Unsupervised classification of multi-class chart images: a comparison of customized CNNs and transfer learning techniques

Article *in* PeerJ Computer Science · October 2025

DOI: 10.7717/peerj-cs.3148

Syed Muhammad Hassan
Sindh Madressatul Islam University

Unsupervised classification of multi-class chart images: a comparison of customized CNNs and transfer learning techniques

Syed Muhammad Hassan Zaidi[1], Abdul Hafeez Khan[2], Syed Azeem Inam[1], Abdullah Ayub Khan[3], Jamil Abedalrahim Jamil Alsayaydeh[4], Ahmad Ali AlZubi[5], Benny Ogunshola[6] and Safarudin Gazali Herawan[7]

[1] Department of Artificial Intelligence and Mathematical Sciences, Sindh Madressatul Islam University Karachi, Karachi, Pakistan
[2] Department of Software Engineering, Sindh Madressatul Islam University Karachi, Pakistan
[3] Department of Computer Science, Bahria University Karachi Campus, Karachi, Pakistan
[4] Department of Engineering Technology, Fakulti Teknologi Dan Kejuruteraan Elektronik Dan Komputer (FTKEK), Universiti Teknikal Malaysia Melaka (UTeM), melaka, Malaysia, Melaka, Malaysia
[5] Department of Computer Science and Engineering, College of Applied Studies, King Saud University, Riyadh, Saudi Arabia
[6] Department of Electrical and Telecommunications Engineering, Frontier institute of technology, Glenroy, Australia
[7] Industrial Engineering Department, Faculty of Engineering, Bina Nusantara University, Jakarta, Indonesia

ABSTRACT

Visualization through charts has become a crucial method for presenting complex data in a clear and structured format, surpassing traditional representation techniques in both clarity and interpretability. However, the automatic classification of chart images remains a significant challenge, particularly in the absence of labeled data. This study investigates the unsupervised classification of chart images using a combination of deep learning and clustering techniques. The primary objective is to develop a method capable of automatically categorizing four common chart types, histogram, bar, line, and pie charts without relying on annotated datasets. These chart types are frequently used in critical domains such as financial, socio-economic, and political analysis. To achieve this, a pre-trained Visual Geometry Group 16 (VGG16) model is employed for feature extraction, followed by principal component analysis (PCA) for dimensionality reduction. The k-means clustering algorithm is then applied to group visually similar chart images. For classification, three pre-trained models, Residual Network 50 (ResNet50), Residual Network 50 Version 2 (ResNet50V2), and Densely Connected Convolutional Network (DenseNet) are evaluated alongside a customized convolutional neural network (CNN) using the ChartVQA dataset. The models achieve classification accuracy of 85.40%, 80.76%, 82.46%, and 90.71%, respectively. By integrating transfer learning with clustering and CNN-based architecture, this study presents a robust and scalable framework for unsupervised chart image classification.

Subjects Algorithms and Analysis of Algorithms, Artificial Intelligence, Computer Vision, Data Mining and Machine Learning, Neural Networks

Keywords Charts, Image, Vgg16, Restnet50, Transfer learning, K-means clustering, Unsupervised, Machine learning, Image classification, Convolutional neural network

Submitted 20 March 2025
Accepted 31 July 2025
Published 7 October 2025

Corresponding authors
Syed Muhammad Hassan Zaidi, m.hassan@smiu.edu.pk
Jamil Abedalrahim Jamil Alsayaydeh, jamil@utem.edu.my

Academic editor
Consolato Sergi

Additional Information and Declarations can be found on page 21

DOI **10.7717/peerj-cs.3148**

How to cite this article Zaidi SMH, Khan AH, Inam SA, Khan AA, Alsayaydeh JAJ, AlZubi AA, Ogunshola B, Herawan SG. 2025. Unsupervised classification of multi-class chart images: a comparison of customized CNNs and transfer learning techniques. **PeerJ Comput. Sci. 11:e3148 DOI 10.7717/peerj-cs.3148**

INTRODUCTION

Visualization through chart images like bar charts, pie charts, scatter plots and line charts have grown in popularity for analyzing data and making proper decisions along with knowledge understanding of large amount of data, however it is very hard to recognize and identify true meanings due to the complex structure of charts. Data visualizations assist in identifying trends within data, simplify the process of pinpointing areas that require improvement, highlight correlations and important details, aid in creating analytical reports, and enhance the visual appeal of the information (*Bajic, Job & Nenadić, 2020*). Individuals with visual impairments or blindness face even greater challenges in accessing information. These individuals utilize screen readers to navigate documents and gather information. However, when encountering visualizations, screen readers can, at best, read the accompanying title located below the visualization itself. The text alone is insufficient for users to comprehend the visualization's intended representation, rarely providing details of such types of the graphic employed (*Elzer et al., 2007*; *Jung et al., 2017*; *Alsayaydeh et al., 2022*).

