Contribution		
	Dataset	Chart type	Chart image pre-processing	Methodology			Chart image corpus	Evaluation of the state-of-art models	Error analysis
				MB	TMLB	DLB			
[1]	–	–	–	✓	✓	–	–	–	–
[2]	✓	–	✓	✓	✓	✓	–	–	–
[3]		–	–	✓	✓	✓	–	–	–
Our paper	✓	✓	✓	✓	✓	✓	✓	✓	✓

MB, TMLB and DLB represent Model-based, Traditional ML-based and DL-based approaches, respectively

mances of different classification models are evident. It may be due to differences in the characteristics of the experimental datasets (nature of the images and corpus size), differences in the number of chart types considered, and classification methods used. Some studies consider datasets as small as 129 [8], while some studies consider datasets as large as 60000 [9]. Some studies consider only two types of charts, while some studies consider 14 charts [10],[1] [11].[2] Due to a lack of coherence among the experimental setups reported in different studies, arriving at a consensus regarding the performance of different chart classification methods is a non-trivial task. To address the above issue and understand the challenges inherently present in chart classification, this paper presents a detailed survey and evaluation of 44 chart classification models under a common experimental setup.

To date, only three reviews [1–3] on chart classification have been reported in the literature. The review in [1] focuses only on classification methods of the pre-deep learning era, viz., model-based and traditional non-neural-based classifiers. The other review in [2] presents a more profound analysis as compared to the prior. It provides a survey from three different perspectives: *image pre-processing methods*, *methodology*, and *chart image corpus*. Unlike the study in [1], it covers classification methods from both eras. Study in [3] focuses mainly on reviewing the studies for decoding the visual data from chart images. It examines over

[1] Though the authors in [10] have considered 28 classes, only 14 of them are the chart types and the remaining classes include non-chart classes such as medical images, natural images. Visualizations that are considered as chart type in this study are described in Sect. 2.1.

[2] Study in [11] provides 15 classes, but one of the classes is the combination of all non-visualization images such as camera-clicked pictures, conceptual diagrams

Table 2 Summary of existing real chart datasets

Study	Search Tool	Publications	Size	#class	Avg#
[5]	Google		840	3	280
[16]		ASM	129	2	64
[8]		BioMed Central	2800	5	560
[17]	Google		653	5	127
[18]	Google		200	4	50
[19]	Google		210	5	42
[20]#	Google		2500	10	250
[21]	Google		300	3	100
[22]	Google		155	4	38
[23]		Arxiv	1500	5	300
[24]	Google		5000	5	1000
[25]		Biodiversity journal	3377	11	307
[26]*	Google		6997	10	699
[27]*#	Google		4837	10	483
[9]		CVPR, ICML, ACL, CHI, AAI	60000	7	8571
[28]		ACL, Quartz	807	5	161
[14]	Google, Baidu Yahoo, BIng AOL, Sogou		11174	5	2234
[29]*	Google		2702	10	270
[6]#		PubMed Central	4242	7	606
[30]# @		PubMed Central	22923	15	1100
[10]		CVPR, ECCV, ICCV	33000	28	1178
[31]	Google		2500	2	1250
[32]*	Google		2702	10	270
[33]	Google, Baidu, Yahoo, Bing, AOL, Sogou		21099	13	1623
[34]	Google		3002	10	300
[35]*	Google		1400	7	200
[36]	Google		8000	10	800
[15]	Wikipedia AI2D dataset, SlideWiki		3629	9	403
In-house	Google, Bing, Yahoo	CVP, ICML, ACL,CHI AAAI, ACM	117271	28	4200

Avg# denotes an average number of samples per class

* Study that adopts the filtering criteria mentioned in Sect. 2

# Study that made their dataset available for public use

@ [30] is the superset of dataset provided by study [6] few existing datasets but is not as deep from the perspective of chart categories. Our survey, on the other hand, offers a review for every module of the pipeline shown in Fig. 1. Table 1 shows the comparison of the three current survey papers with ours concerning the areas in which the reviews are done and the contributions made by each study. From the table, it is clear that our paper, as compared to the three existing studies, presents a detailed literature review and benchmarking of various machine learning (ML) models. In summary, our study contributes to the following points:

1. *Survey:* Review of the findings reported in different studies.
2. *Benchmarking:* Empirical evaluations of 44 machine learning models (9 traditional ML models and 35 deep learning (DL) models) are performed to benchmark the performance over a *common evaluation setup*. To the best of our knowledge, 15 (EfficientNet's eight variants, EfficientNet V2's seven variants, NASNetLarge, and NASNetMobile) of them have never been used in the domain of chart type classification.
3. *Dataset:* Our earlier dataset published in [12] is extended with extra three more chart types. The extended dataset consists of 117271 samples over 28 chart types. To the best of our knowledge, this is the largest real chart dataset consisting of the highest number of chart types in chart classification.
4. *Stability of existing state-of-art models:* Performance of a classification model may be affected by the training sample size. Some models might give the best performance with a small training dataset, and some might not, and vice-versa. To study this situation, the influence of different training sample sizes on different classification models is studied.
5. *Noise:* One of the challenging problems in chart classification is the presence of noises in charts. We identify different noises generally present in charts and investigate their effects on chart classification.
6. *Confusing chart pairs:* Charts of similar characteristics is also one of the significant reason for chart misclassification. We identify confusing chart class pairs and their causal effects to understand their influences.

The rest of the paper is organized as follows. Section 2 presents our discussion on the existing methods used for the creation of chart image corpus. It is further extended with the details of the existing dataset, and chart types that have been used. Different methods of image pre-processing are presented in Sect. 3. Section 4 presents different methodologies from two eras: the pre-deep learning era (rule-based and traditional ML approach) and the deep learning era (deep learning approach). Section 5 presents the evaluation of 44 ML models over two common setups to benchmark different chart classification methods. Two significant challenges in developing an automatic chart classification model: *Noise*, and *Confusing chart class pairs*, and their effects are discussed in Sect. 6. Section 7 concludes our study and highlights future directions.

Table 3 Summary of existing synthetic chart datasets

Dataset	Generation tool	Size	# class	Avg #
[37]	XML/SWF	980	5	196
[28]	Vega	4318	4	1079
[38]	Matplotlib	10000	5	2000
[7]	Vega	14471	10	1447
[6]	Matplotlib	202550	10	20255
[36]	Not-mentioned	10000	10	100
[39]	Plotly	120000	20	6000
[30]	Matplotlib	17399	12	1000

Avg# denotes the average number of samples per class

2 Chart images corpus creation

As mentioned in the previous section, two types of the corpus are generally used: *Real chart corpus* and *Synthetic chart corpus*. Because of the involvement of an expensive and time-consuming effort in creating large annotated datasets of real chart images, most of the studies on chart classification with real chart images consider small datasets. Existing synthetic chart corpora are large as compared to real chart corpora. This section briefly reviews the real and synthetic chart datasets and different chart types used in the earlier studies

2.1 Real chart image corpora

For such corpora, the samples are generally collected from different open internet sources. Typically, two approaches are generally used: (i) search by the name of the chart as a keyword on image search engines like Google, Baidu, Yahoo, Bing, Aol, Sogou, etc., and (ii) extract charts from scientific documents using the tool like PDFfigure [13]. Some of the scientific publications considered for extracting chart images in the literature are CVP, ICML, ACL, CHI, AAAI, ECCV, ICCV, PubMedCentral, CVPR, BioMed(BMC), ASM. Once the images are extracted/collected (applicable for both search engine-based and extraction from publications), the images are generally further annotated by human subjects. Some studies also apply certain filtering criteria, as given below, to remove noisy images.

– More than one data visualization in the same image.
– Partially showing chart plot.

Table 4 Summary of the chart types considered in various existing studies

Sl no.	Chart type	Study	# study
1	Arc	[7, 12, 33] [8, 20, 25, 37]	3
2	Area	[7, 26–29] [10, 12, 32–34] [5, 16–19, 36] [41]	21
3	Bar	[20–22, 24, 26] [7, 9, 27, 28, 38] [6, 10, 14, 29, 31] [36, 41] [12, 32–35]	22
4	Box	[6, 10, 12, 25]	4
5	Block	[10]	1
6	Bubble	[10, 12, 36]	3
7	Column	[8, 12, 25, 37] [6, 10, 36, 38]	8
8	Dendrogram	[12, 25]	2
9	Doughnut	[36]	1
10	Flowchart	[8–10, 12, 24]	5
11	Gantt	[12]	1
12	Heatmap	[10, 12, 25]	3
13	High–Low	[5]	1
14	Line	[5, 8, 16–18] [19–22, 24] [9, 25–28] [6, 7, 10, 14, 29] [12, 32–34, 36] [41]	25
15	Manhattan	[12]	1
16	Node	[12, 25]	2
17	Parallel	[10, 12, 38]	3
18	Pareto	[8, 12, 26–28] [10, 29, 32, 34, 41] [5, 17–20]	10
19	Pie	[12, 21, 22, 24, 25, 37] [7, 26–28, 34, 38] [36, 41] [6, 10, 14, 29, 32, 33]	25
20	Radar	[14, 20, 26–28] [10, 12, 29, 32, 34] [41]	11
21	Reorderable	[7, 12, 33] [8, 17, 20, 24, 25]	3
22	Scatter	[7, 9, 26–28] [6, 10, 14, 29, 31] [12, 32–34, 36] [41]	21
23	Sunburst	[6, 7, 12, 19, 22, 33]	6
24	Surface	[10, 12, 17]	3
25	Table	[9, 20, 26–28] [10, 12, 29, 32, 34] [41]	11
26	Treemap	[7, 12]	2
27	Venn	[14, 20, 26–28] [10, 12, 29, 32, 34] [41]	11
28	Waterfall	[12]	1

- Images with watermarks.
- Image with resolution below 500 × 500.
- Images with transparent background.
- 3D chart images.
- Cluttered chart images.
- Hand-drawn chart images.
- Ignore table charts that are in the form of “fill up the forms.”

Characteristics of datasets reported in earlier studies are summarized in Table 2. The majority of them are relatively small in size with fewer chart types. To date, as compared to all existing studies, the study [9] considered a higher average number of samples per class, i.e., 8571 (for the classification of five chart types). Except for the paper in [14] average sample size (2241), the rest of the papers have considered an average sample size of less than 2000. Slightly different from the rest of the studies, the study [15] collected chart samples (of nine chart types) from Wikipedia, AllenAI Diagram Understanding (AI2D), and educational slides from SlideWiki[3] which is an open educational resource platform.

2.2 Synthetic chart image corpora

In this approach of chart corpus generation, tools like MATLAB, the python library Matplotlib, XML/SWF, and Vega (and its associated tools) are commonly used in the literature for generating synthetic chart images. A study in [28] has reported that some of the images synthesized with Vega are found in the Google search. Therefore, the dataset in this study may be considered a combination of synthetic and real chart images. Study in [6] utilizes Matplotlib to generate chart samples whose underlying data points are taken from real statistics which are available online such as world development indicators, gender statistics, the government of

[3] (https://slidewiki.org/).

India open data, commodity trade statistics, US census data, and price-volume data for stocks and EFTs. Unlike the above datasets, the dataset reported in the study [38] generates its dataset by randomizing the plot aesthetics as given in the study [40]. However, study [37] developed its dataset with the chart images generated using the XML/SWF chart tool (data for the charts were generated randomly). Recently, study [39] developed a huge synthetic dataset entirely made with Plotly. Table 3 presents the characteristics of the existing synthetic chart datasets. From the table, it can be seen that to date study [6] has provided the largest synthetic chart dataset of size 202,550, consisting of 10 chart types (20,255 samples/chart type).

2.3 Chart types

The visualizations considered in the literature under the "Chart type" differ for various studies. Some studies consider Map as one of the chart types, while others do not. This study considers technical data visualization (such as Area, Bar, etc.), data flow presentation (such as Flowchart and Block diagram), and Data tables as the chart types. Table 4 summarizes the types of charts considering different studies. All of the datasets considered in the earlier studies except for our earlier study in [12] consist of chart types < 15. Area, Bar, Line, Pie, and Scatter are commonly considered chart types. As mentioned earlier, this study added three more chart types to our existing dataset presented in [12]. Our latest dataset consists of 28 chart types, and except for the surface plot, the remaining 27 are in 2D. Another point to be re-noted is that even though the study in [10] considered 28 different classes, only 14 of them are the chart types. In the same manner, study [11] used one non-chart class making 14 out of 15 classes a chart type.

3 Chart image pre-processing

Image pre-processing is the lowest level of abstraction for image operations. Its objective is to improve image information, which removes unwanted distortions or improves some relevant image characteristics for further processing and analysis. This section presents a discussion of different image pre-processing methods exploited by the various existing chart classification studies.

The basic image pre-processing applied in most of the studies is to resize the image, preserving aspect ratio [19, 28]. Depending on the further steps, RGB images are converted to grayscale images [18, 19, 32] or binary images [28]. In the various studies, averaging of the channels is considered for converting to grayscale and the threshold approach (using Otsu's method [42]) is considered for converting to a binary image. With the image in its grayscale form, various studies perform connected component generations [8] and connected component analysis [19, 43], to separate the graphic's region from the text regions. The separation is done considering a series of filter that takes into consideration of the size, height and width, aspect ratio, and black pixel density of each connected component. Further, the classification can be extended to the removal of the noise components as most of the noise components are small in size [5]. This traditional connected component analysis is reported to be time-consuming [44]. The authors in [45] modify the connected component analysis to reduce the computational time by using the neighbor-scan labeling approach.

