Effect of Attention and Triplet loss on Chart Classification: A study on Noisy Charts and Confusing Chart Pairs

Jennil Thiyam (jenni176155101@iitg.ac.in)
Indian Institute of Technology Guwahati
Sanasam Ranbir Singh
Indian Institute of Technology Guwahati
Prabin Kumar
Indian Institute of Technology Guwahati

Research Article

Keywords: Chart classificaiton, Chart's noise, Confusing chart class pair, Attention, Triplet loss

Posted Date: May 25th, 2022

DOI: https://doi.org/10.21203/rs.3.rs-1667899/v1

-

DECLARATION

Ethical Approval and Consent to participate
No

Consent for publication
Yes

Availability of supporting data
No

Competing interests
No

Funding
Not Applicable

Authors' contributions

Jennil Thiyam is a Ph.D. research scholar under the guidance of **Sanasam Ranbir Singh** as main supervisor and **Prabin Kumar Bora** as secondary supervisor. So, most of the work is done by the first author only with the help of Sanasam Ranbir Singh, and Prabin Kumar Bora. All the authors contributed to writing the manuscript. However, the main contributor is Jennil Thiyam. All authors reviewed the manuscript

Acknowledgments
Not applicable

Effect of Attention and Triplet loss on Chart Classification: A study on Noisy Charts and Confusing Chart Pairs

Jennil Thiyam*, Sanasam Ranbir Singh and Prabin Kumar Bora

Indian Institute of Technology, Guwahati.

*Corresponding author(s). E-mail(s): jenni176155101@iitg.ac.in;
Contributing authors: ranbir@iitg.ac.in; prabin@iitg.ac.in;

Abstract

Charts are powerful tools for visualizing and comparing data. With the increase in the presence of various chart types in scientific documents in electronic media, the development of automatic chart classification system is becoming an important task. Existing studies on chart classification fail to address the presence of noise in charts, and confusing chart class pairs. Motivated by the above observations, in this paper, we propose an attention and triplet loss based deep CNN framework to address the above issues. From various experimental results over four datasets, it is evident that the proposed framework can effectively handle noises in the chart and confusing chart samples, and outperforms its counterparts.

Keywords: Chart classificaiton, Chart's noise, Confusing chart class pair, Attention, Triplet loss

1 Introduction

With the increase in the presence of various chart types in scientific documents in electronic media, the development of automatic chart classification system is becoming an important task. Though the attention of the researchers on chart classification increases post 2001 [1], its importance has been realized

way back in 1981[2]. Initial studies (pre-deep learning era) on chart classification generally use traditional machine learning methods such as SVM, KNN, Decision tree, etc with handcrafted features [3–5]. However, majority of the recent studies focus on using start-of-the-art deep learning models such as VGGs, ResNets, etc. As reported in [6, 7], majority of the existing chart classification models face problem while handling **(i)Chart noise:** most of the publicly available datasets for chart classification contain samples with various types of noise such as background noise, pattern noise, composite noise, etc., and **(ii) Confusing chart class pairs:** charts of similar characteristics is also one of the major reason for chart misclassification.

To the best of our knowledge, none of the earlier studies on chart classification focus on developing methods which can handle the above two issues. Motivated by this, this paper propose an *attention* and *triplet loss* based model to address the problems of chart's noise and confusing chart class pairs. Though attention-based approaches have been extensively used to handle noises in other image classification tasks [8, 9], none of the earlier studies have investigated the effect of the attention mechanisms on handling chart's noise in the charts classification tasks. Therefore, in this paper, we investigate the effect of attentions namely Convolutional block attention module (CBAM) [10] and Squeeze and Excitation network (SE)[11] on handling chart's noise. We apply these two attention mechanisms on various CNN models (VGGs, ResNets, Inceptions, MobileNets, DenseNets, Xception). The Xception, which is one of the effective chart classification models [12],has not been properly studied with attention mechanism (except for [13]). In this paper, we propose an attention based Xception model by incorporating CBAM and SE attentions with both the residual and non-residual layers (study [13] considers attention only with the last seven residual layers of Xception). Further, this study explore triplet loss function for the first time in the domain of chart classification. As training a model using triplet loss function is one of the common approaches for the fine-grained classification [12–14], this paper investigates the effect of triplet loss function on handling confusing chart class pairs classification. Triplet loss learning method has became a popular approach after the proposal of Facenet [15], created by Google. The goal of the triplet loss is to build triplets *(anchor, positive, negative)* consisting of an anchor image, a positive image (which is similar to the anchor image), and a negative image (which is dissimilar to the anchor image). Focusing on elongating the distances between confusing samples, we develop a strategy to form the triplets considering only confusing samples.

The rest of the paper is organised as follows. Section 2 presents the background and related studies, where we discussed some selected existing studies of chart classification and two issues: chart noise, and confusing chart pairs, which are reported in our earlier study [7]. In section 3, we have presented our proposed attention-based Xception models and a framework where we talked about training the attention-based models using triplet loss function. Section 4 presents the detailed experimental setups. In Section 5, we discussed the

experimental results, where we have reported performance of multiple chart classification models. In section 6, we have presented the detailed analysis of the proposed framework with respect to the noise charts, and confusing chart class pairs. Section 7 concludes our study and highlights the future directions

2 Related and Background studies

Although study on chart analysis can be traced back to year 1991[2], the first study on chart classification is reported in the year 2001[1]. A good survey on chart classification can be found in [16]. Based on the classification methods used by the existing studies, the journey of chart classification can be divided into two phases viz. pre-deep neural network phase (2001-2013) and deep neural network phase (since 2014).

In the pre-deep neural network phase, studies exploited model-based approach [17, 18] and traditional machine learning (ML) approach such as SVM [19, 20], HNN [1], KNN [20, 21], etc. In a model-based approach, each chart type has its own unique model, which is based on the chart's inherent characteristics. Graphical components of the charts, such as axes and colours[18], the layout of the chart, e.g.,rectangular (for bar charts), circular (for pie charts)[22, 23], are among these qualities. SVM, KNN and Decision tree are some well-known state-of-art traditional ML models. It is observed that studies in this phase consider manually extracted small sample sizes and the small number of chart types. One main shortcoming of this phase is most of the approaches does not generalize well. They are not effective when dealing with a large amount of data that could contain significant varieties, as in the chart image dataset[24].

LeNet, AlexNeT, VGG-16,19, Inception V3 and Inception-V4 are some of the CNN-based models that have been exploited in the current deep neural network (DNN) phase, along with Inception-ResNet-V2, MobileNet-V1 and MobileNet-V2. Xception is another CNN-based model that has been exploited in the current DNN phase. Without explicitly extracting features, models take raw images at this phase. Although the authors have investigated models with and without feature extraction in work such as [25], they have also commented on the relevance of feature selection approaches. Study [26] developed a model for learning the characteristics of both different and comparable regions at the same time. An enhanced loss function is utilised to train the model, which is fused with a structural variation-aware dissimilarity index and regularisation parameters to make it more prone to dissimilar areas. In this era, the size of the dataset is one of the obstacles. Most datasets with real chart images are quite small in size. For this reason, the large-scale synthetically created datasets have been examined in a number of research [27, 28]. However, model trained on synthetic dataset fails to perform well in real chart images because real chart images often contain noises as compared to the synthetic images. To address the lack of real chart samples, study [29] proposed a chart classification model

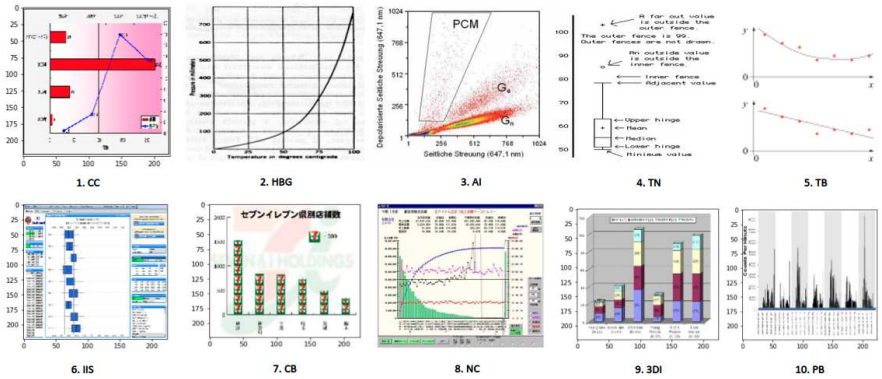


Fig. 1 Ten types of chart noise: 1. Composite Chart (CC), 2. Hard BackgroundGgrid (HBG), 3. Additional Information (AI), 4. Text Noise (Text Noise), 5. Transparent Background (TB), 6. Improper Image Screenshot (IIS), 7. Complex Background (CB), 8. Numerous Component (NC), 9. 3D Images (3DI), 10. Patterned Background (PB)

using Siamese CNN [30]. The Siamese CNN is a network architecture built using two or more identical (twin) networks, in which they used MobiletNet.