In today's data-centric world, charts and graphs play a vital role in transforming complex data into clear and actionable insights. They are commonly used in domains such as finance, healthcare, scientific publishing, and socio-economic policy to support decision-making and communicate trends effectively. Among the most frequently used chart types are bar charts, line graphs, pie charts, and histograms, which are particularly well-suited for visualizing patterns across categories, time, or distributions (*Tufte & Graves-Morris, 1983*). As data visualizations become increasingly central to analytics workflows, the ability to automatically classify chart images has become essential for tasks such as document indexing, accessibility, and automated reporting (*Hossain et al., 2021*).

Traditional features extraction methods used for handling images classification task based on bounding boxes (*Scharr, 2004*), edge detection (*Hastie et al., 2005*), and extracting pixel intensity (*Pereira et al., 2019*) but it struggles on complex and diversify real world visualization data images. Several supervised machine learning algorithms such as decision trees (*Quinlan, 1986*), support vector machines (*Chapelle, Haffner & Vapnik, 1999*), random forests (*Fawagreh, Gaber & Elyan, 2014*), and neural networks algorithms (*LeCun, Bengio & Hinton, 2015*; *Jiang et al., 2021*; *Wu et al., 2021*) achieve high accuracy for image classification. When ground truth labels for images are missing it is difficult to generate annotated data.

Unsupervised learning has been introduced to reduce reliance on labeled data. When the labeling is impractical or costly, *Zhou et al. (2017)* and *Yang et al. (2022)* discussed unsupervised learning importance. Without the need to manual annotation, interpret and group visual data to identify patterns. Pre-trained models are the most powerful approach to improve performance and reduction in training time. To reduce the computational cost and data flexibility (*Simonyan & Zisserman, 2014*; *Ng & Jordan, 2001*; *Weiss, Khoshgoftaar & Wang, 2016*), VGG16 architecture is fine-tuned model and well known for their simplicity and depth to enhance classification accuracy. Existing chart images classification models (*Khang et al., 2020*) based on supervised technique and focus on simple data type.

By focusing on these issues, this study uses novel customized convolutional neural network (CNN) architecture to classify multi-class unsupervised chart images. VGG16 pre-trained model is used for features extraction, followed by principal component analysis (PCA) for dimensionality reduction. To group visually similar charts K-means clustering algorithm has been applied on ChartVQA dataset having bar, line, pie, and histogram charts type from real-world various domains.

To benchmark performance, the proposed model is evaluated against three state-of-the-art transfer learning models Residual Network 50 (ResNet50), ResNet50V2, and Densely Connected Convolutional Network (DenseNet). The customized CNN achieves the highest classification accuracy (90.71%), validating the effectiveness of the proposed unsupervised approach in real-world scenarios.

The rest of the article is organized as follows: 'Related Works' reviews related work on chart classification and visual feature extraction. 'Materials and Methods' describes the proposed methodology, including feature extraction, clustering, and classification. 'Results and Discussion' presents experimental results and discussion. 'Conclusions' concludes the article and outlines directions for future research.

RELATED WORKS

The classification of charts and graphs is a critical task in the fields of data analysis and visualization. With the advent of the machine and deep learning (ML and DL), CNNs and RNNs, significant advancements have been made in automating this task. These literature reviews strive to present detailed assessment of various strategies employed in chart classification.

Classical machine learning techniques

Machine learning algorithms such as support vector machines (SVM) and random forests theoretical concepts were discussed by Later, *Cortes & Vapnik (1995)* and *Breiman (2001)*. *Lavangnananda & Piyatumrong (2005)* used Pearson correlation and slope for between different features on noisy charts for classification. Machine learning algorithms are good on small and simple data while deep learning perform better performance on complex structure proposed by *Wang, Fan & Wang (2020)*. On chart datasets 91% accuracy was achieved by *Del Bimbo, Gemelli & Marinai (2022)*, *Alsayaydeh et al. (2023)* using SVM and random forest models. *Al-Kady et al. (2023)* and *Kim, Lell & Scherp (2024)* perform recent review based on traditional classifier and handcrafted features.