Based on the types of noise expected from the source and the nature of images, various studies used noise removal techniques such as Gaussian smoothing, median filters, bilateral filters, and morphological operations. To vectorize over image edges, which in the future can be used as the features of the images, edge mapping is one of the pre-processing methods implemented by some studies [5, 8, 18, 46] using edge detection algorithms such as canny edge detection or simply calculating the intensity differences between the neighboring pixels. With the assumption that chart type is often part of the image itself, some studies used OCR to extract the text area and compare the OCR results with the classification model's result [17]. The effectiveness of applying the image pre-processing step in chart classification is discussed in the study [32]. It has reported an improvement of 9% in accuracy after applying a series of image pre-processing methods such as resizing, smoothing, sharpening, and contrast enhancement.

4 Chart classification methodologies

This section briefly reviews different methods reported in earlier studies on chart classification. Considering the present trend, the discussion has been presented as *pre-deep learning era* and *deep learning era*.

4.1 Pre-deep learning era

As shown in Fig. 1, all the existing studies in this era have adopted either a rule-based approach or traditional classifications using SVM, Bayesian, KNN, decision trees, etc. These methods generally follow the two-tier process. First, feature engineering is applied to extract features from the images. Second, the classifiers are built using the extracted features. The rule-based and traditional ML approaches used in earlier studies are briefly discussed below.

Table 5 Few selected studies published under pre-deep learning era which used traditional ML classification approach

Paper	Year	Feature descriptor	Classifier	#classes	Dataset Overlap
[8]	2006	Grapheme statistics	MLP	5	–
[17]	2007	Region segmentation features, Edge continuity features, The Scale Invariant Feature Transform (SIFT)	SVM	5	–
[20]	2011	Image patches codebook	SVM	10	–
[21]	2012	Graphical components	SVM MLP	3	–
[22]	2012	Gray Level Co-Occurrence Matrix(GLCM)	SVM KNN	5	–
[23]	2015	Visual—color, edge, lines; HOG, LBP	SVM KNN	5	–
[24]	2015	Local Binary Pattern(LBP) Histogram of Oriented Gradients (HOG)	SVM	5	–
[7]	2018	Histogram of Oriented Gradients (HOG)	Naive Bayes SVM KNN Random forest	10	–
[36]	2021	Heterogeneity index (HI) fused with local penta pattern fused with local penta pattern	SVM	10	–

Dataset overlap denotes the study that uses same dataset (as the reference study) for training or evaluation process

4.1.1 Rule-based classification approach

In a rule-based approach, a model is generally built using a sequence of hierarchical rules derived from domain knowledge. Given the basic image features like a set of edges, edge map, location of the connected components, etc., various studies developed the rules to identify higher-level abstractions such as line segment, circular arc, elliptic arc, data points, tick. Using these segment information, rules to identify further higher-level abstractions such as line chart, pie chart, bar chart are formed. Based on the nature of the chart types, different levels of rules may be applied. At the time of classification, the extracted features and the layouts are compared against each chart model to determine the likelihood that the input sample belongs to the target class. Each chart type has some individual features that help to calculate the likelihood value. The study in [18] uses two levels of domain knowledge to identify graphical symbols representing the data for each chart type. At the top level, it specifies the kind of graphical symbols that are expected to appear in each chart type. At the bottom level, the set of graphical primitives and their constraints are further determined to form the graphical symbol. For instance, to identify the graphical symbols of a pie chart, it should have more than one object called ‘pie,’ which can be further defined by the occurrences of the tuple (l_1, l_2, a_1) (where l_i and a_i refer to a line and an arc, respectively), and constraints like l_1 and l_2 share a common endpoint, one endpoint of l_1 lies on the shape a_1, one endpoint of l_2 lies on the shape a_1 and a_1, etc.. The study in [19] uses multiple instance learning by treating a chart image as a bag containing a set of instances (graphical symbols). It adopts the idea of the correlation factor of each shape for each chart type. Studies in [37, 46] focus on learning characteristics of the edges of different chart types to capture the underlying shapes and spatial layouts. One of the main disadvantages of the rule-based approaches is that they are sensitive to small variations in the chart. Even a slight change in the chart style may affect the rules [18]. Further, different rules need to be created for different types of charts, and it may not be scalable or difficult to obtain the rules for complex charts. As a result, to date, studies using rule-based approaches consider a lesser number of chart types, up to four chart types [46].

4.1.2 Traditional ML classification approach

The disadvantages of the rule-based approach are overcome by traditional ML approaches like SVM, Decision tree, etc., by considering graphical elements of charts from the whole image instead of different segments. However, they rely on a vast set of training samples. Table 5 presents some of the selected studies that use traditional ML classification methods. From the table, it is observed that various studies pair up different feature descriptors with different classification models.

Table 6 Few selected studies published under deep learning era

Paper	Year	Le	Alex	VGG		ResNet			Inception		Mobile	Xcep	Customized	#class	Dataset
		Net	Net	16	19	50	101	152	v3	v4	Net	tion	CNN		Overlap
[26]	2017	✓	✓						✓					10	[20]
[27]	2017	✓												10	–
[9]	2017		✓			✓								7	–
[28]	2017		✓											5	[20, 26]
[25]	2017												✓	11	–
[38]	2018										✓			5	–
[7]	2018				✓	✓			✓					10	–
[14]	2018		✓	✓		✓			✓					5	–
[29]	2019			✓										10	[20]
[6]	2019			✓		✓	✓							10	–
[10][1]	2019			✓										28 (14 chart types)	–
[31]	2019			✓										2	–
[33]	2019				✓			✓			✓	✓		13	–
[32]	2020			✓										10	[20]
[11][2]	2020										✓			15 (14 chart types)	–
[47]	2020												✓	2	–
[35]	2021												✓	2	–
[34]	2021												✓	10	[6, 20]
[41]	2021												✓	10	[9, 20, 27, 40]

Dataset overlap denotes the study that uses the same dataset (as the reference study) for the training or evaluation process

Various types of features have been used, ranging from symbols to visuals to textures. There are studies that extract basic symbols, such as axis coordinates [43], bar [18, 23], histograms [48], pie [18, 19], lines [16, 49], curves [5]. Study [8] uses the count of 16 graphemes, such as vertical tick, horizontal tick, line segment, curve, branch, as features to represent chart images and use MLP for the classification. Some widely used texture descriptors are Local Binary Pattern (LBP) [23, 24], Histogram of Oriented Gradients (HOG) [7, 23, 24], and scale-invariant transform (SIFT) [17, 24]. The study in [17] proposes a classifier based on the shape and spatial relationship of their primitives, such as the structure of salient regions present in the image, the local shape of salient curves, etc. Their extracted features are based on region segmentation, curve saliency, HOG, and SHIFT. The study in [20] is the first one to consider ten chart types and makes the dataset available for public use. In their study, image patches are provided as the features. It uses visual dictionaries made up of repeating image patches or features to learn embedding image representations (visual words). A predetermined number of visual words are sampled densely or pseudo-randomly during training by eliminating low-variance patches. After that, a visual dictionary is created by grouping the visual words and textures, such as lines, points, curves, arcs, and gradients, using k-mean clustering. Their study uses multi-class SVM for the classification task. The study in [22] evaluates three classification models, viz. SVM, KNN, and MLP, with gray-level co-occurrence matrices (GLCM). The GLCM is extracted fundamental statistics parameters like entropy, energy, homogeneity, contrast, and angular second moment as the features. The study in [36] addresses the challenges of micro-variabilities features. It is the one feature that makes each image type different from others, even though chart image belongs to the same class or categories. It proposed a heterogeneous feature descriptor, namely the heterogeneity index (HI), which was built based on the similarity of chroma effects and its intensity in the microstructures. Their study fused HI with local penta pattern which is a color texture descriptor using five code patterns. It is one of the good studies that perform a comparison of different traditional feature descriptors (viz. HOG, SIFT, LBP, GIST) over the classification of chart types.

From Table 5, it is observed that all the studies or works done during the pre-deep learning era utilize their own dataset, and there is no cross-check of their proposed methods over the datasets collected or used by other studies. It might be because of the unavailability of the dataset for public use (it can be seen from Table 2 that the study in [20] is the only one that made the dataset available for public use during the era of pre-deep learning).

4.2 Deep leaning era

In this era, numerous studies exploited several deep learning methods. The deep learning methods overcome the need of feature engineering as it automatically learns feature representation from the training dataset directly. Numerous studies built chart classification models considering various existing CNN architectures such as LeNet [50], AlexNet [51], VGG - 16,19 [52], Inception - V3 [53], Inception - V4 [54], ResNet - 50,101,152 [55], Xception [56], Inception - ResNet-V2 [54], MobileNet [57], DenseNet [58]. Some studies utilize both traditional and deep learning approaches to develop classification models such as [10] use pre-trained VGG-16 as the feature extractor and fed to the classifier SVMs, the study in [59] combined ANN and SVM for the classification of one chart type with different characteristics such as scatter chart(with and without line), 2D and 3D pie charts. Even though it is observed from Table 5 that the majority of the studies used heuristic features with the traditional classification model, the study in [23] used heuristic features as the input for deep belief network (DBN) [60] for the classification of five classes.

In another scenario, some studies developed customized CNN based on the architecture of well-known models. The study in [25] developed a CNN model inspired from the LeNet architecture [50]. Their study presented the results provided by the two orientations of networks, pre-trained on the subset of the large dataset ImageNet[4] and non-pre-trained. In their case, the non-pre-trained network provides better results than the pre-trained network. This could be because of the significant differences between natural images contained in the ImageNet dataset and the chart images in their dataset, which lead to the low performance of the model. The study in [24] developed a model considering CNN and DBN. It used CNN for the feature extraction, and DBN was engaged for dimension reduction with its softmax layer as the prediction layer. Like most of the studies, because of the lack of training examples of charts, their network was trained on the dataset ImageNet ISVCR [61], which contains 1000 categories and 1.2 million images. The study in [34] introduced Siamese CNN in the field of chart classification. Their study considered their earlier network, Simplified VGG [29], SigNet [62], and Omniglot [63], and performed the comparison in their Siamese network form. With the assumption that DenseNet provides access to gradients from the loss function as well as the input layer, the study in [41] developed a chart classification model based on the architecture of DenseNet.

A summary of the above discussion is presented in Table 6. Among these networks, most of the studies adopted VGG-16, followed by AlexNet and ResNet-50. From the table, it is observed that multiple studies, such as studies in [26, 28], etc., utilize various publicly available datasets for the development and validation of their proposed methods. Among the publicly available datasets, the one provided by the study in [20] is commonly used and acts as a benchmarking dataset. It is because of the chart samples it has, which are very noisy (discussed in detail in Section 6.2), and they provide challenging factors to the proposed methods.

5 Benchmarking of chart classification methods

In the above sections, we have presented a brief review of different approaches of chart classification reported in earlier studies since 2001 (discussed in Sect. 4.1.1, and shown in Tables 5 and 6). The following observations can be noted.

1. **Dataset Types:** Two types of datasets have been commonly used; real dataset—chart images extracted from real documents, and synthetic dataset—created synthetically using mathematical models.
2. **Small dataset size:** Except for the synthetically generated dataset, most of the existing studies consider small datasets.
3. **Small number of chart types:** All the existing studies consider a small number of chart types—14 being the highest and five being common.
4. **Differences in model performance:** Differences in performance between different models are reported in different studies. For example, the study in [26] observes that the deep learning-based model, Inception, performs better than AlexNet, whereas study [14] reports the opposite. Another example is that the study in [7] presents a traditional classifier, SVM, which performs better than KNN, while the opposite is reported in the study in [22].

Motivated by the above observations, in this paper, as mentioned in Sect. 1, we build 44 different ML models (9 traditional and 35 deep learning) and evaluate them over a common setup to benchmark their performances and to identify their limitations, and understand the inherent challenges in chart classification. None of the existing surveys evaluate the methods over the common frameworks (except our earlier study [64] and [12]). The remaining part of the paper focuses on the following points:

[4] https://image-net.org/.

- **Common Experimental Setup:** We rebuild all the 44 classification models. The models are evaluated using the following two experimental setups.
  1. *Fivefold cross-validations:* The performances of 44 ML models are compared under random fivefold validation of a locally curated dataset. The models are tested over three publicly available datasets [20, 27, 30].
  2. *With different training sizes:* Different studies have considered datasets of different sizes, and inconsistencies in the reported performances have been observed as shown in Tables 5 and 6. To understand the sensitivity of the models to different training sample sizes, we build the models with different training sample sizes and evaluate their performances.
- **Noisy chart type:** As charts are created for different needs, they come with different types of noises, which may hinder the classification performance. We identify different types of noise present in the chart and study their effects.
- **Confusing chart pairs:** A classifier may confuse charts of different characteristics belonging to different classes or chart types. We identify charts of confusing nature and their effects.
- **Potential future directions:** For building an effective chart classifier, we would need to address the problem of noise and confusing chart class pairs. We discuss the potential future approaches to handle the above challenges.