2.1 Effect of Noisy chart samples, and confusing chart class pairs

Apart from the image quality, and image noises, the performance of a chart classification model depends on other factors such as noisy chart samples, and confusing chart pairs. Our earlier paper [7] discovered ten types of chart's noise, and 13 confusing chart class pairs. As stated earlier, the main objective of this study is to develop a model which can handle noisy samples, and confusing chart samples. We provide the brief discussion of the chart's noise and confusing chart class pairs reported in the study [7].

Chart noise: Noisy chart samples are defined as the samples which are often misclassified because of some other extraneous components present in the charts. Figure 1 shows the examples of the noisy chart samples. The brief definition of the ten chart noises are given below.

1. Composite-like chart type (CC): A chart with extra component which resemble other chart type as shown in Figure 1 (1). 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 Figure 1 (2).
3. Additional Information (AI): A chart with embedded information presented in the form of shapes such as circles, etc. as shown in Figure 1 (3).
4. Text Noise (TN) : A chart with enormous amount of additional information presented in the form of text as shown in Figure 1 (4).

5. Transparent Background (TB): A chart image with complete transparent background as shown in Figure 1 (5).
6. Improper Image Screenshot (IIS):An image with some additional unrelated document regions as shown in FIgure 1 (6).
7. Complex Background (CB): A chart with distinct background watermark as shown in Figure 1 (7).
8. Numerous Components (NC): A chart with multiple chart components such as additional shapes and text as shown in Figure 1 (8).
9. 3D images (3DI): As our study do not consider 3D chart images except for surface plot, 3D image become one noise type. Even the image with little degree of the third dimension as shown in Figure 1 (9), becomes a noise.
10. Patterned Background (PB): A chart image that have background with patterns sucha as shown in Figure 1 (10). It is an area chart but because of vertical blocks in the background, it is misclassified.

Confusing chart class pair: In study [7], we have observed false classification throughout the experiments because of the similarity between two or more chart types. The chart class pair (X, Y) is considered as confusing chart class pair if $t\%$ of the sample population belonging to class X are classified as class Y. Study [7] considered t as 4% [1], and reported 13 confusing chart class pairs. They are briefly discussed below.

1. *(Area, Bar):*Area charts with multiple regions denoted by parallel or nearly parallel sharp edges are often confused with bar charts.
2. *(Area, Line):* Some area chart samples that have distinct coloured edges but fill up with shaded colour are classified as line charts.
3. *(Box, Dendrogram):* Box chart samples with huge sized multiple boxes are often confused with Dendrogram.
4. *(Bubble, Node):* The bubble charts with small size bubbles and highly visible background grid are sometimes classified as Node link.
5. *(Line, Node):* Line charts with bigger size of nodes to indicate data points are sometimes confused with Node links.
6. *(Line, Bar):* Some of the Line charts with various coloured backgrounds are sometime identified as Bar charts.
7. *(Manhattan, Scatter):* Manhattan charts with enormous amount of data with not clear edges to indicate vertical margin often classified as Scatter charts.
8. *(Node Link, Scatter):* Some samples of Node links with small nodes but low intensity links are frequently classified as Scatter charts.
9. (Pie , Venn): It is observed that Pie charts with one partition dominating other are sometimes classified as Venn. In another case, chart images with multiple pies which have minimum gap between them are also prone to be classified as Venn.

[1]Study [7] experimented the threshold t with 2%, 3%, and 4%. With 2% or 3% as t, it failed to confidently draw a similar characteristics among the misclassified samples, instead most of the mis-classification are mainly because of noisy samples, and lack of training samples.

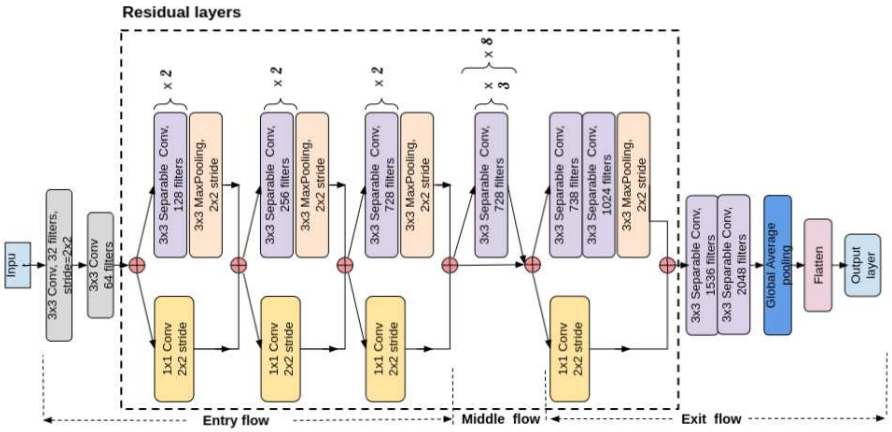


Fig. 2 Architecture of Xception: 14 modules with 12 residual layers

10. *(Radar , Venn):* Although most Radar charts have hexagonal outer layers, they may have circle or nearly circle-like outer layers. Those samples are often are misclassified as Venn diagrams.
11. *(Scatter , Line):* The scatter charts with lines are sometimes misclassified as line charts.
12. *(Table, Scatter):* Some samples of Table without borders and with a crowded data are often classified as Scatter.
13. *(Treemap, Heatmap)* : It is observed that Treemap appears to be visually similar with heatmap most of the time. The main difference is that Treemap has thick or highly visible edges for each blocks.

3 Proposed Framework

As attention mechanism is one of the popular approaches for classifying fine-grained categories, before developing our proposed framework, this study exploits various classification models with an attention mechanism. Although several studies introduced attention mechanisms into computer vision, to the best of our knowledge, this is the first of its kind to study the effect of attention in the chart classification domain. As stated in Section 1, there are various studies on developing attention-based models of several DL models. However, there is limited exploration of developing attention-based Xception models. Since it is one of the well-performed chart classification models in our earlier work[7], we proposed multiple attention-based Xception models and investigate their performance on the chart classification task. This section discusses our proposed attention-based Xception and introduces the proposed framework which exploits both attention mechanism and triplet loss function.

3.1 Attention on Xception

Xception consists of 14 modules with linear residual connections, except for the first and last modules, as shown in Figure 2. In other words, it has three main flows: entry (4 modules including initial CNN layers), middle (8 modules), and exit (2 modules). Study [13] proposed attention-based Xception for the classification of flower types. Their study incorporated the Convolutional block attention mechanism (CBAM) [10] in the last six residual layers. However, we proposed inserting an attention mechanism in the residual layers and the non-residual layers as well. Because of the places where we can insert attention mechanism, this study proposed five variants of attention-based Xception :

1. *Xception-Entry (XEN)* - The attention mechanism is inserted only in the entry filed. So, in this variant, three attention modules are inserted.
2. *Xception-Middle (XM)* - The attention mechanism is inserted only in the middle filed. So, for each eight modules (all of them have residual connection), one attention module is integrated, and hence eight attention modules are used in this variant.
3. *Xception-Exit (XEX)*- The attention mechanism is inserted only in the exit filed in which there are two modules (one with non-residual connection). So, two attention modules are integrated in this variant.
4. *Xception-Middle-Exit (XMEX)*- Considering both flows middle, and exit, there are nine residual connections and one non-residual connection. This is the combination of XM, and XEX, where attention modules are inserted to nine residual layers and one non-residual layer,
5. *Xception-All (XA)*- In this variant, all 14 modules of the Xception is followed by attention module. Hence 14 attention modules are integrated.

This study considers two well-known attention mechanisms, namely CBAM and Squeeze and Excitation network (SE)[11]. With five variants of Xception and two attention mechanisms, we have proposed ten attention-based Xception: CBAM-based XE, XM, XEX, XMEX, XA, and SE-based XE, XM, XEX, XMEX, XA.

3.2 Attention & triplet loss based Framework

The schematic architecture of the proposed framework is shown in Figure 3. The framework has multi-stage training which can be broadly divided into two blocks: **Triplet loss training**, and **Chart type classification**.

(1) Triplet loss training

The purpose of this block is to generate triplet samples from the confusing chart class pairs, and trained the model using triplet loss function. For a given confusing chart class pair (R, S), the process of this block is described below.