Deep learning with CNNs

Chagas et al. (2018), *Shaha & Pawar (2018)*, *Kosemen & Birant (2020)*, and *Shaheen et al. (2019)* employed CNNs for various chart type classification tasks, achieving improved response times and accuracy values of up to 97%. *Araujo et al. (2020)* evaluated the performance of different pretrained models, including ResNet, MobileNet, and Xception, across 13 chart classes. *Bajić & Job (2021)* developed a Siamese CNN model to improve classification performance and accuracy to tackle challenges related to diverse chart types and low-data scenarios. For specialized scenarios, such as scatter plot classification for the visually impaired, *Birant et al. (2022)* also employed CNNs effectively. *Tran et al. (2022)* lead a systematic investigation of machine and deep learning models for chart classification. They implement standard ML techniques that rely on manually engineered features and compare these with DL models that automatically learn features from raw image data. CNNs achieve an impressive accuracy rate of 95%, compared to SVMs and random forests, which hover around 85% to 88%. More recently, *Ren et al. (2024)* applied deep learning with projection histograms to 3D models, showcasing the flexibility of these methods for complex visual data beyond 2D.

Hybrid deep learning models (CNN + RNN)

To address the temporal or sequential nature of some visual data, researchers have combined CNNs with recurrent neural networks (RNNs). *Bauerle et al. (2022)* and *Yang, Jiang & Guo (2019)* used CNN-RNN hybrids, particularly LSTM-based architectures, for animated and time-series chart classification. These models effectively captured both spatial features and temporal dependencies. *Wang (2024)* achieved an accuracy of 96% using a CNN-LSTM architecture, outperforming standalone CNNs and RNNs.

Transfer learning approaches

Transfer learning has enabled models pre-trained on large datasets like ImageNet to be adapted for chart classification, improving performance while reducing training time. *Olivas et al. (2009)* proposed an extensive taxonomy of unsupervised methods to address scenarios where labeled data is unavailable. *Mnih et al. (2015)* developed Deep Q-Networks using reinforcement learning, achieving human-level performance and demonstrating strong capabilities in sequential decision-making tasks. Pretrained models such as VGG16, ResNet, and DenseNet have been fine-tuned to handle unstructured inputs, as demonstrated by *Shaha & Pawar (2018)*, *Mishra, Kumar & Chaube (2022)*, and *Attique et al. (2022)*. *Bansal et al. (2021)* demonstrated the efficiency of using transfer learning with the VGG19 model for chart classification problems.

Generative and data augmentation techniques

To extend chart dataset synthetically (*Kusiak, 2019*) was used by generative adversarial networks (GANs) which produce 6% improved classification accuracy in CNN. Excel data with rotation, noise and scaling, data augmentation techniques used by *Del Bimbo, Gemelli & Marinai (2022)* resulting in 88% to 94% improvement in CNN.

Accessibility and assistive systems

Without visual input, *Singh et al. (2023)* explored and interpret charts visualizations in different ways. For visually impaired individuals, *Al-Kady et al. (2023)* proposed Chartlytics that generate audible output and description in text for chart images. For chart classification. MobileNet is used with 99.42% accuracy, YOLO is used and evaluated with 0.99 mean average precision (mAP) at 50 score to component detection and 89.2% to 98.5% accuracy are achieved for rule-based data extraction.

Table 1 Charts images classification taxonomy.