Table 7 Comparison of four datasets

	Dataset				
Chart type	D1	D2	D3	In-house	
				P	**SE**
Arc	–	–	–	936	1105
Area	39	246	172	2401	4321
Bar	51	346	787	1251	3757
Box	–	–	763	762	1427
Block	–	–		1872	1140
Bubble	–	–	–	1590	1624
Column	65	340	5454	2011	3188
Dendrogram	–	–	–	747	2274
Doughnut	–	–	–	120	1947
Flowchart	–	–	–	891	3210
Gantt	–	–	–	56	2103
Heatmap	–	–	197	650	2569
High–low	–	–	–	628	1382
Line	85	488	10556	2763	5128
Manhattan	–	–	176	792	1899
Node	–	–	–	1733	4200
Parallel	–	–	–	389	1713
Pareto	63	262	–	738	4683
Pie	27	365	242	2605	2614
Radar	50	50	–	2101	3173
Reorderable	–	–	–	386	1618
Scatter	90	556	3168	2841	5890
Sunburst	–	–		1002	2465
Surface	–	–	155	3002	3440
Table	42	267	–	4034	2507
Treemap	–	–	–	261	2927
Venn	44	343	75	2011	4199
Waterfall	–	–	–	46	2149
Total sample	320	3867	21745	38619	78652

P and SE denote two different sources, viz. Publications and Search Engine, respectively

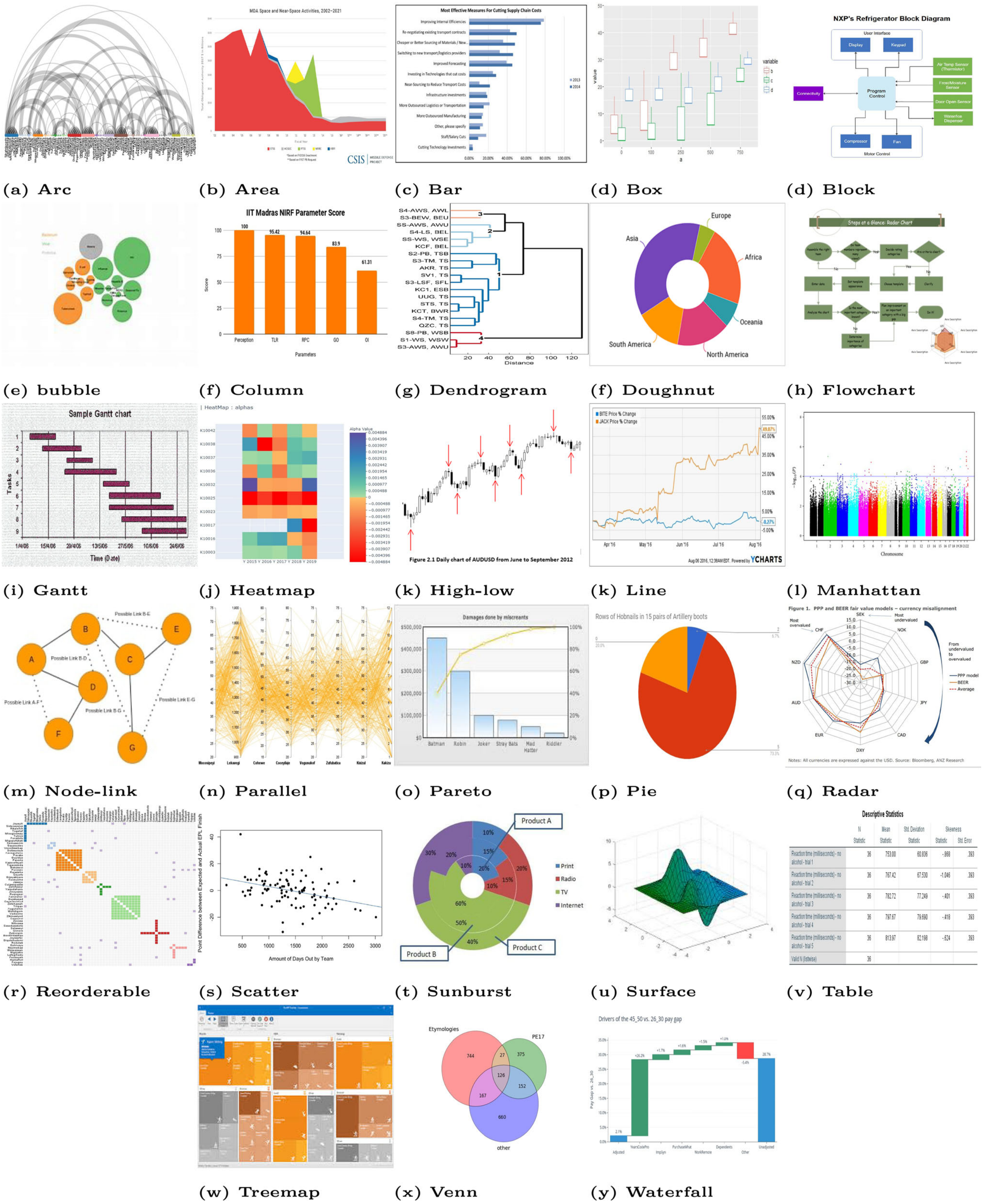
5.1 Experimental setups

5.1.1 In-house dataset

As mentioned above, most of the earlier studies considered small datasets with a small number of chart types. It may be noted that several new chart types have been used in recent scientific documents. Many of these chart types have not been considered yet. Motivated by this, we have created a large annotated dataset of 110,182 samples over 25 chart types in our previous work [12]. In this previous work, we have come up with 25 chart types considering: (i) all the data visualizations mentioned in the various existing state-of-the-art chart classification studies and (ii) Google's responses on searching *chart types* as a keyword. With the above two ways of searching different chart types, we considered only those types in which we can collect at least 1500 samples.[5] So we ignore some chart types that we found in existing studies and Google searches, such as Interval, Doughnut, high–low (because we failed to collect at least 1500 samples for each of them).

With time and following the same procedure in our earlier work (considering both ways of collection, i.e., using the search engine Google, and papers published under CVP, ICML2, ACL, CHI, AAAI, and ACM), we are able to collect at least 1500 samples for three chart types we ignored during the publication of our earlier work: *Block, Doughnut, and High–low chart*. Our extended in-house dataset consists of 117,271 samples over 28 chart types. Figure 2 shows an example of each of the 28 chart types. Our dataset is the largest real chart image dataset with the highest number of chart types, to the best of our knowledge. We plan to make this extended final dataset public.

[5] To keep the balance between the sample size, we aim to set the minimum sample size of every chart type as 1500.



(a) Arc (b) Area (c) Bar (d) Box (d) Block

(e) bubble (f) Column (g) Dendrogram (f) Doughnut (h) Flowchart

(i) Gantt (j) Heatmap (k) High-low (k) Line (l) Manhattan

(m) Node-link (n) Parallel (o) Pareto (p) Pie (q) Radar

(r) Reorderable (s) Scatter (t) Sunburst (u) Surface (v) Table

(w) Treemap (x) Venn (y) Waterfall

Fig. 2 Example of 28 chart type

Table 8 Mean accuracy and standard deviation (of the five accuracy scores) for 9 traditional ML-based chart classifiers under fivefold cross-validation of in-house dataset

Model	Training	Testing			
		D1	D2	D3	In-house
HOG + SVM	**88.67**	**74.20**	**79.49**	**80.75**	**85.06**
	±0.56	±0.14	±0.15	±0.31	±0.31
LBP + SVM	82.67	68.45	81.06	80.16	80.95
	±0.78	±0.50	±1.21	±0.71	±0.87
GLCM + SVM	83.01	64.08	73.22	80.02	82.39
	±0.78	±0.32	±0.97	±0.45	±0.54
HOG + KNN	82.78	70.91	78.31	79.21	80.84
	±0.69	±0.38	±0.82	±0.76	±0.97
LBP + KNN	83.04	73.05	77.55	79.99	81.80
	±0.92	±0.17	±1.31	±0.91	±0.87
GLCM + KNN	82.78	77.17	77.55	80.54	81.19
	±0.78	±0.78	±0.64	±0.57	±0.71
HOG + RF	82.87	72.81	79.49	79.21	82.76
	±0.82	±0.52	±0.91	±0.84	±0.82
LBP + RF	83.78	67.66	73.60	78.72	82.67
	±0.49	±0.83	±0.54	±0.84	±0.55
GLCM + RF	83.06	69.40	73.64	79.06	82.14
	±0.87	±0.76	±0.67	±0.91	±0.79

The performance is reported for both training and testing. Bold entries represent the highest accuracy score

On top of our dataset, as mentioned earlier, this study considers another three publicly available real chart datasets for inference. They are provided by the studies [20, 27] and [30].[6] For the rest of this paper, these three datasets have been designated as D1, D2, and D3, respectively. It is important to note that some of the samples in D1 and D2 were found to be non-functional due to a lack of maintenance in their repositories. Some of the links provided for D1 resulted in errors or pointed to irrelevant images, while some of the files in D2 were also found faulty when opened. As a result, these datasets were carefully examined, and only functional samples were included in the study. On the other hand, the third dataset D3 is free from those errors. Even though our dataset, in-house, considers a larger number of chart types, D3 comprises two chart types that are not considered in our dataset, namely Interval (both horizontal and vertical), and Map. Therefore, D3 is not equal to the dataset provided by study [30] but a subset of it which have all the samples of all the chart types except for the aforementioned two types. Table 7 compares four datasets.

5.1.2 Setting up the classifiers

In the pre-deep learning era, SVM, KNN, and Random forest are commonly used classifiers in chart classification (see Table 5). Further, three methods of feature extraction, namely Histogram of Oriented Gradients (HOGs), Local Binary Pattern (LBP), and Gray Level Co-Occurrence Matrix (GLCM), are commonly used in chart classification [7, 23, 65–68] as well as object detection [69, 70]. Considering this, we evaluate the performance of three classifiers (SVM, KNN, and Random forest) with three different feature extraction methods (HOG, LBP, and GLCM). We briefly describe the experimental setups below.

- **Feature Extraction:**
  - *Histogram of Oriented Gradients (HOG):* HOG represents an object using the local distribution of intensity gradients and edge directions. It is the normalized histogram of image gradients with respect to various orientations collected within localized regions in the image. We segment the images into local patches and extract the features from the patches. We have experimented with various HOG cell sizes of 16 × 16, 28 × 28, 32 × 32, and 64 × 64. Though study [7] recommends using 28 × 28 cell size, an empirical study on our dataset observes that the cell size of 16 x 16 provides the best performance. As a result, HOG-related experiments in this paper use cell size of 16 × 16.
  - *Local Binary Pattern (LBP):* In general, LBP encodes local pixel neighborhoods using binary representation. First, we segment the original image resolution 256×256 into *16* × *16* patches with size $((256/16) \times (256/16))$. Then, in each patch, the LBP feature with radius 2 is extracted and developed a histogram.[7] Finally, a global histogram is created by concatenating histograms obtained from all the patches.
  - *Gray Level Co-Occurrence Matrix (GLCM):* Gray Level Co-Occurrence Matrix (GLCM) extracts features from an image using a co-occurrence matrix. A co-occurrence matrix of an image is a matrix representing the distribution of co-occurring values at a specific offset. The GLCM has been applied in a variety of image analysis applications [61, 71]. As in the study [22], we consider the statistical measure of an image such as area, median, minimum and maximum intensity, contrast, homogeneity, energy, entropy, mean, variance, standard deviation,

[6] The dataset provided by study [30] is the superset of the one provided by the study in [6]. Apart from increasing the number of samples to the existing dataset provided by the study in [6], the study in [30] added Heatmap, Manhattan, and Venn as new chart types.

[7] In study [24], patches of 16 and radius with 2 give the better result, and hence we follow their way.

and correlation, as our features to construct the co-occurrence matrix.

- **Classifiers:** The classifiers SVM, KNN, and Random forest (RF) are built considering features extracted using HOG, LBP, and GLCM. For KNN, we use 7 as the parameter ”k” which is the number of neighbors to be selected.[8] For SVM, the default parameters in Scikit-Learn [72] are used. For random forest, 105 random trees are considered.[9]

For the deep learning methods, we consider 35 CNN-based models: AlexNet, Inception-v4, and all 33 different CNN-based models provided by keras,[10] namely VGG -16, 19, ResNet - 50 (v1, v2), 101 (v1, v2), 152 (v1, v2), Inception -v3, Inception - ResNet, Xception, MobileNet - v1, v2, DenseNet - 121, 169, 201 [44], EfficientNet - B0, B1, B2, B3, B4, B5, B6, B7 [73], EfficientNetV2-B0, B1, B2, B3, S, M,L [74], NASNetLarge [75], and NASNetMobile [75]. Except for the EfficientNets, EfficientNetV2s, NASNetLarge, and NASNetMobile, all other models have been used in the chart classification task in the literature. To analyze the behavior of the models more deeply, this study evaluates them under two scenarios:

1. *Training the model from scratch:* In this scenario, the default models are initialized with random weights and then trained on the in-house dataset from scratch. This means that the models are not based on any pre-existing knowledge or training, but instead, they start with random weights as a blank slate.
2. *Training the model with transfer learning:* In this scenario, our study leverages the pretrained version of the model, trained on the ImageNet ISVCR dataset with 1000 classes [61]. We initialized the models with pre-trained weights and froze all layers except for the last convolutional layer (and the dense layers). As done in studies [76, 77], such a transfer learning setup allows us to leverage the powerful feature extraction capabilities of pre-trained models while fine-tuning their last layers to the specific classification task at hand. In the process, the last layer is replaced (because it is trained for the classification of 1000 classes) by another dense layer with 28 classes. The changes made in the model’s architecture are different for different models. The CNN models, such as Xception, EfficientNets, DenseNets, MobileNets, NASNetLarge, and NASNetMobile, utilize separable convolutional layers to reduce the number of parameters. In order to achieve the goal of parameter reduction in models with separable convolutional layers, we adopt the common practice of using global average pooling. This technique reshapes the output tensor from the last convolutional layer into a 1D vector, which is then fed into a dense layer. In contrast, models like AlexNet, VGGs, and ResNets employ normal convolutional layers. For these models, a simpler approach is taken where the tensor from the last convolutional layer is flattened into a 1D vector before being passed to the dense layer. By adopting the reshaping technique to match the specific architecture and objective of each model, the efficiency and effectiveness of the parameter reduction process can be maximized.