Triplet generation: A triplet consists of an anchor sample (a reference sample, which is a confused sample in our case), a positive sample (which is similar to the anchor), and a negative sample (which is dissimilar to the

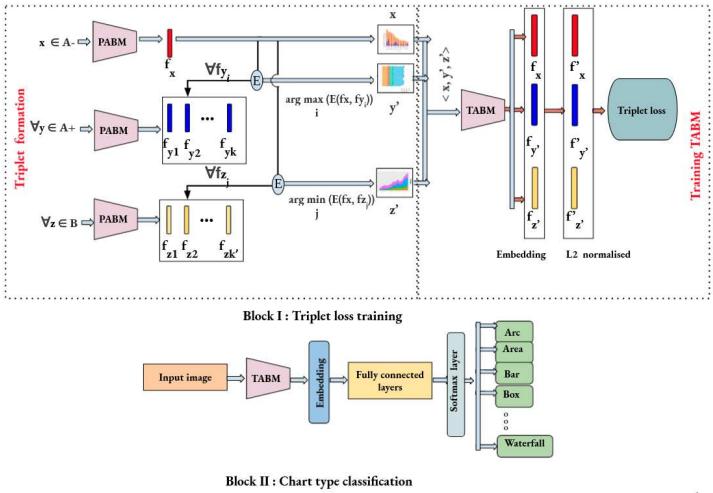


Fig. 3 Architecture of Proposed framework: Attention-triplet loss based chart classification

anchor). Let A be the training set of class R then the set of anchor samples (A^*) , and positive samples (A^+) are defined as $A^* = \{x \in A \mid p(x) \in S\}$, and $A^+ = \{y \in A \mid p(y) \in R\}$, respectively. Where p(.) is the operation that returns a class, it performs a manual checking of patterns in a sample (discussed in 2.1) and assigns a class (either R or S) to which the sample might belong. Finally, the set of negative samples is the training set of class S, which is denoted as B. The algorithm for the triplet generation from these three sets is provided in Algorithm 1.

The algorithm consists of two main steps: *Finding feature embedding*, and *Finding hard positive and negative samples*. In the prior step, we used pretrained attention-based model (PABM) to obtain the anchor's feature vector (f_x), set of positive feature vectors (denoted by F_{A+}) for all the samples in A^+, and a set of negative feature vectors (denoted by F_B) for all the samples in B. The second step performs distance calculations and comparisons. The Euclidean distance (denoted by $E(.)$) between an anchor feature vector f_x with every negative feature vectors (in F_B) and every positive feature vectors (in F_{A^+}) are calculated. We have three options to select a positive feature vector $f_{x_p} \in F_{A+}$ and a negative feature vector $f_{x_n} \in F_B$ for a given anchor feature vector f_x : **easy**, **hard**, **and semi-hard**. In an easy selection, the distance between an anchor and a negative is very large than the distance of an anchor and a positive, which can be denoted as $E(f_x, f_{x_n}) >> E(f_x, f_{x_p})$. In the hard selection, the negative feature vector is closer to an anchor than the positive feature vector which can be denoted as $E(f_x, f_{x_n}) < E(f_x, f_{x_p})$. In case of semi-hard selection, the negative is not closer to an anchor than the positive but with some margin, which can be denoted as $E(f_x, f_{x_p}) < E(f_x, f_{x_n}) < E(f_x, f_{x_p}) + margin$. As stated in study [31], hard selection yields the best performance. So, we adopt hard selection process. In the hard selection, two types of

Algorithm 1 Triplet formation

Require: Anchor sample: x, Positive sample's set A^+, and Negative sample's set: B

Ensure: $Triplet : (x, y', z')$

Finding feature embedding:

$f_x = PABM(x), x \in A^*$

$F_{A^+} = \{f_y \mid f_y = PABM(y), y \in A^+\}$

$F_B = \{f_z \mid f_z = PABM(z), z \in B\}$

PABM(.) is the model that is used for feature extraction. PABM stands for Pre-trained Attention based Model.

Finding the hard positive and the hard negative features:

$f_{xp} = \{f_y \mid f_y = \arg\max\ E(f_x, F_{A^+})\}$

$f_{xn} = \{f_z \mid f_z = \arg\min\ E(f_x, F_B)\}$

E(.) is the Euclidean distance between two vectors.

Returning triplet sample:

$y' \leftarrow f_{xp}$ ▷ fetching corresponding image sample $y'(\in A^+)$ of f_{xp}

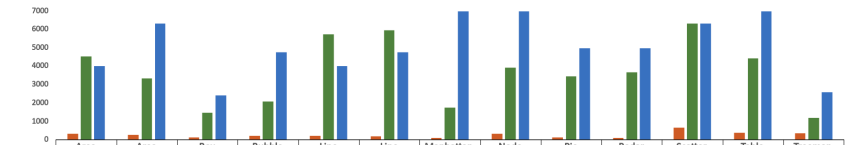
$z' \leftarrow f_{xn}$ ▷ fetching corresponding image sample $z'(\in B)$ of f_{xn}

return (x, y', z')

masks are identified: *a positive hard triplet mask* and *a negative hard triplet mask*, to select a hard positive vector f_{x_p}, and hard negative vector f_{x_n}, respectively. f_{x_p} is the one in F_{A^+} which has the highest distance from f_x, and the hard negative sample f_{x_n} is the one in F_B which has a minimum distance from f_x. Finally, for a given confusing chart class pair (R,S), we generate the triplet samples $(x \in A^-, y' \in A^+, z' \in B)$ corresponding to the above obtained triplet embeddings (f_x, f_{x_p}, f_{x_n}).

Training Triplet loss attention based model (TABM): As shown in Figure 3, once we have the triplet samples, the next step is to initialise the pre-trained weights of PABM and train with triplet loss to obtained TABM. In triplet loss function, the idea is to use three identical networks (one each for anchor, positive and negative) having the same neural net architecture and they should share underlying weight vectors to train using triplet loss. We implemented this idea using only one network and a triplet where the network expect three input samples. These three samples does not go with each other but separately. Given a triplet (x, y', z'), in order to estimate the triplet loss, we give x, y' and z' one after another to obtain f_x, $f_{y'}$, and $f_{z'}$ respectively. Once the above embedded vectors are obtained, as done in [12, 13, 32], loss function is estimated using L_2-normalization. The normalized vector $\acute{f}_x$ of f_x is estimated as

$$\acute{f}_{x_i} = \frac{f_{x_i}}{\left(\sum_j f_{x_j}^2\right)^{1/2}}$$



	Area Bar	Area Line	Box Dendrogram	Bubble Node	Line Bar	Line Node	Manhattan Scatter	Node Scatter	Pie Venn	Radar Venn	Scatter Line	Table Scatter	Treemap Heatmap
anchor	321	264	119	201	204	176	97	311	132	105	664	374	341
positive	4534	3331	1463	2075	5735	5939	1736	3930	3454	3655	6321	4412	1190
negative	4007	6313	2417	4747	4007	4747	6985	6985	4968	4968	6313	6985	2576

Fig. 4 Identified number of anchor samples, possible number of negative, and positive samples for each of the 13 confusing chart pairs.

Similarly, the normalized vectors $\acute{f}_{y'}$ and $\acute{f}_{z'}$ are also estimated.The distance between the anchor and the positive samples, and the distance between the anchor and the negative samples are estimated using softmax as given below:

$$[s_{ap}, s_{an}] \leftarrow softmax\Big(E(\acute{f}_x, \acute{f}_{y'}), E(\acute{f}_x, \acute{f}_{z'})\Big)$$

Then, the triplet loss for a given batch B is estimated as

$$Loss_t = \frac{1}{\mathrm{B}} \sum_{i=1}^{\mathrm{B}} (s_{ap} + (1 - s_{an}))$$

We optimize the loss using adam algorithm which is the combination of the 'gradient descent with momentum' algorithm and the 'RMSP' algorithm.

(2) Chart type classification

In the classification block, the pretrained triplet loss learned model (obtained in the previous block) is used as a feature generator for the final task of chart type classification as shown in the Figure 3. It is followed by three fully connected layers and then a softmax layer. The parameters we used in this block are as follows: 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.

4 Experimental setup

4.1 Dataset

We consider the dataset reported in our earlier paper [7]. It consists of 1,10,182 samples with 25 chart types. To perform triplet formation (described in Algorithm 1), we develop a sub-dataset with the samples from only confusing chart class pairs. The number of anchors, possible negative samples, and possible positive samples for each confusing chart class pair are shown in Figure 4. With 3308 number of anchors, this study obtained 3308 triplets. To study the responses of our proposed framework over other

datasets, we consider three publicly available datasets [33–35][2]. For the rest of the paper, we will be referring the datasets provided by [33],[34],[35], and [7] as **D1**, **D2**, **D3**, and **In-house**. The comparison of these four datasets is presented in Figure 5.

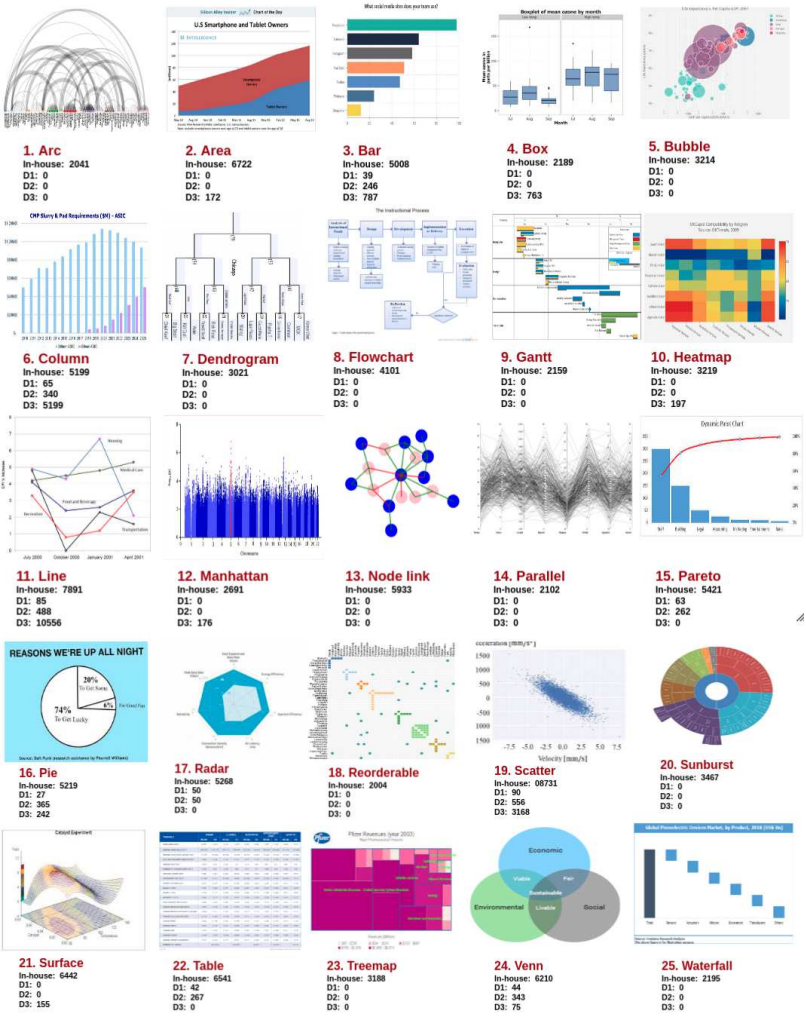


Fig. 5 Comparison of four datasets : In-house, D1, D2, and D3. D1, and D2 contributes to only ten chart types, D3 contributes to 11 chart types, and In-house contributes to 25 chart types.

[2]The dataset provided by the study [35] is the superset of the one provided by [28], which we have used for the evaluation in our earlier study [7]

Table 1 Mean accuracy of 14 baseline CNN based chart classification models under 5 fold cross-validation of In-house dataset

Testing	VGG		ResNet			Inception		Inception	MobileNet		DenseNet			Xception
Dataset	16	19	50	101	152	v3	v4	ResNet	v1	v2	121	169	201	
In-house	88.98	88.83	79.02	83.4	83.59	79.89	79.53	81.57	88.98	88.87	89.05	88.79	88	**89.91**
D1	78.68	78.248	69.842	70.286	71.05	72.378	72.55	74.58	76.55	76.38	80.63	80.06	79	**80.94**
D2	81.064	81.056	72.732	73.282	72.616	74.2	72.69	79.572	78.258	77.942	81.11	80.2	81.02	**81.39**
D3	81.756	84.06	79.248	81.836	81.552	79.826	80.004	83.12	80.8	80.28	81.05	80.23	80.67	**81.24**

4.2 Classification model

The classification models considered in this study could be broadly grouped into three: Baseline models, attention-based models, and attention-triplet loss based models. They are briefly defined below:

1. *Baseline models :* This study considered 14 DL classification models: VGG (-16, -19), ResNet (-50,-101,-152), MobileNet (-v1, -v2), Inception (-v3, -v4), Inception-ResNet, DenseNet (-121, -169, -201), and Xception. As done in our earlier study [6, 7], and various studies [5, 27, 36–38], we consider their pre-trained models trained on the ImageNet ISVCR dataset.
2. *Attention-based models* : On the top of our proposed five attention-based Xception models and attention-based Xception proposed by the study [13] (it will be referred to as X*), we considered the attention based models of VGG (-16, -19), ResNet (-50,-101,-152), MobileNet (-v1, -v2), Inception (-v3, -v4), Inception-ResNet, DenseNet (-121, -169, -201), provided by the study [10].
3. *Attention-triplet loss based:* We have considered all the attention based models used in this study to be trained on using triplet loss as shown in the proposed framework.

5 Experimental results

5.1 Baseline Models

Table 1 shows the mean accuracy under five fold cross validation of 14 CNN based models on In-house dataset, and further tested on D1, D2 and D3. The following observations may be noted.

- All the 14 models provide best results with in-house, then with D3, D2, and D1 respectively. Xception outperforms all other models.
- Apart from Xception, DenseNets, and VGGs also provide comparatively better performance than the rest.
- Among all the models, ResNet, and Inception provide the least performance for all the datasets.

Table 2 Comparison of 19 attention-based models. The number inside the bracket indicates the rise and fall of the model's accuracy from their baseline versions.

Model	Attention mechanism	Dataset			
		In-house	D1	D2	D3
VGG-16	**CBAM**	90.03 (+1.05)	80.01 (+1.33)	83.23 (+2.166)	83.9 (+2.144)
	SE	89.11 (0.13)	78.9 (+0.22)	82.09 (+1.026)	83.21 (+1.454)
VGG-19	**CBAM**	89.98 (+1.15)	81.89 (+3.642)	82.76 (+1.704)	85.98 (+1.92)
	SE	88.98 (+0.15)	80.45 (+2.202)	82.01 (+0.954)	84.05 (-0.01)
ResNet-50	**CBAM**	75.21 (-3.81)	72.11 (+2.268)	75.67 (+2.938)	82.11 (+2.862)
	SE	74.21 (-4.81)	69.21 (-0.632)	74.01 (+1.278)	86.88 (+8.632)
ResNet-101	**CBAM**	81.98 (-1.42)	72.98 (+2.694)	74.61 (+1.328)	84.09 (+2.254)
	SE	81.01 (-2.39)	74.21 (+3.924)	74.21 (+0.928)	83.51 (+1.674)
ResNet-152	**CBAM**	83.01 (-0.58)	72.01 (+0.96)	75.11 (+2.494)	83.78 (+2.228)
	SE	82.11 (-1.48)	70.18 (-0.87)	74.21 (+1.594)	82.11 (+0.558)
Inception-v3	**CBAM**	83.02 (+3.13)	73.99 (+1.612)	75.67 (+1.47)	84.21 (+4.384)
	SE	81.33 (+1.44)	73.21 (+0.832)	75.02 (+0.82)	82.11 (+2.284)
Inception-v4	**CBAM**	82.11 (+2.58)	74.02 (+1.47)	79.11 (+6.42)	83.99 (+3.986)
	SE	81.12 (+1.59)	73.31 (+0.76)	77.21 (+4.52)	83.32 (+3.316)
Inception-ResNet	**CBAM**	83.12 (+1.55)	78.21 (+3.63)	83.67 (+4.098)	83.12 (+0)
	SE	82.91 (+1.34)	76.45 (+1.87)	81.21 (+1.638)	83.05 (-0.07)
MobileNet-v1	**CBAM**	90.12 (+1.14)	79.89 (+3.34)	82.78 (+4.522)	83.12 (+2.32)
	SE	90.03 (+1.05)	77.45 (+0.9)	81.09 (+2.832)	82.98 (+2.18)
MobileNet-v2	**CBAM**	91.87 (+3)	79.98 (+3.6)	83.22 (+5.278)	82.95 (+2.67)
	SE	91.01 (+2.14)	78.67 (+2.29)	82.17 (+4.228)	82.97 (+2.69)
DenseNet-121	**CBAM**	92 (+2.19)	82.71 (+2.08)	85.21 (+3.318)	84.21 (+1.67)
	SE	90.32 (+0.51)	80.67 (+0.04)	83.9 (+2.008)	83.99 (+1.45)
DenseNet-169	**CBAM**	91.91 (+3.12)	82.98 (+1.92)	84.78 (+4.578)	84.21 (+2.55)
	SE	89.78 (+0.99)	80.33 (-0.73)	82.67 (+2.468)	83.31 (+1.65)
DenseNet-201	**CBAM**	91.01 (+1.96)	83.06 (+2.12)	85.35 (+3.958)	84.21 (+2.97)
	SE	90.31 (+1.26)	81.09 (+0.15)	83.01 (+1.618)	84.01 (+2.77)
X*	**CBAM**	91.21 (+2.1)	83.45 (+2.5)	87.12 (+5.7)	85.65 (+4.4)
	SE	91.08 (+2.0)	82.11 (+1.1)	86.79 (+5.4)	83.45 (+2.2)
XMEX	**CBAM**	**93.65 (+4.6)**	**87.89 (+6.9)**	**91.89 (+10)**	**87.21 (+5.9)**
	SE	92.89 (+3.8)	83.12 (+2.1)	91.02 (+9.6)	86.31 (+5.0)
XEN	**CBAM**	89.07 (+0.02)	82.13 (+1.1)	81.87 (+0.4)	83.11 (+1.8)
	SE	89.86 (+0.8)	81.21 (+0.2)	80.78 (-0.6)	81.78 (+0.5)
XEX	**CBAM**	91.34 (+2.2)	86.23 (+5.2)	89.89 (+8.5)	85.28 (+4.0)
	SE	92.01 (+2.9)	83.01 (+2.0)	87.36 (+5.9)	82.91 (+1.6)
XM	**CBAM**	91.06 (+2.0)	84.56 (+3.6)	84 (+2.6)	82.76 (+1.5)
	SE	91.23 (+2.1)	82.16 (+1.2)	83.11 (+1.7)	82.45 (+1.2)
XA	**CBAM**	89.01 (-0.04)	82.67 (+1.7)	81.42 (+0.03)	82.01 (+0.7)
	SE	89.99 (+0.9)	81.04 (+0.1)	81 (-0.3)	81.45 (+0.2)

5.2 Attention-based Models

Table 2 shows the performance of 19 attention-based models . The following observations may be noted.