We use Stochastic Gradient Descent (SGD) as an optimizer, 0.9 as momentum, 0.0001 as learning rate, 40 as batch size, and 2 as steps-per-epoch.

5.2 Experimental observations

Table 8 shows the average accuracy over fivefold cross-validation of SVM, KNN, and Random forest with HOG, LBP, and GLCM feature extractors on the in-house dataset. It also shows that the performance of the models is observed for both validations (during training) and testing datasets (D1, D2, D3, and testing dataset of in-house). From the table, we observe the following.

- One of the important point to be considered while evaluating ML models is overfitting. We investigate the possible case of overfitting using two parameters: (i) mean accuracy: the large difference between mean training accuracy and testing accuracy may be considered as the indicator of a particular model suffering from overfitting., and (ii) standard deviation (of accuracy): large standard deviation of the performances among the fivefold (for both training and testing accuracy) can also be considered an indicator of a particular model suffering from overfitting. From the table, it can be seen that there is no drastic or major difference in the performance (mean accuracy score) of all the classifiers during training and testing (concerning in-house). In average, the difference in the accuracy score for all the classifiers while training against testing is 2% (the difference in accuracy scores of training and testing on the in-house dataset ranges from 0.11% to 3.61%). It can be further noted that the standard deviation obtained by all the classifiers in both training and testing (concerning in-house) is less than ±1.00 (ranges from ±0.11 to ±3.61). These two observations can be an indication that all the classifiers are steady and free from overfitting issues.

[8] We tune the value of K ranging from 1 to 10 and finalized to 7 as it gives the best performance.

[9] The 105 number of random tress has been decided after a brief empirical analysis of values between 50 and 120.

[10] https://keras.io/api/applications/.

Table 9 Evaluation of deep learning methods under the first scenario: performance of 35 CNN-based chart classification models under fivefold cross-validation of in-house dataset

Model			Training		Testing							
					D1		D2		D3		In-house	
			μ	σ	μ	σ	μ	σ	μ	σ	μ	σ
VGG	16		64.21	±1.27	54.41	±0.21	34.16	±0.74	49.02	±0.71	59.16	±0.35
	19		66.21	±0.72	58.45	±0.23	**63.21**	±0.41	58.92	±0.73	64.21	±0.89
ResNet	50	v1	56.73	±0.93	44.24	±0.51	40.03	±0.28	39.61	±0.74	50.01	±2.02
		v2	52.13	±0.07	41.02	±0.62	32.27	±0.25	29.05	±0.15	47.43	±0.67
	101	v1	67.12	±1.32	41.22	±0.56	43.44	±1.56	39.07	±1.51	59.32	±1.23
		v2	62.05	±1.02	41.04	±0.83	45.67	±2.76	52.03	±0.52	56.04	±0.67
	152	v1	59.36	±1.21	32.07	±1.56	41.09	±0.63	50.05	±1.18	51.09	±0.86
		v2	58.43	±0.87	39.24	±0.57	40.35	±0.78	48.03	±0.49	52.64	±0.78
Inception	v3		61.01	±0.56	54.17	±0.69	43.67	±0.69	44.22	±0.41	57.93	±0.74
	v4		67.23	±0.56	55.45	±0.6	60.06	±0.75	60.31	±0.61	63.37	±0.74
AlexNet			45.21	±0.72	34.57	±0.56	38.16	±0.62	40.01	±0.80	43.37	±0.67
Xception			65.11	±0.67	55.82	±0.73	59.67	±0.66	**59.47**	±0.55	**64.67**	±0.71
Inception -ResNet			59.12	±0.81	45.15	±0.51	50.07	±0.31	46.49	±0.22	56.79	±0.32
MobileNet	V1		62.48	±0.74	57.55	±0.72	56.59	±0.79	56.98	±0.71	58.87	±0.89
	V2		64.47	±0.74	56.14	±0.89	59.74	±1.06	57.81	±1.26	54.27	±0.63
DenseNet	121		**68.31**	±1.69	53.61	±0.56	58.12	±0.72	59.64	±0.86	62.41	±1.65
	169		61.56	±0.78	54.56	±1.35	54.07	±0.34	55.78	±0.85	59.49	±0.75
	201		63.21	±0.56	57.14	±0.33	52.19	±0.61	55.31	±0.57	60.65	±0.74
EfficientNet	B0		50.17	±3.32	42.38	±0.54	45.98	±0.68	46.53	±0.50	48.19	±0.89
	B1		51.12	±0.04	47.98	±0.71	44.94	±0.72	46.15	±0.90	49.55	±0.73
	B2		50.69	±2.14	44.17	±0.21	46.21	±0.25	47.02	±1.06	48.95	±1.67
	B3		54.74	±1.02	47.42	±0.34	47.15	±0.73	49.42	±1.47	52.19	±0.75
	B4		50.12	±1.78	81.01	±0.71	42.07	±1.67	41.94	±0.92	48.14	±0.38
	B5		52.17	±0.62	44.70	±0.29	47.71	±0.93	46.03	±0.30	49.72	±0.49
	B6		55.13	±1.26	47.16	±0.23	44.16	±0.95	45.67	±1.09	54.17	±0.58
	B7		53.21	±0.24	47.64	±1.07	47.12	±1.09	44.66	±1.53	51.56	±1.58
EfficientNet V2	B0		58.12	±1.80	47.19	±1.20	52.18	±0.49	52.04	±1.05	55.23	±1.09
	B1		54.22	±1.02	51.07	±1.09	51.04	±1.08	51.07	±1.04	52.06	±0.57
	B2		55.27	±0.34	52.10	±0.71	49.20	±0.61	49.12	±1.03	54.83	±1.08
	B3		55.51	±1.02	48.02	±1.06	51.81	±1.09	49.01	±1.02	48.23	±1.06
	S		55.12	±0.62	51.02	±0.61	51.08	±0.55	51.73	±0.36	53.34	±0.05
	M		53.07	±0.54	50.16	±1.09	51.09	±1.42	49.12	±0.54	51.02	±0.78
	L		55.78	±1.04	51.32	±1.03	50.12	±1.03	50.57	±0.48	49.25	±1.02
NASNetLarge			62.03	±1.07	57.05	±0.35	54.26	±1.02	58.73	±0.28	60.26	±0.04
NASNetMobile			61.03	±0.56	**60.15**	±0.74	58.16	±1.07	57.17	±1.82	60.02	±0.62

μ and σ denote mean accuracy and standard deviation, respectively. The performance is reported for both training and testing. Bold entries represent the highest accuracy score

- All the nine classifiers provide better classification performance when tested with the in-house dataset than the testing on the D1, D2, and D3. It may be because of the fact that the in-house dataset has a smaller percentage of noisy samples as compared to the other three. Noise analysis is discussed in Sect. 6.2.
- In most cases, while testing the new samples, the classifiers with HOG as a feature extractor outperformed the classifiers with LBP or GLCM.
- While training and testing, the classifier SVM with HOG, i.e., (HOG + SVM), provides better performance compared to others and hence outperforms the rest.

Table 10 Evaluation of deep learning methods under the second scenario: performance of 35 CNN-based chart classification models under fivefold cross-validation of in-house dataset

Model			Training		Testing							
					D1		D2		D3		In-house	
			μ	σ	μ	σ	μ	σ	μ	σ	μ	σ
VGG	16		94.21	±0.67	78.54	±0.59	81.06	±1.00	81.77	±0.51	89.46	±0.55
	19		94.74	±0.52	78.24	±1.09	81.05	±1.21	84.21	±0.31	88.83	±0.47
ResNet	50	v1	85.33	±1.53	69.74	±0.76	72.73	±0.56	79.46	±0.55	79.02	±1.01
		v2	86.73	±0.62	69.84	±0.52	73.28	±0.85	80.81	±0.72	78.93	±0.55
	101	v1	88.32	±0.72	69.62	±0.3	72.61	±1.03	81.72	±0.47	83.40	±0.53
		v2	87.60	±0.52	70.28	±0.28	74.2	±3.46	81.78	±0.47	83.90	±0.47
	152	v1	87.33	±0.61	71.05	±0.39	72.69	±1.03	80.04	±0.81	83.59	±0.55
		v2	88.11	±0.57	69.04	±0.81	73.65	±0.53	81.35	±0.70	83.44	±0.45
Inception	v3		89.21	±0.81	72.37	±0.71	73.03	±0.71	79.79	±0.21	79.89	±0.57
	v4		88.01	±0.62	72.55	±0.6	74.10	±0.15	79.86	±0.41	79.53	±0.94
AlexNet			88.21	±0.52	75.57	±1.39	78.51	±0.56	81.51	±0.57	84.10	±0.94
Xception			95.21	±0.31	**85.68**	±0.07	**86.77**	±0.01	**87.79**	±0.21	**90.34**	±0.11
Inception -ResNet			88.62	±0.98	74.58	±1.72	79.57	±0.61	80.17	±0.66	81.57	±1.12
MobileNet	V1		92.34	±0.28	76.55	±0.63	78.25	±1.19	80.81	±1.45	88.98	±0.19
	V2		92.47	±0.54	76.34	±0.78	77.94	±1.22	80.28	±0.49	88.87	±0.53
DenseNet	121		93.71	±0.89	84.63	±0.74	85.82	±0.65	86.51	±0.48	89.81	±0.25
	169		94.81	±0.54	84.06	±0.63	85.20	±1.14	85.61	±1.07	89.74	±0.8
	201		85.31	±2.89	83.94	±0.63	84.39	±1.21	84.11	±0.74	89.05	±1.70
EfficientNet	B0		70.67	±2.02	58.73	±1.15	55.69	±0.46	84.92	±0.75	64.91	±1.59
	B1		91.32	±2.01	82.09	±0.41	82.84	±0.50	85.72	±0.63	89.05	±0.49
	B2		92.10	±1.31	83.37	±1.95	84.11	±0.04	86.62	±0.47	86.59	±0.43
	B3		93.21	±0.92	85.28	±2.04	84.71	±0.42	88.12	±0.55	89.09	±0.15
	B4		94.02	±1.32	81.01	±0.21	81.67	±0.47	81.02	±0.75	89.54	±0.38
	B5		94.07	±1.02	82.40	±0.46	80.43	±0.55	81.79	±0.71	89.62	±0.37
	B6		93.21	±0.89	82.26	±1.13	81.66	±0.79	81.72	±0.88	89.87	±0.30
	B7		94,21	±0.92	81.64	±0.67	80.92	±0.79	81.15	±0.45	89.56	±0.38
EfficientNet V2	B0		94.62	±0.98	84.49	±0.23	84.68	±1.19	85.05	±0.37	89.23	±0.49
	B1		94.32	±0.82	85.37	±0.69	85.74	±0.58	86.18	±0.17	89.96	±0.17
	B2		93.67	±0.54	81.90	±1.71	83.80	±0.57	85.09	±0.44	89.63	±0.18
	B3		95.61	±0.52	85.12	±0.26	86.08	±0.39	87.02	±0.36	90.03	±0.06
	S		95.12	±0.62	85.44	±0.96	86.18	±0.89	86.82	±0.57	90.15	±0.08
	M		93.67	±0.54	85.36	±0.09	86.09	±0.72	87.12	±0.15	90.12	±0.38
	L		95.48	±0.54	84.72	±0.93	86.62	±0.54	87.57	±1.27	90.05	±0.09
NASNetLarge			**95.80**	±0.17	85.16	±1.67	86.06	±0.09	86.71	±0.25	90.16	±0.04
NASNetMobile			93.17	±0.42	82.85	±0.44	82.46	±0.57	81.58	±0.88	89.89	±0.22

μ and σ denote mean accuracy and standard deviation, respectively. The performance is reported for both training and testing. Bold entries represent the highest accuracy score

As stated earlier, the evaluation of 35 CNN-based models is done in two scenarios. Tables 9 and 10 show the fivefold cross-validation results of 35 models on the in-house (and further tested on D1, D2, and D3) under the first and second scenarios, respectively. From these tables, the following observations may be noted.

– As done with nine traditional ML classifiers, we investigate the overfitting issues for all 35 CNN-based models. In the same manner, we consider two parameters for the investigation, namely mean accuracy and standard deviation. In Tables 9, and 10, the similar characteristics that we noted for traditional classifiers (reported in Table 8) are observed. We observe no major difference in the mean training accuracy and testing accuracy (concerning in-house). It is further observed that most of the models accrued small standard deviations indicating that they are stable over different sample sets. On average, the differences between mean training accuracy and the testing accuracy are 5% (ranges from 1 to 8%), and 4% (ranges from 1% to 6%) for all models under the first and second scenario, respectively. Regarding the standard deviation, it is observed that the majority of the models accrued less than ±1.00 for both training and testing (concerning in-house) in both scenarios (ranges from ±0.04 to ±3.22 in the first scenario, and ±0.17 to ±2.89 in the second scenario). From all these observations, it can be said that all 35 CNN-based models are stable and free from overfitting issues.