- Most of the models experienced rise in the mean accuracy from that of their baseline version.
- There is fall in the accuracy from that of the baseline version for all the variants of ResNet in In-house dataset, and Inception-ResNet in D3 dataset.
- For all four datasets, among our proposed five variants of Xception, all the models except for XA improve their performances with an integrated attention mechanism compared to that of the baseline Xception. With the integration of the attention mechanism on all the modules of Xception, we are retraining all the modules on our dataset, which in turn has no effect of using pre-trained weights. For highly deep networks such as Xception, the size of the In-house dataset might not be enough to learn efficiently, and hence XA fails to provide promising results with attention mechanisms.
- Among all 19 models, XMEX provides the highest mean accuracy for all four datasets.
- Among the two attention mechanisms, all the models provide better results with CBAM for all four datasets.

5.3 Attention & Triplet loss based Models:

Table 3 shows the performance of 19 attention-triplet loss based models, obtained under our proposed framework. The following observations may be noted.

- With our proposed framework, all the models experienced rise in the accuracy over their respective attention-based models.
- With the proposed framework, XMEX provides better performance for all four datasets.
- Among the two attention mechanisms, our proposed framework works well with CBAM for all four datasets.

From the above observations, it is clear that our proposed framework can increase the performance of all the state-of-the-arts. By integrating only the attention mechanism, the models address the issues of noisy charts (discussed in detail in Section 6), yet the challenges provided by confusing chart class pairs remain unsolved. However, with the combination of attention and triplet loss in our proposed framework, both the issues are addressed on a large scale (discussed in detail in Section 6).

6 Discussion

From Table 2 and 3, it is observed that for all four datasets, triplet loss based CBAM-XMEX (TCBAM-XMEX) outperforms all other models in handling noisy samples and confusing class pairs, followed by triplet loss based

Table 3 Comparison of 19 attention-triplet-based models. The number inside the bracket indicates the rise and fall of the model's accuracy from that of their respective attention - based models.

Model	Attention mechanism	Dataset			
		In-house	D1	D2	D3
VGG-16	**CBAM**	95.78 (+5.75)	89.56 (+9.55)	90.23 (+7)	91.23 (+7.33)
	SE	93.78 (+4.67)	88.45 (+9.55)	90.05 (+7.96)	91 (+7.79)
VGG-19	**CBAM**	95.43 (+5.45)	93.12 (+11.23)	90.45 (+7.69)	91.34 (+5.36)
	SE	95.12 (+6.14)	92.64 (+12.19)	90.65 (+8.64)	90.12 (+6.07)
ResNet-50	**CBAM**	84.34 (+9.13)	82.11 (+10)	86.78 (+11.11)	91.09 (+8.98)
	SE	84.02 (+9.81)	80.23 (+11.02)	83.23 (+9.22)	91.07 (+3.19)
ResNet-101	**CBAM**	90.21 (+8.23)	89.45 (+16.47)	89.55 (+14.94)	91.56 (+7.47)
	SE	90.11 (+9.1)	89.65 (+15.44)	84.56 (+10.35)	89.79 (+6.28)
ResNet-152	**CBAM**	94.12 (+11.11)	89.44 (+17.43)	83.11 (+8)	90.34 (+6.56)
	SE	93.14 (+11.03)	86.32 (+16.14)	80.12 (+5.91)	90.06 (+7.95)
Inception-v3	**CBAM**	89.08 (+6.06)	82.89 (+8.9)	89.45 (+13.78)	90.21 (+6)
	SE	87.42 (+6.09)	81.09 (+7.88)	85.89 (+10.87)	90.45 (+8.34)
Inception-v4	**CBAM**	89.56 (+7.45)	88.56 (+14.54)	91.67 (+12.56)	90.12 (+6.13)
	SE	88.67 (+7.55)	87.78 (+14.47)	91.66 (+14.45)	91.56 (+8.24)
Inception-ResNet	**CBAM**	89.34 (+6.22)	87.56 (+9.35)	91.23 (+7.56)	90.45 (+7.33)
	SE	89.11 (+6.2)	85.23 (+8.78)	89.23 (+8.02)	88.79 (+5.74)
MobileNet-v1	**CBAM**	96.02 (+5.9)	89.45 (+9.56)	91.34 (+8.56)	91.34 (+8.22)
	SE	95.67 (+5.64)	89.11 (+11.66)	90.45 (+9.36)	91.45 (+8.47)
MobileNet-v2	**CBAM**	96.21 (+4.34)	88.78 (+8.8)	90.01 (+6.79)	91.34 (+8.39)
	SE	96.03 (+5.02)	86.45 (+7.78)	89.87 (+7.7)	90.11 (+7.14)
DenseNet-121	**CBAM**	97 (+5)	91.56 (+8.85)	92.35 (+7.14)	91.21 (+7)
	SE	96.43 (+6.11)	91.89 (+11.22)	92.06 (+8.16)	90.334 (+6.34)
DenseNet-169	**CBAM**	96.89 (+4.98)	93.12 (+10.14)	94.67 (+9.89)	94.67 (+10.46)
	SE	96.01 (+6.23)	93.02 (+12.69)	93.78 (+11.11)	90.05 (+6.74)
DenseNet-201	**CBAM**	96 (+4.99)	92.12(+9.06)	93.23 (+7.88)	91.67 (+7.46)
	SE	95.67 (+5.36)	91.45 (+10.36)	92.56 (+9.55)	89.35 (+5.34)
X*	**CBAM**	95.23 (+4.02)	94 (+10.55)	94.01 (+6.89)	92.01 (+6.36)
	SE	95.04 (+3.96)	93.33 (+11.22)	94 (+7.21)	92.76 (+9.31)
XMEX	**CBAM**	**98.05 (+4.01)**	**94.07 (+6.18)**	**95 (+3.11)**	**95.12 (+7.91)**
	SE	97.21 (+4.32)	93.74 (+10.62)	93 (+1.98)	93.98 (+7.67)
XEN	**CBAM**	96.01 (+6.94)	93.1 (+10.97)	91.98 (+10.11)	91.67 (+8.56)
	SE	94.11 (+4.25)	92.4 (+11.19)	91.11 (+10.33)	90.12 (+8.34)
XEX	**CBAM**	96.12 (+4.78)	91.78 (+5.55)	93.67 (+3.78)	93.11 (+7.83)
	SE	94.67 (+2.66)	90.45 (+7.44)	93.31 (+5.95)	91.02 (+8.11)
XM	**CBAM**	94.21 (+3.15)	91.54 (+6.98)	92.13 (+8.13)	90.34 (+7.58)
	SE	92.11 (+0.88)	90.32 (+8.16)	90.42 (+7.31)	89.32 (+6.87)
XA	**CBAM**	90.23 (+1.22)	89.56 (+6.89)	90.32 (+8.9)	90.56 (+8.55)
	SE	89.45 (+0.06)	88.45 (+7.41)	88.45 (+7.45)	87.45 (+6)

Table 4 Summary of four testing datasets with respect to chart noise.

Dataset	Noise type	# testing samples (TS)	# noisy samples (NS)	% noisy samples in the testing dataset (NTS)
D1	Except for TB, D1 contributes to all	320	88	27.50
D2	Except for CC, D2 contributes to all	3876	701	18.12
D3	Except for IIS, D3 contributes to all	21745	3948	18.32
In-house	HBG and PB	22036	1322	6.00

CBAM-Xception* (TCBAM-X*). So, this study presents the quantitative analysis for these models and their earlier versions before training with triplet loss: CBAM-XMEX, and CBAM-X*, and the baseline Xception. We used the Grad-CAM [39] to do the analysis. Grad-CAM is developed for a visualization approach that calculates the relevance of spatial positions in convolutional layers using gradients. Grad-output CAM's clearly displays attended regions since gradients are calculated with regard to a unique class. We attempt to look at how this network makes excellent use of features by monitoring the areas that the network considers crucial for predicting a class. The visualization results are shown in Figure 6. For all these challenging input images, as shown in the figure, Xception fails to focus on the regions of interest. With the attention mechanism, it is observed that CBAM-X*, and CBAM-XMEX started to focus on the object's regions for some samples, and classified them correctly. However, with the combination of triplet loss and attention mechanism, TCBAM-X*, and TCBAM-XMEX, the issues of most of the challenging samples are resolved with high classification confidence. We can see that the TCBAM-XMEX network's Grad-CAM masks cover the target object areas better than other approaches. It learns well to exploit information in target object regions and aggregate features from them and can decrease the distance of intra-class and increase the distance of inter-class samples. Note that target class scores also increase accordingly.

This section presents a detailed discussion of the well performed attention-based Xception model, TCBAM-XMEX against CBAM-XMEX, CBAM-X*(provided by the study [13]), TCBAM-X*, and the baseline Xception, concerning the ten noise types and 13 confusing chart pairs.

6.1 Noisy charts

Table 4 presents the detail of four testing datasets with respect to the noisy samples. Among them, In - house contributes very less noisy samples as compared to other remaining datasets. The NTS in the table shows the percentage of noisy samples for a given testing dataset. For any given testing dataset t_i, the NTS may be defined as $NTS = (\frac{NS_i}{TS_i}) \times 100$, where NS_i and TS_i are the total number of noisy samples, and total number of testing samples, respectively. In-house dataset contributes to only two types of noise viz. Hard background grid (HGB) and Patterned background (PB) by providing NTS of only 6%. Except for the noise type Transparent background (TB), the dataset

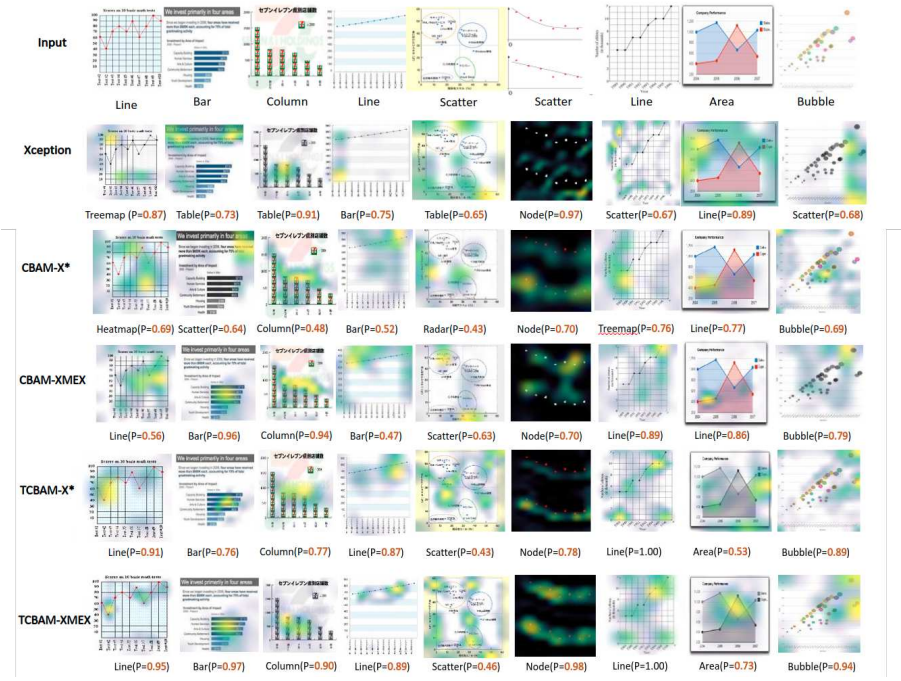


Fig. 6 Grad-CAM visualization result: Comparison of the visualization results of input image (images in first row), responses from Xception (images in second row), responses from CBAM-X*(images in third row), CBAM-XMEX (images in fourth row), TCBAM-X* (images in fifth row), and TCBAM-XMEX (images in sixth row).The grad-CAM visualization is calculated for the last convolutional outputs. P denotes the softmax score of each network for the classified class

D1 contributes to all noise types by occupying 27.50% of its 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 occupied 18.32% of its dataset by nine noise types (leaving Improper image screenshot (IIS)).

Table 5 presents the response of Xception, CBAM-X*, CBAM-XMEX, TCBAM-X*, and TCBAM-XMEX on four datasets with respect to the chart noise. The TNMC(Total noise misclassification) and TNMCO (Total noise misclassification overall) column in the table shows the misclassification because of noises among the noisy samples and over the entire dataset, respectively. TNMC is estimated as the macro average percentage of sample misclassification among the noisy samples i.e., $TNMC = \left(\sum_i^N \frac{F_i}{T_i+F_i}\right) \times 100$, where T_i and F_i are the true classification and false classification, respectively for the noise type i. Similarly, TNMCO is defined by the percentage of misclassifications from the noisy samples over the entire testing samples (TS), and estimated as $TNMCO = \left(\frac{\sum_i^N F_i}{TS}\right) \times 100$. From the table, the following points are observed:

Table 5 Performance of Xception, CBAM-X*, CBAM-XMEX, TCBAM-X*, and TCBAM - XMEX over four datasets with respect to ten types of chart noise. T and F are the true classification and false classification, respectively.

Model	Testing dataset	Noise type																				TNMC	TNMCO
		CC		HBG		AI		TN		TB		IIS		CB		NC		3DI		PB			
		T	F	T	F	T	F	T	F	T	F	T	F	T	F	T	F	T	F	T	F		
Xception	In-house	0	0	514	270	0	0	0	0	0	0	0	0	0	0	0	0	0	0	260	278	**41.61**	**3.65**
	D1	0	3	12	10	2	6	5	11	0	0	3	2	6	8	0	3	0	2	1	14	**63.63**	**17.5**
	D2	0	0	38	176	0	79	46	81	0	16	26	4	39	101	0	5	0	5	8	79	**77.6**	**14.03**
	D3	35	115	256	547	852	121	40	340	0	52	0	0	81	449	29	52	121	449	92	317	**61.8**	**11.22**
CBAM-X*	In-house	0	0	665	120	0	0	0	0	0	0	0	0	0	0	0	0	0	0	302	236	**26.49**	**1.67**
	D1	0	3	18	4	4	4	8	8	0	0	3	2	8	6	0	3	0	2	5	10	**47.72**	**13.12**
	D2	0	0	77	137	19	60	58	69	0	16	26	4	39	101	0	5	0	5	17	70	**56.05**	**9.68**
	D3	35	115	343	460	866	107	70	310	0	52	0	0	93	437	34	47	121	449	118	291	**59.03**	**10.72**
CBAM-XMEX	In-house	0	0	720	65	0	0	0	0	0	0	0	0	0	0	0	0	0	0	387	151	**16.12**	**1.02**
	D1	0	3	20	2	6	2	13	3	0	0	3	2	10	4	0	3	0	2	7	8	**32.95**	**9.06**
	D2	0	0	98	116	45	24	87	38	0	16	29	1	42	98	0	5	0	5	49	38	**54.21**	**8.02**
	D3	35	116	685	118	903	70	87	293	0	52	0	0	386	144	35	46	126	444	276	133	**17.07**	**4.01**
TCBAM-X*	In-house	0	0	669	116	0	0	0	0	0	0	0	0	0	0	0	0	0	0	311	227	**24.19**	**1.35**
	D1	0	3	20	2	5	1	11	5	0	0	3	2	8	6	0	3	0	2	8	7	**36.32**	**5.91**
	D2	0	0	77	137	19	60	58	69	0	16	26	4	39	101	0	5	0	5	17	70	**27.86**	**5.89**
	D3	35	115	412	391	898	75	90	290	0	52	0	0	111	419	37	44	121	449	210	199	**32.94**	**7.32**
TCBAM-XMEX	In-house	0	0	722	63	0	0	0	0	0	0	0	0	0	0	0	0	0	0	421	117	**13.76**	**0.8**
	D1	0	3	22	0	7	1	12	4	0	0	3	2	12	2	1	2	0	2	12	2	**19.45**	**5.26**
	D2	0	0	202	12	75	4	109	16	0	16	48	9	40	1	99	5	0	5	67	20	**12.55**	**2.27**
	D3	35	116	721	82	911	62	121	259	0	52	0	0	479	51	53	28	126	444	320	89	**13.12**	**3.21**

1. *Xception* : Except for In-house, it provides false result for more than 50% of noisy samples as given by TNMC. There are some noise types where it recognises some of their instances such as PB noise. However, in some cases, it provides inconsistent results by classifying some instances of CC noise correctly (in case of D3), and sometimes fail to recognise even a single instance of the same noise type (in case of D1). The same characteristics is observed for noise type NC, where it fails to recognise a single instance of NC noise in case of D1, and D2, but provide true classification for some instances in case of D3. As given by TNMCO, Xception's performance is highly disturbed by chart noises for all datasets specifically for D1, and D2.
2. *CBAM-X** : With this attention based Xception model, the improvement in the performance is observed. TNMC for In-house, D1, D2, and D3 are reduced to 26.49%, 47.74%, 56.05%, and 59.03%, respectively. Even though it provides inconsistent results for some noise type such as NC (like in original Xception), there are increase in the number of true classifications for other noise types. Like baseline Xception, it fails to classify a single instances of NC noise type for D1, and D2. However, it can classify correctly more number of NC noise samples from D3.
3. *CBAM-XMEX* : Our proposed CBAM-XMEX model has the same characteristics as CBAM-X*. It does not provide true results for any samples from those noise types where CBAM-X* fails to recognize even a single instance

such as CC (for D1, and D2), TB (for D1, D2, and D3), and NC (for D1, and D2). However, it provides a significant rise in the frequency of correct classifications for other noise types. It classified 386 samples from CB noise type correctly, but CBAM-X* only gets 93. Furthermore, it reduces the TNMC for In-house, D1, D2, and D3 to 16.12%, 32.95%, 54.21%, and 17.07% respectively.