- All the models provide better performance with the pre-trained weights.
- The best performance while training is provided by DenseNet-121 and NASNetLarge for the first and second scenarios, respectively. It is further observed that in the case of the first scenario, the performances of the models are inconsistent while evaluating the testing datasets, such as NASNetMobile provides the best result on D1, but VGG-19 outperforms the rest on D2, and Xception outperforms others on D3, and in-house. The opposite behavior is observed in the case of the second scenario, i.e., the consistency behavior is observed. The best performance is provided by NASNetLarge while training (multiple models also provide almost similar accuracy scores, such as Xception, EfficientNet V2 B3, EfficientNet V2 S, and EfficientNet V2 L). Still, Xception outperforms the rest while evaluating against all testing datasets.
- Various models provide different performance characteristics for different scenarios. For instance, AlexNet outperforms all the versions of ResNet, and all the versions of EfficientNet V2 in the second scenario, while it underperforms them in the first scenario. In another instance, both versions of MobileNet outperform all the versions of EfficientNet V2 (except for EfficientNet V2 L) in the first scenario. Still, the opposite relationship is observed in the second scenario.

From Tables 9 and 10, it can be concluded that using pre-trained weights, all the models provide better performance. As all the models provide their best performance in the second scenario where they are loaded with pre-trained weights, we present their performance discussion and analysis (while evaluating against the new samples or testing datasets) considering the second scenario.

- All 35 classifiers provide the best results with the in-house dataset. In the traditional classification approach, all the models perform poorly over dataset D1 compared to D2 and D3. However, in this case, EfficientNet B5, B6, and B7 provide better results over D1 than the other two. Most of the models provide better results in the given sequence: in-house, D3, D2, and D1.
- With the in-house dataset, the best-performed models are Xception, EfficientNet V2-B3, S, M, L, and NASNetLarge. These six models provide a mean accuracy of $\geq$ 90%. Following them, VGG-16, all versions of DenseNet, all variants of EfficientNet (except for B0, and B2), EfficientNet V2-B0, B1, B2, and NASNetMobile provide a mean accuracy above $\geq$ 89%. Further, both versions of MobileNet and VGG-19 provide mean accuracy above 88%. Finally, other remaining models provide a mean accuracy of less than 85%. It can be seen from the table that all six best-performed models provide consistent performance as denoted by small σ.
- With Dataset D1, Xception, EfficientNet-B3, EfficientNet V2-B3, S, M, NASNetLarge outperforms all other models by accruing a mean accuracy of $\geq$85%. Among them, Xception and EfficientNet V2-M provide more consistent performance with σ of only ±0.07, and ±0.09, respectively.
- In case of Dataset D2, with the mean accuracy of $\geq$ 86%, the best performance comes from seven models: Xception, EfficientNet V2-B3, S, M, L, NASANetLarge. Among these seven models, Xception and NASNetLarge provide more consistent performance with the σ of only ±0.01 and ±0.09, as shown in the table.
- Finally, with dataset D3, there are three best-performed models, viz: Xception, EfficientNet V2-B3, M, and L, with a mean accuracy of $\geq$87%. EfficientNet V2-M offered a more stable performance with a σ of only ±0.15. Xception follows it with a σ of ±0.21.
- Among all the classifiers, ResNet, Inception, and combined provide the least performance for all the datasets.
- As observed in the above points, Xception and EfficientNet - V2 B3 are common models for all four datasets that provide high mean accuracy. Among these two models, Xception delivers more consistent performance.

5.2.1 Effect of training sample sizes on classification performance

As noted above, different studies have considered datasets of different sizes and reported different performances. To understand the sensitivity of the models, we further built the above models with different training sample sizes. Our dataset is divided into fivefold, and onefold is kept as a testing set. We build four classifiers from the remaining folds by incrementally considering samples in onefold, twofold, threefold, and fourfold. These four classifiers are built for all nine tradi-

Table 11 Accuracy of 9 traditional classification setups with the increase in the training example

Model	Training size (# sample)			
	23454	46908	70362	92816
HOG+SVM	79.36	83.97	87.02	87.62
LBP+SVM	68.91	78.01	82.78	78.22
GLCM+SVM	75.67	80.18	80.21	81.34
HOG+KNN	75.64	81.43	81.67	81.96
LBP+KNN	68.08	74.12	82.01	81.52
GLCM+KNN	70.03	76.04	80.59	82.71
HOG+RF	68.01	72.02	77.91	82.45
LBP+RF	70.21	79.32	82.01	81.17
GLCM+RF	81.01	83.98	87.12	84.42

tional classification setups and the 35 CNN-based models. For the first set of experiments with a single fold, the number of training samples ranges from 400 to 1600. On average, it has 800 training samples per chart type. In the second set of experiments with twofold, another proportionate number of samples is added to the first setup for all the chart types, and so on.

Tables 11 and 12 show the accuracy of all the nine traditional classification setups and 35 CNN-based models (as the pre-trained CNN-based models provide better performance and outperform their version which is trained on chart samples from the scratch, we consider only the pre-trained version), respectively. The tables show that different classification models provide different convergence characteristics with the increase in training size. We observe the following different convergence characteristics:

1. Some models manage to increase the accuracy with the increase in training size. Among the traditional classifier, GLCM + SVM, and GLCM + KNN provide increasing performance with the increase in training size. For the CNN-based models, the same characteristics is observed in the following 18 models: AlexNet (AN), ResNet-101 v1 (R101v1), ResNet - 101 v2 (R101v2), Inception - ResNet (IR), DenseNet-121 (Dv121), DenseNet-169 (Dv169), EfficientNet B1 (EB1), EfficientNet B2 (EB2), EfficientNet B3 (EB3), EfficientNet B4 (EB4), EfficientNet B6 (E64), EfficientNet B7 (EB7), EfficientNetV2 B1 (Ev2B1), EfficientNetV2 B3 (Ev2B3), EfficientNetV2 Smaller (Ev2S), EfficientNetV2 Medium (Ev2M), NASNetMobile (NNM), and NASNetLarge (NNL). Hence they might be able to provide more profound performance with the increase in the training samples.
2. It is observed that an increase in the training size does not affect the performance of some models after some instances. HOG + KNN, MobileNet - v2 (MNv2), and EfficientNet V2 B0 (Ev2B0) converge at the second fold, which means they reached their saturation point with only 46,908 samples, and the increase in their training size has no effect afterward. In the same manner, the following two traditional classification setups and ten CNN-based models converge at the third fold: (HOG + SVM, HOG + RF), (VGG-16, VGG-19, ResNet-152 v1 (R152v1), ResNet-152 v2 (R152v2), DenseNet - 201 (Dv201), Xception, MobileNet v1 (MNv1), EfficientNet B5 (EB5), EfficientNet V2 B2 (Ev2B2), EfficientNet V2 Smaller (Ev2S)). So, they all reached their best-performed version with only 70,362 training samples.
3. We observed another situation where the models provided better performance with a lesser number of training samples. The following four traditional classification setups and five CNN-based models show their performance under this situation: (LBP + SVM, LBP + KNN, LBP + RF and GLCM + RF), (ResNet-50 v1 (R50v1), ResNet-50 v2 (R50v2), Inception v3 (Inv3), and Inception v4 (Inv4), EfficientNet B0 (EB0)). Except for EB0 (which provides the highest accuracy at the second fold), all other models provide the best performance at the third fold, but a drop in accuracy is observed at the fourth fold. In other words, we can state that they fail to provide coherent performance with the training size.

It can be inferred from the above observations (considering results provided by both traditional ML and DL models) that an increase in training sample size may not always provide a proportionate improvement in performance. However, an appropriate number of samples should be considered for each chart type to obtain convergence in performance. In our study, most of the classification setups start converging at the third fold. So the appropriate number of samples is about 2400 (= 800 × 3) samples for each chart type in this study.

6 Error analysis

The above sections show that the best classification performance is about 90.34% mean accuracy, which is obtained with Xception over the in-house dataset. The reduction in performances of all the classifiers is also observed with the datasets D1, D2, and D3. To understand the reason for the low performances of the classification models and the reduction of performances over the other three datasets, we perform the following error analysis. Considering *Xception* as one of the most stable classification models with all four datasets, as observed in the above sections, we consider it for the error analysis. The analysis has been performed from two different perspectives:

Table 12 Accuracy of 35 CNN-based chart classification models with the increase in the training example

Model	Training size (# sample)			
	23454	46908	70362	92816
AN	71.87	79.56	83.14	88.34
VGG16	67.89	73.98	88.82	88.23
VGG19	71.08	74.87	89.98	89.45
R50v1	67.12	69.14	78.09	74.99
R50v2	68.34	72.91	79.78	77.24
R101v1	71.11	77.91	81.89	83.06
R101v2	74.14	78.12	81.90	83.98
R152v1	77.76	79.12	83.01	83.87
R152v2	78.67	81.04	84.48	84.05
Inv3	68.12	71.67	80.98	79.12
Inv4	72.12	77.04	80.45	79.01
IR	69.21	74.21	79.34	82.16
MNv1	68.12	73.67	87.00	87.05
MNv2	62.31	89.67	89.12	89.67
Xception	72.90	82.12	91.35	91.57
Dv121	67.42	72.89	89.95	90.48
Dv169	71.01	74.12	86.12	89.67
Dv201	78.34	81.67	89.98	89.99
EB0	62.34	64.22	62.12	63.67
EB1	58.99	66.43	78.98	88.55
EB2	60.23	64.69	75.33	86.32
EB3	64.22	68.89	78.21	89.12
EB4	62.89	64.89	79.34	89.45
EB5	62.89	72.11	89.67	89.90
EB6	59.89	67.12	79.12	88.98
EB7	71.09	79.78	85.98	90.01
Ev2B0	72.22	89.76	89.11	89.99
Ev2B1	69.21	78.34	89.99	90.12
Ev2B2	69.01	79.76	89.34	89.98
Ev2B3	71.11	78.78	88.63	90.12
Ev2S	69.78	73.89	89.11	89.98
Ev2M	73.11	88.90	89.92	91.02
Ev2L	74.11	84.32	90.02	90.57
NNL	69.35	69.11	89.34	90.12
NNM	57.98	64.13	78.98	90.02

- *Confusing Chart-pairs:* Charts of similar characteristics may lead to misclassification.
- *Noise in Chart:* Presence of noise may lead to misclassification.

6.1 Confusing chart class pairs and their effects

In order to identify confusing chart class pairs leading to misclassifications, we analyze the distribution of the misclassified samples across all chart class pairs. Table 13 shows the average number of classification outputs (over fivefold cross-validation, rounded to integer) obtained with the Xception model. The rows indicate the input chart types, and the columns indicate the predicted chart types. The diagonal element could be interpreted as the percentage of correct classification for any given sample. Among all, any samples from the Arc chart have 100% of correct classification. It is followed by a column chart with a correct classification of 96%, and so on. It can be further stated that among all 28 chart types, classifying box charts correctly is the most challenging, followed by the waterfall chart. If the probability of misclassifying a sample from chart type x as chart type y is above a threshold t, we consider the class pair as confusing class pair. We have experimented t with 0.02, 0.03, and 0.04. With 0.04 (i.e., 4% of misclassification), we are able to find a strong overlapping characteristic among the misclassified samples.[11] Considering the 4% threshold, we observe the following 15 confusing chart class pairs from Table 13. The chart type-wise error characteristics and their confusing class pairs are briefly reported below.

1. **Area chart:** Area charts with multiple regions denoted by parallel or nearly parallel sharp edges are often confused with bar charts. In addition to this, some Area chart samples that have distinct colored edges but fill up with shaded color are sometimes classified as line charts. So, with Area chart, we obtained two confusing class pairs: *(Area, Bar)* and *(Area, Line)*.
2. **Box chart:** Some Box chart samples with huge-sized multiple boxes are often confused with Dendrogram. So, with the Box chart, we have one confusing chart pair: *(Box, Dendrogram)*.
3. **Block Diagram:** Some block diagram samples with multiple shapes such as rectangles, trapezium, rhombus are often confused with flowcharts. So, with the block diagram, we have one confusing class pair: *(Block, Flowchart)*.
4. **Bubble chart:** The bubble charts with small-size bubbles and highly visible background grids are sometimes classified as Node links. From the Bubble chart, we have one confusing class pair: *(Bubble, Node)*.
5. **Doughnut chart:** Visually Doughnut chart is very close to pie chart. Some doughnut charts with very little space in the middle are often confused with pie charts. With the doughnut chart, we have one confusing class pair: *(Doughnut, Pie)*.
6. **Line chart:** Line charts with bigger sizes of nodes to indicate data points are sometimes confused with Node

[11] With 0.02 or 0.03 as t, we fail to confidently draw a similar characteristic among the misclassified samples; instead, most of the misclassification is mainly because of noisy samples (which will be discussed in the next section) and lack of training samples.