4. *TCBAM-X**: This model provides promising results compared to the baseline Xception, and its version with only attention module i.e CBAM-X*. It provides rise in the number of true classification for all four datasets. It reduces TNMC to 24.19%, 36.32%, 27.86%, and 32.94% for In-house, D1, D2, and D3, respectively.
5. *TCBAM-XMEX*: Among these four models, it gives the best performance. Despite the fact that it fails to recognize a single instance of noise types TB and 3DI, the frequency of true classification of all other noise types appears to be increasing. It provides minor noise error in case of In-house. It reduces TNMC 13.76%, 19.45%, 12.55%, and 13.12% for In-house, D1, D2, and D3, respectively.

Table 6 Summary of four testing datasets with respect to confusing chart class pairs.

Dataset	Confusing chart class pairs	# testing samples (TS)	# Confusing # samples (CS)	% confusing samples in the testing dataset (CCS)
D1	Contributes only to	320	5	1.56
D1	$(Area, Bar)$	320	5	1.56
D2	Contributes to **5** confusing pairs			
D2	$(Area, Bar), (Line, Node)$	3876	133	3.43
	$(Pie, Venn), (Scatter, Line)$			
	$(Table, Scatter)$			
D3	Contributes to **6** confusing pairs	21745	1416	6.51
	$(Box, Dendrogram), (Line, Bar)$			
	$, (Line, Node), (Pie, Venn)$			
	$(Scatter, Line), (Table, Scatter)$			
In-house	Contributes to all **13** confusing pairs	22036	1448	6.57

6.2 Confusing chart class pairs

Table 6 presents the summary of four testing datasets from the view of confusing chart class pairs. It is observed from the table that In-house and D3 contributes comparatively large number of confusing samples than D1, and D2. A pair (x, y) in the table denotes misclassification of the input samples from the chart type $'x'$ as chart type $'y'$. So, the five samples of D1 that contributes to the pair (Area, Bar) are five area chart samples (with the particular characteristics mentioned in 2.1) which gets classified as Bar chart type. The CCS in the table shows the percentage of confusing chart samples for a given testing dataset. For any given testing dataset t_i, CCS may be defined as $NCS = (\frac{CS_i}{TS_i}) \times 100$, where CS_i and TS_i are the total number of confusing samples, and total number of testing samples, respectively. As observed in the table, In-house contributes to all 13 confusing chart class misclassification

Table 7 Performance of Xception, CBAM-X*, CBAM-XMEX, TCBAM - X*, and TCBAM - XMEX over four datasets with respect to confusing chart class pairs. T and F are true classification and false classifications, respectively.

Model	Testing Dataset	Result	Confusing chart pairs													TCMC	TCMCO
			(Area,) (Bar)	(Area,) (Line)	(Box,) (Dendro)	(Bubble) (Node)	(Line,) (Bar)	(Line,) (Node)	(Manhat.,) (Scatter)	(Node,) (Scatter)	(Pie,Venn)	(Radar,) (Venn)	(Scatter,) (Line)	(Table,) (Scatter)	(Treemap,) (Heatmap)		
Xception	In-house	T	21	16	11	20	11	9	5	21	12	14	21	19	11	**79.26**	**5.21**
		F	125	132	31	37	198	76	37	11	87	56	297	76	94		
	D1	T	1	0	-	-	0	0	-	-	0	0	0	0	-	**80**	**1.25**
		F	4	0	-	-	0	0	-	-	0	0	0	0	-		
	D2	T	1	0	-	-	0	2	-	-	4	0	9	1	-	**87.21**	**3**
		F	21	0	-	-	0	20	-	-	6	0	45	24	-		
	D3	T	-	-	0	-	59	65	-	-	32	0	56	55	-	**94.3**	**6.14**
		F	-	-	267	-	211	201	-	-	142	0	132	196	-		
CBAM-X*	In-house	T	28	16	11	20	11	9	5	21	12	14	21	19	11	**78.83**	**5.16**
		F	118	132	31	37	198	76	37	11	87	56	297	76	94		
	D1	T	1	0	-	-	0	0	-	-	0	0	0	0	-	**80**	**1.25**
		F	4	0	-	-	0	0	-	-	0	0	0	0	-		
	D2	T	1	0	-	-	0	2	-	-	4	0	9	1	-	**87.21**	**3**
		F	21	0	-	-	0	20	-	-	6	0	45	24	-		
	D3	T	-	-	0	-	59	65	-	-	32	0	56	55	-	**94.3**	**6.14**
		F	-	-	267	-	211	201	-	-	142	0	132	196	-		
CBAM-XMEX	In-house	T	45	16	11	20	11	9	5	21	12	14	34	19	11	**72.42**	**4.76**
		F	101	132	31	37	198	76	37	11	87	56	284	76	94		
	D1	T	1	0	-	-	0	0	-	-	0	0	0	0	-	**80**	**1.25**
		F	4	0	-	-	0	0	-	-	0	0	0	0	-		
	D2	T	1	0	-	-	0	2	-	-	4	0	21	1	-	**78.19**	**2.75**
		F	21	0	-	-	0	20	-	-	6	0	33	24	-		
	D3	T	-	-	0	-	59	65	-	-	32	0	92	55	-	**91.86**	**5.98**
		F	-	-	267	-	211	201	-	-	142	0	96	196	-		
TCBAM-X*	In-house	T	137	116	14	28	178	77	24	21	19	21	109	32	100	**36.21**	**2.38**
		F	9	32	28	29	31	11	18	11	13	49	209	63	5		
	D1	T	4	0	-	-	0	0	-	-	0	0	0	0	-	**2**	**1**
		F	1	0	-	-	0	0	-	-	0	0	0	0	-		
	D2	T	19	0	-	-	0	17	-	-	7	0	40	25	-	**18**	**0.6**
		F	3	0	-	-	0	4	-	-	3	0	14	0	-		
	D3	T	-	-	201	-	245	212	-	-	109	0	127	217	-	**21.53**	**1.4**
		F	-	-	66	-	25	54	-	-	65	0	61	34	-		
TCBAM-XMEX	In-house	T	144	148	38	56	209	85	40	32	98	66	315	93	103	**1.45**	**0.09**
		F	2	0	4	1	0	0	2	0	1	4	3	2	2		
	D1	T	5	0	-	-	0	0	-	-	0	0	0	0	-	**0**	**0**
		F	0	0	-	-	0	0	-	-	0	0	0	0	-		
	D2	T	22	0	-	-	0	17	-	-	7	0	40	25	-	**15.78**	**0.54**
		F	0	0	-	-	0	4	-	-	3	0	14	0	-		
	D3	T	-	-	266	-	265	261	-	-	172	0	179	248	-	**1.76**	**0.11**
		F	-	-	1	-	5	5	-	-	2	0	9	3	-		

overall) in the table show the overall error contributions among the confusing samples and the entire dataset, respectively. TCMC is estimated as the macro average percentage of sample misclassification between the confusing chart class pairs i.e.,the percentage of misclassifications from the confusing pairs over the entier testing samples (TS), and estimated as $TCMC = \left(\sum_{(x,y)} \frac{F_{(x,y)}}{T_{(x,y)}+F_{(x,y)}}\right) \times 100$, where (x, y) is a confusing pair, $T_{(x,y)}$ and $F_{(x,y)}$ are the true and false classifications, respectively, for (x, y) pair. Similarly, TCMCO is defined as $TCMCO = \left(\frac{\sum_{(x,y)} F_{(x,y)}}{TS}\right) \times 100$.
pairs providing CCS of 6.57%, and the lowest CCS of 1.56% comes from D1 that contributes to only one confusing chart class pair.

Table 7 presents the performances of five models: Xception, CBAM-X*, CBAM-XMEX, TCBAM-X*, and TCBAM-XMEX over four datasets from the perspective of identified confusing chart class pairs. The TCMC(Total Confusing pairs misclassification) and TCMCO (Total Confusing pairs From the table, the following points are observed:

1. *Xception* : It fails to provide promising results for all four datasets. Even though it manages to provide correct classification for some instances of all the confusing chart class pairs for all datastes, it fails to recognize a single instance of the confusing class pair $(Box, Dendrogram)$ for the dataset D3. It is further observed that Xception provides TCMC of 94.3% and TCMCO of 6.14% for the dataset D3, which is larger than In-house.
2. *CBAM-X** : This attention based Xception model fails to provide significant performance. It offered some rise in the number of true classification for the confusing class pair $(Area, Bar)$ for In-house. However, it yields same characteristics as baseline Xception for other remaining confusing chart class pairs, for all datsets. Hence ,it fails to deliver significant drop in the rate of misclassification with respect to the confusing samples. It fails to reduce the TCMC for all datasets except for In-house (reduced by only 0.43%).
3. *CBAM-XMEX* : Our proposed CBAM-based Xception, CBAM-XMEX, able to provide marginally better performance than Xception, and CBAM-X*. There is a rise in the frequency of correct classification for the confusing chart class pair $(Area, Bar)$ for In-house. Unlike, CBAM-X*. it produces true classification for some of the instances of the pair $(Scatter, line)$ for D2, D3, and In-house. Furthermore, it reduces TCMC for In-house, D1, D2, and D3, to 72.42%, 80%, 78.19%, and 91.86%, respectively
4. *TCBAM-X** : As compared to Xception, and CBAM-X*, it provides profound performance not only with noise charts, but also with confusing chart class pairs. It classified all instances of the pair $(Table, Scatter)$ for D2 correctly, and provides better results for all four datasets. Unlike the previous model, it able to reduce the misclassification error (because of confusing samples) for all four datasets. The TCMC for In-house, D1, D2, and D3, are reduced to 36.21%, 2%, 18%, and 21.53%, respectively.