Table 13 Average (of fivefold cross-validation, rounded to integer) classification outputs over in-house dataset

	AC	AR	BR	BX	BL	BB	CM	DG	DN	FC	GT	HM	HL	LN	MH	PND	PR	PL	PI	RD	RM	ST	SB	SF	TB	TM	VE	WF
AC	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
AR	0	86	6	0	0	0	0	0	0	0	0	0	0	5	0	0	1	1	0	0	1	0	0	0	0	0	0	0
BR	0	8	92		0	0	0 0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
BX	0	0	2	77	0	0	0	0	15	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	3	1	0	0
BL	0	0	0	0	91	0	0	0	0	7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0
BB	0	0	0	0	0	80	0	1	2	1	0	0	0	0	0	9	0	0	0	0	0	3	0	0	1	0	2	0
CM	0	0	0	2	3	0	96	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
DG	0	0	2	3	3	0	0	85	0	3	0	0	0	0	0	0	0	0	0	0	2	1	1	0	0	0	0	0
DN	0	0	4	0	0	0	0	0	92	0	0	0	0	0	0	0	0	0	5	0	0	0	0	1	0	0	2	0
FC	0	3	0	0	3	0	0	4	0	89	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
GT	0	0	3	0	0	0	0	0	0	3	84	1	0	0	0	0	0	0	2	0	0	3	0	0	0	2	0	2
HM	0	0	2	0	0	2	0	1	0	1	0	87	0	0	0	1	0	0	0	2	0	0	0	0	0	3	2	0
HL	0	0	0	6	0	0	0	0	0	2	0	0	81	4	0	0	1	0	0	1	0	0	1	0	3	1	0	0
LN	0	0	6	0	0	0	0	0	0	0	0	0	0	89	0	5	0	0	0	0	0	1	0	0	0	0	0	0
MH	0	2	0	0	0	0	1	0	0	0	0	1	0	0	85	0	0	0	0	0	0	9	0	0	2	0	0	0
ND	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	87	0	0	0	0	0	9	0	0	0	0	1	0
PT	0	0	0	0	0	0	3	0	0	0	0	0	0	2	0	1	86	0	0	1	0	0	1	2	1	0	0	3
PR	0	0	0	0	0	0	3	0	0	0	0	0	0	1	0	1	0	88	0	0	0	0	3	3	0	1	0	0
PI	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	89	0	0	0	2	0	0	0	7	0
RD	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	2	0	3	82	0	0	3	0	0	0	8	0
RO	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	2	0	0	0	0	87	2	0	0	4	3	0	0
ST	0	0	0	0	0	0	0	0	0	0	0	0	0	14	0	0	0	0	0	0	0	83	0	0	3	0	0	0
SB	0	0	2	0	0	0	0	0	0	0	0	1	0	0	2	0	1	0	1	0	0	0	90	0	0	0	3	0
SF	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	1	0	0	9	0	0	0	0
TB	0	0	0	0	0	0	0	0	0	2	0	2	0	0	0	0	0	0	0	0	0	7	0	0	89	0	0	0
TM	0	0	0	0	0	0	2	0	0	0	0	11	0	0	0	0		0	0	0	0	0	0	0	0	87	0	0
VE	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	3	2	0	0	0	2	1	0	0	90	0
WF	0	2	3	3	1	0	2	0	0	0	0	3	0	0	1	2	0	0	0	0	1	3	0	0	0	0	0	79

AC-Arc, AR-Area, BR-Bar, BX-Box, BL-Block, BB-Bubble, CM-Column, DG-Dendrogram, DN-Doughnut, FC-Flowchart, GT-Gantt, HM-Heatmap, HL-High–low, LN-Line, MN-Manhattan, ND-Node, PR-Pareto, PL-Parallel, PI-Pie, RD-Radar, RM-Reorderable Matrix, ST-Scatter, SB-Sunburst, SF- Surface plot, TB-Table, TM-Treemap, VE-Venn, WF-Waterfall links. Further, some of the Line charts with various colored backgrounds are sometime identified as Bar charts. So, with the Line chart, we observed two confusing class pairs: *(Line, Node)* and *(Line, Bar)*.

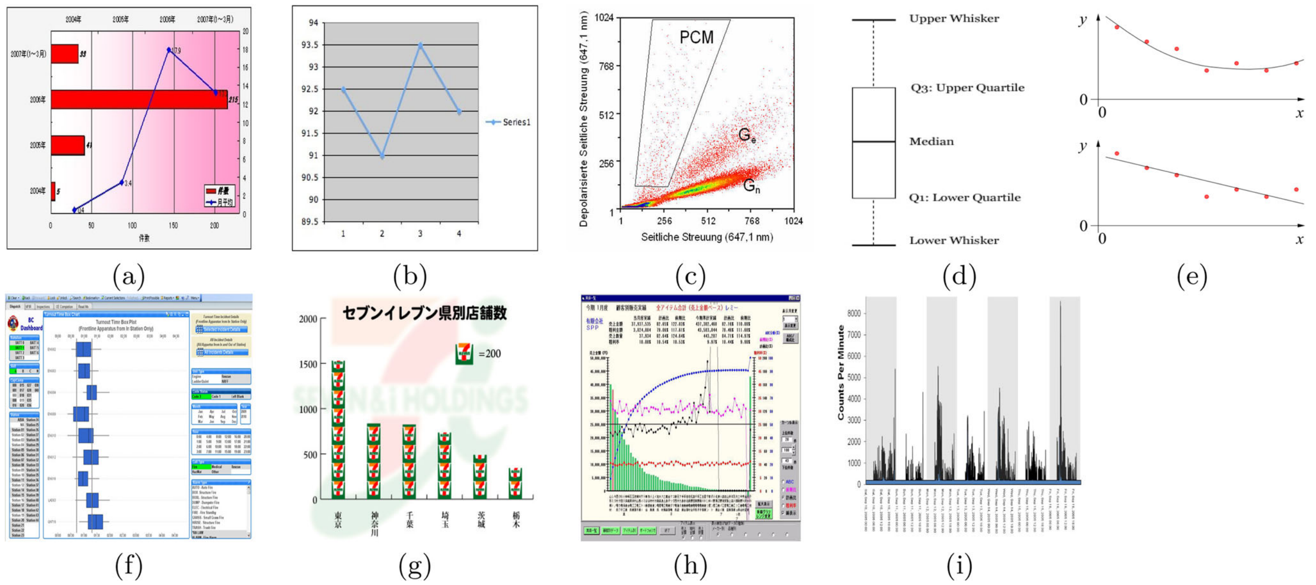
7. **Manhattan chart:** Manhattan charts with an enormous amount of data with no clear edges to indicate vertical margins are often classified as Scatter charts. From Table 13, it is observed that these two charts are among those charts which are highly confused. So, with Manhattan scatter, we have one confusing class pair: *(Manhattan, Scatter)*.
8. **Node Link:** Some samples of Node links with small nodes but low-intensity links are frequently classified as Scatter charts. So, with this chart type, we have one confusing class pair: *(Node, Scatter)*.
9. **Pie chart:** It is observed that Pie charts with one partition dominating the other are sometimes classified as Venn. In another case, chart images with multiple pies which have minimum gaps between them are also prone to be classified as Venn. So, with Pie chart, we have one confusing class pair: *(Pie, Venn)*.
10. **Radar chart:** Although most Radar charts have hexagonal outer layers, they may have a circle or nearly circle-like outer layers. Those samples are often misclassified as Venn diagrams. So, with this chart type, we only have one confusing class pair: *(Radar, Venn)*.
11. **Scatter Chart:** The scatter charts with lines are sometimes misclassified as line charts. Like the pair (Manhattan, Scatter) these two charts are among those which are highly confused. So, with the Scatter chart, we have one confusing class pair: *(Scatter, Line)*.
12. **Table:** Some samples of Table charts without borders and with crowded data are often classified as the scatter chart type. So, we have one confusing class pair with Table: *(Table, Scatter)*.
13. **Treemap** : From Table 13, it is observed that 11% of Treemap samples are classified as Heatmap. They appear to be visually similar most of the time. The main difference is that Treemap has thick or highly visible edges for each block. So, we have another confusing class pair: *(Treemap, Heatmap)*.

Table 14 Performance of Xception with respect to the 15 confusing chart class pairs for all four datasets. TP and FN are the true positive classification and false negative classification, respectively

Confusing	Dataset							
Chart pair	In-house		D1		D2		D3	
	TP	FN	TP	FN	TP	FN	TP	FN
(Area, Bar)	21	79	1	4	1	21	–	–
(Area, Line)	16	47	0	0	0	0	–	–
(Box, Dendrogram)	11	31	–	–	–	–	0	14
(Block, Flowchart)	16	28	–	–	–	–	–	–
(Doughnut, Pie)	9	42	–	–	–	–	–	–
(Bubble, Node)	20	37	–	–	–	–	–	–
(Line, Bar)	11	15	0	0	0	0	2	51
(Line, Node)	9	32	0	0	2	20	2	33
(Manhattan,) Scatter)	5	37	–	–	–	–	–	–
(Node, (Scatter)	21	11	–	–	–	–	–	–
(Pie, Venn)	12	60	0	0	4	6	3	21
(Radar, Venn)	14	56	0	0	0	0	0	0
(Scatter, Line)	21	197	0	0	9	45	4	71
(Table,) (Scatter)	19	34	0	0	1	24	3	42
(Treemap, Heatmap)	11	94	–	–	–	–	–	–
CTS	7.36		1.56		3.43		5.39	
TCMC	83.30		80.00		87.21		94.30	
TCMCO	6.51		1.25		3.00		5.14	

Table 14 shows the classification outputs of the confusing class pairs over the four datasets using Xception. As mentioned above, a pair (X, Y) in the table denotes the misclassification of the input samples from the chart type X as chart type Y. So, the five samples of D1 that contribute to the pair (Area, Bar) are five area chart samples (with the particular characteristics mentioned at the beginning of this section) which get classified as Bar chart type. The CTS in the table shows the percentage of confusing chart samples for a given testing dataset. It is defined as $CTS = \left(\frac{\sum_{(X,Y)}(TP_{X,Y}+FN_{X,Y})}{TS}\right) \times 100$, where $TP_{(X,Y)}$ and $FN_{(X,Y)}$ are the true positives and false negatives, respectively, for (X, Y) pair, and TS denotes the entire testing samples. From Table 14, it can be seen that the in-house dataset contributes to all 15 confusing chart pairs by occupying 7.36% of the whole testing dataset. D3 dataset contributes to only six confusing chart pairs: (Box, Dendrogram), (Line, Bar), (Line, Node), (Pie, Venn), (Scatter, Line), and (Table, Scatter). These six confusing chart pairs occupy 5.39% of D3 dataset. D2, on the other hand, contributes only 3.43% of its population by five types of confusing chart pairs ((Area, Bar), (Line, Node), (Pie, Venn), (Scatter, Line), (Table, Scatter)). Among the four datasets, the smallest contribution of confusing chart pairs comes from D1. It contributes only 1.56% of its population by only one type of confusing chart pair, which is (Area, Bar).

The TCMC (Total Confusing pairs misclassification) and TCMCO (Total Confusing pairs misclassification overall)



(a) (b) (c) (d) (e)

(f) (g) (h) (i)

Fig. 3 Samples of nine types of chart noise: **a** Composite Chart (CC), **b** Hard Background Grid (HBG), **c** Additional Information (AI), **d** Text Noise (TNS), **e** Transparent Background (TB), **f** Improper Image Screenshot (IIS), **g** Complex Background (CB), **h** Numerous Component (NC), **i** Patterned Background (PB)

rows in Table 14 further show the error contributions (because of confusing chart pairs) with respect to the total confusing pair samples and with respect to the entire dataset, respectively. It is estimated as the macroaverage percentage of sample misclassification between the confusing chart pairs, i.e., $TCMC = \left(\sum_{(X,Y)} \frac{FN_{X,Y}}{TP_{X,Y}+FN_{X,Y}}\right) \times 100$. Similarly, TCMCO is defined by the percentage of misclassifications from the confusing pairs over the entire testing samples (TS) and estimated as below.$TCMCO = \left(\frac{\sum_{(X,Y)} FN_{X,Y}}{TS}\right) \times 100$. From the table, the following points are observed:

- *In-house dataset:* 83.30% of the confusing samples are misclassified. It is further observed that 6.51% of the misclassification (overall) is because of the confusing samples.
- *D1 Dataset):* 80% of the confusing samples are misclassified. It contributes 1.25% of misclassifications overall.
- *D2 Dataset:* 87.21% of the confusing samples are misclassified. It contributes 3% of misclassifications overall.
- *D3 Dataset:* 94.3% of the confusing samples are misclassified. It contributes 5.41 % of misclassifications overall.

From the above discussion, it is observed that confusing chart class pairs affect the performance of the chart classification model. They contribute more than 5% of the misclassification in the case of in-house and D3 datasets but less than 5% for the datasets D1 and D2. Among the four datasets, both in-house and D3 (in which misclassification of confusing class pairs is high) have a higher number of samples per class than the other two. This can infer that with the increase in chart type and the inclusion of various samples, the number of confusing class pairs is likely to increase.

6.2 Noise types and their effects

Apart from confusing chart pairs, misclassification can also be due to the presence of noise. In order to identify noise types leading to misclassifications, we analyze the distribution of the misclassified samples of all chart types. We observe the following 9 noise types. The noise type, error characteristics, and their effects are discussed below.

1. **Composite-like chart type (CC)**: A chart with an extra component that resembles other chart types, as shown in Fig. 3a. It is actually a bar chart but composed of bars and two lines.
2. **Hard Background Grid (HBG)**: A chart with hard and dominating background grid lines as shown in Fig. 3b.
3. **Additional Information (AI)**: A chart with embedded information presented in the form of shapes such as circles, rectangles as shown in Fig. 3c.
4. **Text Noise (TNS)** : A chart with an enormous amount of additional information presented in the form of text as shown in Fig. 3d.
5. **Transparent Background (TB)**: A chart image with a completely transparent background, as shown in Fig. 3e.
6. **Improper Image Screenshot (IIS)**: An image with some additional unrelated document regions as shown in Fig. 3f.

Table 15 Performance of Xception with respect to the nine noise types over four datasets. TP and FN are the true positive classification and false negative classification, respectively

Testing	Noise type																		NTS	TNMC	TNMCO
dataset	CC		HBG		AI		TNS		TB		IIS		CB		NC		PB				
	TP	FN	TP	FN	TP	FN	TP	FN	TP	FN	TP	FN	TP	FN	TP	FN	TP	FN			
In-house	0	0	340	181	0	0	0	0	0	0	0	0	0	0	0	0	172	184	5.20	37.73	3.65
D1	0	3	12	10	2	6	5	11	0	0	3	2	6	8	0	3	1	14	27.50	63.63	17.50
D2	0	0	38	176	0	79	46	81	0	14	26	4	39	101	0	5	8	79	18.08	77.60	14.03
D3	6	20	45	95	148	94	11	67	0	9	0	0	14	96	5	9	16	105	19.64	65.80	12.37

7. **Complex Background (CB)**: A chart with a distinct background watermark, as shown in Fig. 3g.
8. **Numerous Components (NC)**: A chart with multiple chart components, such as additional shapes and text, as shown in Fig. 3h.
9. **Patterned Background (PB)**: A chart image that has a background with patterns, such as shown in Fig. 3i. It is an area chart, but because of the vertical blocks in the background, it is misclassified.

For any given testing dataset, we manually collect the noisy samples considering the characteristics and patterns discussed above. So, for any testing dataset, we form a subset considering only the identified noisy samples. Table 15 shows the classification outputs of the noisy samples of the four datasets using Xception. Among all four testing datasets, In-house contributes very few noisy samples as compared to the other remaining datasets. The NTS in the table shows the percentage of noisy samples for a given testing dataset. For any given dataset with N number of noise types, the NTS may be defined as $NTS = \left(\frac{\sum_i^N (TP_i + FN_i)}{TS}\right) \times 100$ where TP_i and FN_i are the true positives and false negatives, respectively, for the noise type i. In-house dataset contributes to only two types of noise, viz., Hard background grid (HGB) and Patterned background (PB) by providing NTS of only 5.2%. Except for the noise type Transparent background (TB), the dataset D1 contributes to all noise types. 27.50% of its dataset is occupied by the noisy samples. Leaving the noise type Composite chart (CC), the dataset D2 contributes to all other remaining nine noise types. Noisy samples from these nine types occupy 18.08% of its dataset. Finally, the dataset provided by D3 occupies 19.64% of its dataset with nine noise types (leaving Improper Image Screenshot (IIS)).

The TNMC (Total noise misclassification) and TNMCO (Total noise misclassification overall) rows in Table 15 further show the overall error contributions with respect to chart noise and over the entire dataset. TNMC is estimated as the macroaverage percentage of sample misclassification between the noisy samples, i.e., $TNMC = \left(\sum_i^N \frac{FN_i}{TP_i+FN_i})\right) \times 100$. Similarly, TNMCO is defined by the percentage of misclassifications from the noisy samples over the entire testing samples (TS) and estimated as $TNMCO = \left(\frac{\sum_i^N FN_i}{TS}\right) \times 100$. From the table, the following points are observed:

- *In-house dataset:* 37.73% of the noisy samples are misclassified. It is further observed that 3.65% of the misclassification (overall) is because of the noisy samples.
- *D1 Dataset):* 63.63% of the noisy samples are misclassified. It contributes 17.5% of misclassifications overall. It may be noted that not a single instance of CC and NC types are correctly classified.
- *D2 Dataset:* 77.60% of the noisy samples are misclassified. It contributes 14.03% of misclassifications overall. It may be noted that not a single instance of AI, TB, and NC types is correctly classified.
- *D3 Dataset:* 65.80% of the noisy samples are misclassified. It contributes 12.37% of misclassifications overall. It may be noted that not a single instance of TB is correctly classified.

From the above discussion, it is observed that besides confusing chart class pairs, chart noise is another issue to consider while developing a chart classification model. Apart from the in-house dataset, it can be said that the primary cause of the misclassification is noisy samples.

6.3 Response of the model on 3D samples

As our study does not consider 3D chart images except for surface plots, 3D samples present in other publicly available datasets become one reason for misclassification. The datasets D1 and D2 have only two and five 3D samples, which are classified incorrectly. The dataset D3 has 99 3D samples, of which 21 are classified correctly. It is because those images are not entirely in the third dimension, but only some of the chart's components are.

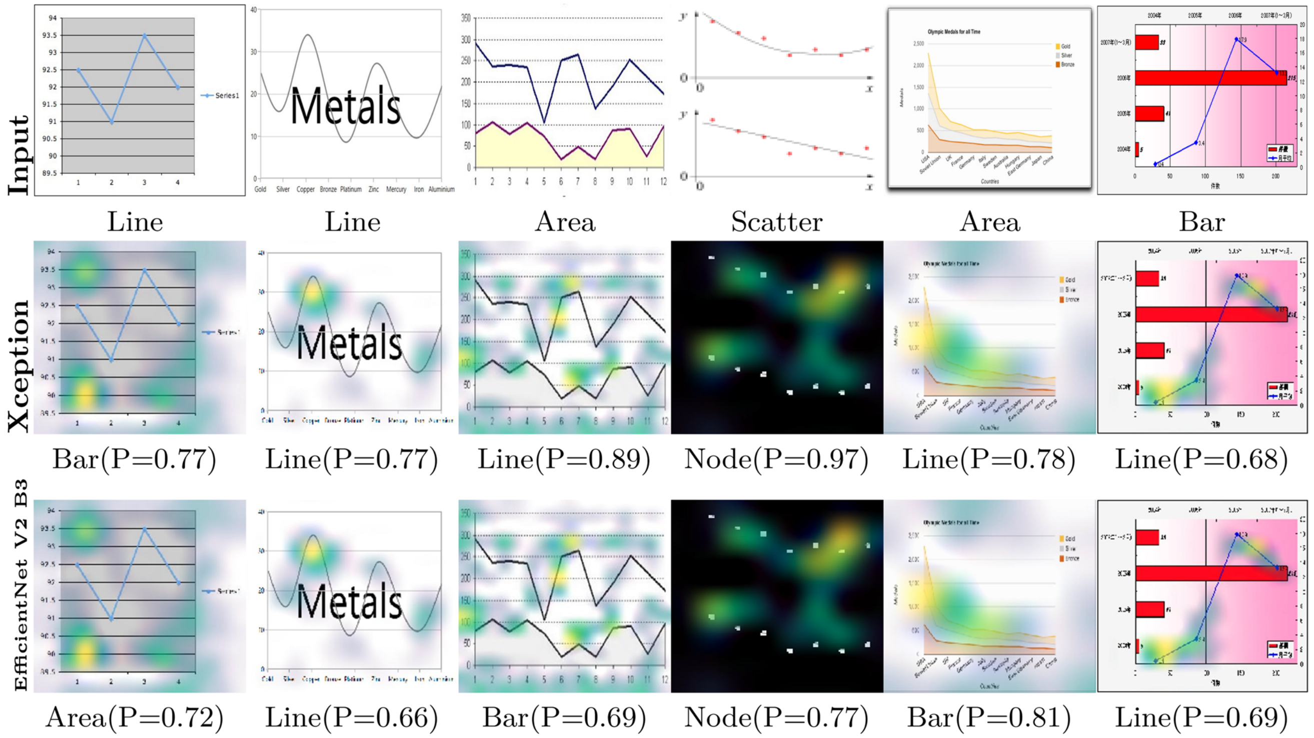


Fig. 4 Grad-CAM [78] visualization result: Comparison of the visualization results of input image (images in first row), responses from Xception (images in the second row), and responses from EfficientNet v2 B3 (images in the last row). The grad-CAM visualization is calculated from the last convolutional outputs. *P* denotes the softmax score of each network for the classified class

7 Conclusions and potential future directions

This paper presented a review of the studies on chart classification reported in the literature since 2001. Noting that the majority of the earlier studies had considered smaller datasets with a smaller number of classes, this paper further evaluated chart classification methods (44 in total) under a common experimental framework over a large dataset with a larger number of classes (28 classes). To the best of our knowledge, this paper has considered the largest number of chart classification models and the largest dataset. In addition to a theoretical review of the existing methods, it also attempted to benchmark their performances under different experimental setups. Further, it also identified two challenging aspects: noise in the chart and confusing chart pairs. In short, this paper made the following contributions.

1. *Brief survey:* This study reviewed studies on chart classification studies since 2001 from three different aspects—datasets, feature extractors, and methods.
2. *Largest chart image corpus:* The paper provides the extended version of the dataset reported in our earlier study. It consists of 117,271 samples over 28 chart types. To the best of our knowledge to date, the in-house is the largest dataset among the publicly available and non-publicly available datasets in terms of sample size and the number of chart types.
3. *Evaluation of 44 ML models:* The paper evaluates 44 machine learning-based chart classification models under a common experimental framework. It is the largest comparison reported in the domain of chart classification.
4. *Challenges in chart classification:* On top of the challenges reported in our earlier paper, this paper also contributed on discovering more confusing chart class pairs and also investigated their effects on chart classification performances.

As evident in Tables 14 and 15, noise present in the chart and confusing chart class pairs affect the classification performance significantly. It brought out an interesting future research aspect. To understand it deeper, Fig. 4 presents a visualization of the attentive positions captured in the last convolution layer of chart classification models (Xception, and EfficientNet V2 B3) of a noisy/confusing image sample. The attentive positions inherently captured in a vector can be visualized using Grad-CAM [78]. Grad-CAM calculates the attentive spatial positions in a convolution layer using gradients. It can be seen that the attentive positions captured by both Xception and EfficientNet v2 B3 do not match the target chart regions for most of the input images. As a result, they fail to classify the input images correctly. It is further observed that Xception misclassifies any chart type with low confidence compared to EfficientNet V2 B3. As shown in the figure, for the input chart type scatter (fourth input image from the left), both the models classified it as node link type, but EfficientNet V2 B3 is more confident than Xception by providing a probability score of 0.97.

As several studies on image classification (not explored with chart classification yet) tasks have shown encouraging performances with attention-based models, a study exploring attention-based deep learning models for chart classification could be a future research direction for handling chart noise. Further, the issue of confusing chart pairs may also be addressed by exploring *triplet loss* learning. As applied in several studies [79–81], triplet loss can incorporate intra-class and increase inter-class distances while generating embedding of the samples. Therefore, incorporating triplet loss into chart classification models could also be an interesting research direction.

References

1. Liu, Y., Lu, X., Qin, Y., Tang, Z., Xu, J.: Review of chart recognition in document images. Vis. Data Anal. **2013**(8654), 384–391 (2013)
2. Davila, K., Setlur, S., Doermann, D., Kota, B.U., Govindaraju, V.: Chart mining: a survey of methods for automated chart analysis. IEEE Trans. Pattern Anal. Mach. Intell. **43**(11), 3799–3819 (2020)
3. Shahira, K.C., Lijiya, A.: Towards assisting the visually impaired: a review on techniques for decoding the visual data from chart images. IEEE Access **9**, 52 926-52 943 (2021)
4. Futrelle, R.P., Kakadiaris, I.A., Alexander, J., Carriero, C.M., Nikolakis, N., Futrelle, J.M.: Understanding diagrams in technical documents. Computer **25**(7), 75–78 (1992)
5. Zhou, Y., Tan, C.L.: Learning-based scientific chart recognition. In: 4th IAPR International Workshop on Graphics Recognition, GREC, vol. 7, pp. 482–492. Citeseer, Kingston (2001)
6. Davila, K., Kota, B.U., Setlur, S., Govindaraju, V., Tensmeyer, C., Shekhar, S., Chaudhry, R.: ICDAR 2019 competition on harvesting raw tables from infographics (chart-infographics). In: 2019 International Conference on Document Analysis and Recognition (ICDAR), pp. 1594–1599. IEEE, Sydney (2019)
7. Chagas, P., Akiyama, R., Meiguins, A., Santos, C., Saraiva, F., Meiguins, B., Morais, J.: Evaluation of convolutional neural network architectures for chart image classification. In: International Joint Conference on Neural Networks (IJCNN), Shenzhen, China, pp. 1–8 (2018)
8. Shao, M., Futrelle, R.P.: Recognition and classification of figures in pdf documents. In: Graphics Recognition. Ten Years Review and Future Perspectives, pp. 231–242. Springer, Berlin (2006)
9. Siegel, N., Horvitz, Z., Levin, R., Divvala, S., Farhadi, A.: Figureseer: parsing result-figures in research papers. In: Computer Vision-ECCV: 14th European Conference, Amsterdam, Oct 11–14, 2016, Proceedings, Part VII 14, vol. 2016, pp. 664–680. Springer, Amsterdam (2016)
10. Jobin, K., Mondal, A., Jawahar, C.: Docfigure: a dataset for scientific document figure classification. In: 2019 International Conference on Document Analysis and Recognition Workshops (ICDARW), vol. 1, pp. 74–79. IEEE, Sydney (2019)
11. Kaur, P., Kiesel, D.: Combining image and caption analysis for classifying charts in biodiversity texts. In: VISIGRAPP (3: IVAPP), Valletta, Malta, pp. 157–168 (2020)
12. Thiyam, J., Singh, S.R., Bora, P.K.: Chart classification: an empirical comparative study of different learning models. In: Proceedings of the 12th Indian Conference on Computer Vision, Graphics and Image Processing, pp. 1–9. ACM, New York (2021)
13. Clark, C., Divvala, S.: Pdffigures 2.0: mining figures from research papers. In: ACM/IEEE-CS, ser. JCDL'16, pp. 143–152. ACM, New York (2016)
14. Dai, W., Wang, M., Niu, Z., Zhang, J.: Chart decoder: generating textual and numeric information from chart images automatically. J. Vis. Lang. Comput. **48**, 08 (2018)
15. Morris, D., Müller-Budack, E., Ewerth, R.: Slideimages: a dataset for educational image classification. In: European Conference on Information Retrieval, ser. Lecture Notes in Computer Science, vol. 12036, pp. 289–296. Virtual Event: Springer (2020)
16. Futrelle, R.P., Shao, M., Cieslik, C., Grimes, A.E.: Extraction, layout analysis and classification of diagrams in pdf documents. In: International Conference on Document Analysis and Recognition (ICDAR), vol. 3, pp. 1007–1014. Edinburgh, Scotland
17. Prasad, V.S.N., Siddiquie, B., Golbeck, J., Davis, L.S.: Classifying computer generated charts. In: International Workshop on Content-Based Multimedia Indexing. Talence, France, vol. 2007, pp. 85–92 (2007)
18. Huang, W., Tan, C.L.: A system for understanding imaged infographics and its applications. In: Proceedings of the 2007 ACM Symposium on Document Engineering, ser. DocEng'07. Association for Computing Machinery, New York, p. 9–18 (2007)
19. Huang, W., Zong, S., Tan, C.L.: Chart image classification using multiple-instance learning. In: IEEE Workshop on Applications of Computer Vision (WACV'07), vol. 2007, pp. 27–27. IEEE, Washington, DC (2007)
20. Savva, M., Kong, N., Chhajta, A., Fei-Fei, L., Agrawala, M., Heer, J.: Revision: Automated classification, analysis and redesign of chart images. In: Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology, Santa Barbara, California, USA, pp. 393–402 (2011)
21. Gao, J., Zhou, Y., Barner, K.E.: View: visual information extraction widget for improving chart images accessibility. In: 2012 19th IEEE International Conference on Image Processing, Florida, USA, pp. 2865–2868 (2012)
22. Karthikeyani, V., Nagarajan, S.: Machine learning classification algorithms to recognize chart types in portable document format (pdf) files. Int. J. Comput. Appl. **39**, 1–5 (2012)
23. Giannakopoulos, T., Foufoulas, Y., Stamatogiannakis, E., Dimitropoulos, H., Manola, N., Ioannidis, Y.: Visual-based classification of figures from scientific literature. In: Proceedings of the 24th International Conference on World Wide Web, Florence, Italy, May 2015, pp. 1059–1060
24. Tang, B., Liu, X., Lei, J., Song, M., Tao, D., Sun, S., Dong, F.: Deepchart: combining deep convolutional networks and deep belief networks in chart classification. Signal Process. **124**, 10 (2015)
25. Amara, J., Kaur, P., Owonibi, M., Bouaziz, B.: Convolutional neural network based chart image classification. In: 25th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision: Plzen. Czech Republic, vol. 2017, pp. 83–88 (2017)
26. Jung, D., Kim, W., Song, H., Hwang, J.-i., Lee, B., Kim, B., Seo, J.: Chartsense: interactive data extraction from chart images. In: Proceedings of the 2017 Chi Conference on Human Factors in Computing Systems, New York, NY, USA, pp. 6706–6717 (2017)
27. Chagas, P., Freitas, A., Daisuke, R., Miranda, B., De Araújo, T.D.O., Santos, C., Meiguins, B., De Morais, J.M.: Architecture proposal for data extraction of chart images using convolutional