5. *TCBAM-XMEX* : This model has a significant performance as compared to the above three models. For all four datasets, it increases the number of true classifications of all confusing chart class pairs. Multiple confusing chart class pairs with 100% correct classifications are reported. It's also worth noting that all the confusing samples of D1 are correctly categorised, implying that the contribution to the misclassification due to confused samples is zero. Considering other remaining dataets, In-house, D2, and D3, the TCMC are reduced to 1.451%, 15.78% and 1.76%, respectively.

From the above discussion, it is observed that the attention-based Xception models are capable of addressing some of the challenges presented by noisy data. However, attention-triplet based Xception models are able to address both the issues of chart classification: chart noise, and confusing chart class pairs.

7 Conclusion and Future work

This research offered a framework for dealing with two major chart categorization issues:*chart noise* and *confusing class chart pairs*. This is the first study of its kind to tackle these difficult challenging issues in developing the chart classification models. For the first time in the domain of chart classification, the proposed framework used two attention mechanisms, CBAM and SE, as well as the triplet loss function. In addition, the developed framework employed the offline model for producing triplet samples from confusing chart pairs. This study conducted comprehensive trials with multiple state-of-the-art models to evaluate its efficacy, confirming that our proposed framework outperforms all baselines on four different datasets. In addition, we visualize how it infers an input image precisely. Interestingly, we discovered that our framework focuses appropriately on the target object. In a nutshell, the attention mechanism deals with the majority of chart noise, while the triplet loss function tackles the problem of confusing chart pairs. We intend to expand the number of chart kinds in the future.

References

[1] Zhou, Y., Tan, C.L.: Learning-based scientific chart recognition. In: IAPR GREC2001, pp. 482–492 (2001)

[2] 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)

[3] Shao, M., Futrelle, R.P.: Recognition and classification of figures in pdf documents. In: Liu, W., Lladós, J. (eds.) GREC, pp. 231–242. Springer, Berlin, Heidelberg (2006)

[4] Prasad, V.S.N., Siddiquie, B., Golbeck, J., Davis, L.S.: Classifying computer generated charts. In: IWCBMI, pp. 85–92 (2007)

[5] Jung, D., Kim, W., Song, H., Hwang, J.-i., Lee, B., Kim, B., Seo, J.: ChartSense: Interactive Data Extraction from Chart Images, pp. 6706–6717. ACM, New York, NY, USA (2017)

[6] Thiyam, J., Singh, S.R., Bora, P.K.: Challenges in chart image classification: A comparative study of different deep learning methods. In: ACM Symposium on DocEng. DocEng '21. ACM, NY, USA (2021)

[7] Thiyam, J., Singh, S.R., Bora, P.K.: Chart Classification: An Empirical Comparative Study of Different Learning Models. ACM, NY, USA (2021)

[8] Wang, S.-H., Fernandes, S., Zhu, Z., Zhang, Y.-D.: Avnc: Attention-based vgg-style network for covid-19 diagnosis by cbam. IEEE Sensors, 1–1 (2021)

[9] Wang, S.-H., Zhou, Q., Yang, M., Zhang, Y.-D.: Advian: Alzheimer's disease vgg-inspired attention network based on convolutional block attention module and multiple way data augmentation. Frontiers in Aging Neuroscience **13**, 313 (2021)

[10] Woo, S., Park, J., Lee, J.-Y., Kweon, I.S.: Cbam: Convolutional block attention module. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) ECCV, pp. 3–19. Springer, Cham (2018)

[11] Hu, J., Shen, L., Sun, G.: Squeeze-and-excitation networks. In: IEEE/CVF, pp. 7132–7141 (2018)

[12] Wang, J., Li, Y., Miao, Z., Zhao, X., Rui, Z.: Multi-level metric learning network for fine-grained classification. IEEE Access **7**, 166390–166397 (2019)

[13] Zhang, M., Su, H., Wen, J.: Classification of flower image based on attention mechanism and multi-loss attention network. Computer Communications **179**, 307–317 (2021)

[14] Cui, Y., Zhou, F., Lin, Y., Belongie, S.J.: Fine-grained categorization and dataset bootstrapping using deep metric learning with humans in the loop. IEEE CVPR, 1153–1162 (2016)

[15] Schroff, F., Kalenichenko, D., Philbin, J.: Facenet: A unified embedding for face recognition and clustering. CoRR **abs/1503.03832** (2015)

[16] Davila, K., Setlur, S., Doermann, D., Bhargava, U.K., Govindaraju, V.: Chart mining: A survey of methods for automated chart analysis. IEEE

TPAMI, 1–1 (2020)

[17] Huang, W., Tan, C.L.: A system for understanding imaged infographics and its applications. In: ACM Symposium on DocEng. DocEng '07, pp. 9–18. ACM, New York, NY, USA (2007)

[18] Huang, W., Zong, S., Tan, C.L.: Chart image classification using multiple-instance learning. In: IEEE WACV, pp. 27–27 (2007)

[19] Futrelle, R.P., Shao, M., Cieslik, C., Grimes, A.E.: Extraction, layout analysis and classification of diagrams in pdf documents. In: ICDAR. ICDAR '03, p. 1007. IEEE Computer Society, USA (2003)

[20] 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, pp. 2865–2868 (2012)

[21] Karthikeyani, V., Nagarajan, S.: Machine learning classification algorithms to recognize chart types in portable document format (pdf) files. IJCA **39**, 1–5 (2012)

[22] Yokokura, W.T. Naoko: Layout-based approach for extracting constructive elements of bar-charts. In: Tombre, C.A.K. Karl (ed.) GRAS, pp. 163–174. Springer, Berlin, Heidelberg (1998)

[23] Mishchenko, A., Vassilieva, N.: Model-based recognition and extraction of information from chart images. In: JMPT, vol. 2, pp. 76–89 (2011)

[24] Amara, J., Kaur, P., Owonibi, M., Bouaziz, B.: Convolutional neural network based chart image classification. (2017)

[25] 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 Processing **124** (2015)

[26] Mishra, P., Kumar, S., Chaube, M.K.: Dissimilarity-based regularized learning of charts. ACM TOMM **17**(4) (2021)

[27] 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: IJCNN, pp. 1–8 (2018)

[28] 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: ICDAR, pp. 1594–1599 (2019)

[29] Bajić, F., Job, J.: Chart classification using siamese CNN. Journal of

Imaging **7**(11), 220 (2021)

[30] Koch, G., Zemel, R., Salakhutdinov, R.: Siamese neural networks for one-shot image recognition. (2015)

[31] Hermans, A., Beyer, L., Leibe, B.: In Defense of the Triplet Loss for Person Re-Identification. arXiv (2017)

[32] Cui, Y., Zhou, F., Lin, Y., Belongie, S.: Fine-grained categorization and dataset bootstrapping using deep metric learning with humans in the loop, pp. 1153–1162 (2016)

[33] Savva, M., Kong, N., Chhajta, A., Fei-Fei, L., Agrawala, M., Heer, J.: Revision: Automated classification, analysis and redesign of chart images. UIST '11. ACM, NY, USA (2011)

[34] Chagas, P., Freitas, A., Daisuke, R., Miranda, B., Araújo, T.D.O.D., Santos, C., Meiguins, B., Morais, J.M.D.: Architecture proposal for data extraction of chart images using convolutional neural network. In: 2017 IV, pp. 318–323 (2017)

[35] Davila, K., Tensmeyer, C., Shekhar, S., Singh, H., Setlur, S., Govindaraju, V.: Icpr 2020. In: Del Bimbo, A., Cucchiara, R., Sclaroff, S., Farinella, G.M., Mei, T., Bertini, M., Escalante, H.J., Vezzani, R. (eds.) ICPR, pp. 361–380. Springer, Cham (2021)

[36] Siegel, N., Horvitz, Z., Levin, R., Divvala, S., Farhadi, A.: Figureseer: Parsing result-figures in research papers, vol. 9911, pp. 664–680 (2016)

[37] Poco, J., Heer, J.: Reverse-engineering visualizations: Recovering visual encodings from chart images. Computer Graphics Forum **36**, 353–363 (2017)

[38] Balaji, A., Ramanathan, T., Sonathi, V.: Chart-text: A fully automated chart image descriptor. CVPR (2018)

[39] 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: IEEE ICCV, pp. 618–626 (2017)