neural network. In: 21st International Conference Information Visualisation (IV), vol. 2017, pp. 318–323. IEEE, London (2017)

28. Poco, J., Heer, J.: Reverse-engineering visualizations: recovering visual encodings from chart images. Comput. Graphics Forum **36**, 353–363 (2017)
29. Bajić, F., Job, J., Nenadić, K.: Chart classification using simplified VGG model. In: 2019 International Conference on Systems, Signals and Image Processing (IWSSIP), pp. 229–233. IEEE, Osijek (2019)
30. Davila, K., Tensmeyer, C., Shekhar, S., Singh, H., Setlur, S., Govindaraju, V.: Icpr 2020-competition on harvesting raw tables from infographics. In: Pattern Recognition. ICPR International Workshops and Challenges: Virtual Event, January 10–15, 2021, pp. 361–380, Proceedings. Part VIII. Springer (2021)
31. Liu, P.N.X., Klabjan, D.: Data extraction from charts via single deep neural network. In: arXiv preprint arXiv:1906.11906 (2019)
32. Bajić, F., Job, J., Nenadić, K.: Data visualization classification using simple convolutional neural network model. Int. J. Electr. Comput. Eng. Syst. **11**(1), 43–51 (2020)
33. Araújo, T., Chagas, P., Alves, J., Santos, C., Sousa Santos, B., Serique Meiguins, B.: A real-world approach on the problem of chart recognition using classification, detection and perspective correction. Sensors 20(16) (2020)
34. Bajić, F., Job, J.: Chart classification using siamese cnn. J. Imaging **7**(11), 220 (2021)
35. Dadhich, K, Daggubati, S.C., Sreevalsan-Nair, J., Barchartanalyzer: digitizing images of bar charts. In: IMPROVE, Virtual Event, pp. 17–28 (2021)
36. Mishra, P., Kumar, S., Chaube, M.K.: Chartfuse: a novel fusion method for chart classification using heterogeneous microstructures. In: Multimedia Tools and Application, vol. 80, pp. 10417–10439 (2021)
37. Mishchenko, A., Vassilieva, N.: Model-based chart image classification. In: Advances in Visual Computing, pp. 476–485. Springer, Berlin (2011)
38. Balaji, A., Ramanathan, T., Sonathi, V.: Chart-text: a fully automated chart image descriptor. Comput. Vis. Pattern Recognit. **3**, 12 (2018)
39. Bajic, F., Job, J.: Data extraction of circular-shaped and grid-like chart images. J. Imaging **8**(5) (2022)
40. Cliche, M., Rosenberg, D., Madeka, D., Yee, C., Scatteract: automated extraction of data from scatter plots. In: Machine learning and knowledge discovery in databases: European conference, ECML PKDD: Skopje, Macedonia, September 18–22, 2017, Proceedings, Part I 10, vol. 2017, pp. 135–150. Springer (2017)
41. Singh, M., Goyal, P.: Chartsight: an automated scheme for assisting visually impaired in understanding scientific charts. In: VISIGRAPP 2021, pp. 309–318. SCITEPRESS, Austria (2021)
42. Otsu, N.: A threshold selection method from gray-level histograms. IEEE Trans. Syst. Man Cybern. **9**(1), 62–66 (1979)
43. Zhou, Y.P., Tan, C.L.: Hough technique for bar charts detection and recognition in document images. In: Proceedings 2000 International Conference on Image Processing (Cat. No. 00CH37101), vol. 2, pp. 605–608, Vancouver, BC, Canada (2000)
44. Huang, W., Tan, C.-L.: Locating charts from scanned document pages. In: 9th International Conference on Document Analysis and Recognition (ICDAR 2007), vol. 1, pp. 307–311. IEEE, Parana (2007)
45. Akmal, R., Othman, N.Z., Bade, A., Rahim, M., Amin, I.: Connected component labeling using components neighbors-scan labeling approach. J. Comput. Sci. **6**, 12 (2010)
46. Mishchenko, A., Vassilieva, N.: “Model-based recognition and extraction of information from chart images. J. Multimedia Process. Technol. **2**, 76–89 (2011)
47. Ishihara, T., Morita, K., Shirai, N.C., Wakabayashi, T., Ohyama, W.: Chart-type classification using convolutional neural network for scholarly figures. In: Pattern Recognition: 5th Asian Conference, ACPR 2019, Auckland, New Zealand, November 26–29, 2019, Revised Selected Papers, Part II 5, pp. 252–261. Springer, Auckland (2020)
48. Yokokura W.T., Naoko: Layout-based approach for extracting constructive elements of bar-charts. In: Graphics Recognition Algorithms and Systems, pp. 163–174. Springer, Berlin (1998)
49. Huang, W., Tan, C.L., Leow, W.K.: Associating text and graphics for scientific chart understanding. In: 8th International Conference on Document Analysis and Recognition (ICDAR’05), pp. 580–584. IEEE, Seoul (2005)
50. LeCun, Y., Boser, B., Denker, J.S., Henderson, D., Howard, R.E., Hubbard, W., Jackel, L.D.: Backpropagation applied to handwritten zip code recognition. Neural Comput. **1**(4), 541–551 (1989)
51. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. Commun. ACM **60**(6), 84–90 (2017)
52. Simonyan, K., Zisserman, A. (2014) Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556
53. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, Nevada, USA, pp. 2818–2826 (2016)
54. Szegedy, C., Ioffe, S., Vanhoucke, V., Alemi, A.: Inception-v4, inception-resnet and the impact of residual connections on learning. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 31(1), California, USA (2017)
55. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, Nevada, USA, pp. 770–778 (2016)
56. Chollet, F. Xception: deep learning with depthwise separable convolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, Hawaii , USA, pp. 1251–1258 (2017)
57. Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T. Andreetto, M., Adam, H.: Mobilenets: efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861 (2017)
58. Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4700–4708 (2017)
59. Kanjanawattana, S., Kimura, M.: ANNSVM: a novel method for graph-type classification by utilization of Fourier transformation, wavelet transformation, and hough transformation, BRAIN. Broad Res. Artif. Intell. Neurosci. **8**(2), 5–25 (2017)
60. Hinton, G.E., Salakhutdinov, R.R.: Reducing the dimensionality of data with neural networks. Science **313**(5786), 504–507 (2006)
61. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C.: Imagenet large scale visual recognition challenge. Int. J. Comput. Vis. **115**, 211–252 (2015)
62. Dey, S., Dutta, A., Toledo, J.I., Ghosh, S.K., Lladós, J., Pal, U.: Signet: convolutional siamese network for writer independent offline signature verification. arXiv preprint arXiv:1707.02131 (2017)
63. Lake, B.M., Salakhutdinov, R., Gross, J., Tenenbaum, J.B.: One shot learning of simple visual concepts. Cognit. Sci. **33** (2011)
64. Thiyam, J., Singh, S.R., Bora, P.K.: Challenges in chart image classification: a comparative study of different deep learning methods. In: Proceedings of the 21st ACM Symposium on Document Engineering, ser. DocEng’21. ACM, New York (2021)

65. Zhou, Y., Tan, C.L.: Hough-based model for recognizing bar charts in document images. In: Document Recognition and Retrieval VIII, vol. 4307, pp. 333–340. SPIE (2000)
66. Zhou, Y.P., Tan, C.L.: Hough technique for bar charts detection and recognition in document images. In: Proceedings 2000 International Conference on Image Processing (Cat. No. 00CH37101), vol. 2. IEEE, Vancouver, pp. 605–608 (2000)
67. Prasad, V.S.N., Siddiquie, B., Golbeck, J., Davis, L.S.: Classifying computer generated charts. In: International Workshop on Content-based Multimedia Indexing, vol. 2007, pp. 85–92. IEEE, Bordeaux (2007)
68. Karthikeyani, V., Nagarajan, S.: Machine learning classification algorithms to recognize chart types in portable document format (pdf) files. Int. J. Comput. Appl. **39**, 1–5 (2012)
69. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), vol. 1, pp. 886–893. IEEE, San Diego (2005)
70. Karis, M.S., Razif, N.R.A., Ali, N.M., Rosli, M.A., Aras, M.S.M., Ghazaly, M.M.: Local binary pattern (lbp) with application to variant object detection: A survey and method. In: IEEE 12th International Colloquium on Signal Processing & Its Applications (CSPA), vol. 2016, pp. 221–226. IEEE, Melaka (2016)
71. Deng, L.: A tutorial survey of architectures, algorithms, and applications for deep learning. APSIPA Trans. Signal Inf. Process. **3**, e2 (2014)
72. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: machine learning in Python. J. Mach. Learn. Res. **12**, 2825–2830 (2011)
73. Tan, M., Le, Q.V.: Efficientnet: rethinking model scaling for convolutional neural networks. ArXiv:1905.11946 (2019)
74. Tan, M., Le, Q.: Efficientnetv2: smaller models and faster training. In: International Conference on Machine Learning. Virtual Event: PMLR, pp. 10096–10106 (2021)
75. Zoph, B., Vasudevan, V., Shlens, J., Le, Q.V.: Learning transferable architectures for scalable image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, San Francisco, CA, USA, pp. 8697–8710 (2018)
76. Reyes, A.K., Caicedo, J.C., Camargo, J.E.: Fine-tuning deep convolutional networks for plant recognition. CLEF (Working Notes) **1391**, 467–475 (2015)
77. Kieffer, B., Babaie, M., Kalra, S., Tizhoosh, H.R.: Convolutional neural networks for histopathology image classification: training vs. using pre-trained networks. In: 2017 7th International Conference on Image Processing Theory, Tools and Applications (IPTA), pp. 1–6 (2017)
78. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D.: Grad-cam: visual explanations from deep networks via gradient-based localization. In: Proceedings of the IEEE international conference on computer vision, Venice, Italy, pp. 618–626 (2017)
79. Zhang, M., Su, H., Wen, J.: Classification of flower image based on attention mechanism and multi-loss attention network. Comput. Commun. **179**, 307–317 (2021)
80. Wang, J., Li, Y., Miao, Z., Zhao, X., Rui, Z.: Multi-level metric learning network for fine-grained classification. IEEE Access **7**, 166 390-166 397 (2019)
81. Cui, Y., Zhou, F., Lin, Y., Belongie, S.: Fine-grained categorization and dataset bootstrapping using deep metric learning with humans in the loop. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, pp. 1153–1162 (2016)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.