Reference	Year	Model(s)/ Technique used	Dataset	Chart type	Key findings	Identified gaps/ Limitations
Del Bimbo, Gemelli & Marinai (2022)	2022	SVM, CNN, Random forest	Chart image dataset	Bar, Line, Pie, Scatter, Histogram, Area, Box, Heatmap, Bubble, Radar, Tree map, Gantt	CNN outperformed SVM (88%) and RF (91%) with 95% accuracy	No statistical test on results; limited insight into chart-specific challenges
Bajić & Job (2021)	2021	KNN	Scatterplot dataset	Scatter plots	KNN achieved 85% using point density and distribution features	Focus limited to scatter plots; shallow learning model
Kosemen & Birant (2020)	2020	CNN	Line chart dataset	Line charts	CNN achieved 93.75% on real-world line charts; supports trend classification	Only line charts; no transfer learning or augmentation used
Chagas et al. (2018)	2018	Xception, VGG19, ResNet152, MobileNet	Chart image dataset	13 chart types	Xception achieved highest accuracy >95% among all models	Dataset not publicly validated; lacks cross-domain benchmarking
Mishra, Kumar & Chaube (2022)	2022	SVM, KNN, ResNet	Chart recognition dataset	Mixed (unspecified)	ResNet reached 90% accuracy; higher than SVM (85%) and KNN (78%)	Dataset diversity unclear; lacks error analysis
Thiyam, Singh & Bora (2023)	2023	ResNet-18, SVM	Scientific chart dataset	Bar, Line, Pie	Achieved 93% using hybrid vision-text model (RMSProp, ReLU)	Misclassification in similar chart types; only 3 chart classes
Kim, Lell & Scherp (2024)	2024	Multimodal Transformers, BERT	Scientific chart dataset	Text-role charts	Achieved 93% by combining visual + textual data	Focused on text role, not full chart classification
Chartlytics/*Al-Kady et al. (2023)*	2023	MobileNet, YOLO, Rule-based	Chartlytics dataset	Mixed	Achieved 99.42% in chart classification using MobileNet	Focuses more on component detection than classification
Araujo et al. (2020)	2020	VGG19, ResNet152, MobileNet, Xception	Web-collected	13 chart types	Xception scored highest among all models	Lacks detailed error analysis or model efficiency benchmarks

Healthcare, finance, and geotechnical engineering applications

Machine learning algorithms used in medical, engineering and finance domain for chart analysis (*Alomari et al., 2025*). MRI apparent diffusion coefficient (ADC) brain images used by *Vijithananda et al. (2022)* to extract features and applied machine learning for chart image classification. *Thiyam, Singh & Bora (2023)* benchmarked medical chart classification methods, while *Huang, Zong & Tan (2007)* introduced a multi-instance learning system that excelled in pie and bar chart identification. This evaluated various chart image datasets, employed learning rate of 0.001, root mean square propagation (RMSProp) optimizer, ReLu activation, and dropout regularization. *Nakano et al. (2023)* stratified chronic heart failure data using advanced machine learning a complex classification task analogous to visual data grouping. Ensemble techniques originally introduced by *Dietterich (2000)* have been adapted to financial data visualizations such as candlestick charts. *Santur (2022)* developed a system named Candlestick, which analyzes financial asset charts to support decision-making under noisy conditions. Machine learning techniques were applied to decision-making in the financial market to improve trading performance. *Eslami, Heidarie Golafzani & Naghibi (2023)* entered the field of geotechnical engineering, where they created and implemented triangular charts for the classification of deltaic CPTu-based soil behavior. Their work presents a cunning technique for portraying soil directly, which can have tremendous consequences for improvement and establishment projects. *Cheng, Chen & Ho (2022)* used multi-label CNNs to classify concurrent control chart patterns, while *Zan et al. (2020)* incorporated control chart pattern recognition (CCPR) techniques that utilize CNN and information fusion to classify control chart patterns. These domain-specific studies not only reflect the adaptability of machine learning models but also show the importance of tailored methodologies for visual data in real-world contexts.

Table 1 provides a taxonomy of recent studies on different techniques for chart image classification. It highlights key models, datasets, chart types, and findings, along with limitations identified across different approaches.

MATERIALS AND METHODS

The purpose of studying the classification of chart images from an unsupervised dataset seeks to develop a method that can autonomously organize and categorize visualized data without the need for manual labeling. Data preprocessing techniques, *i.e.*, resize the image that makes models with properly on chart images, are applied. Extract features are computed using VGG-16 of a subject that are categorized into the local and global color-based features, texture, or shape. To reduce the quantity of features to the most unique and important ones utilized is PCA. K-means clustering algorithm used to group similar chart images based on their visual characteristics into 4 cluster. The powerful advanced algorithms of customized CNN and a pre-trained transfer learning model named ResNet50, ResNet50V2, and DenseNet are used for the image classification of four classes. By classification this methodology is useful to analyze meaningful visual data from charts by making proper.

Figure 1 illustrates the process where chart images are provided as input and subjected to preprocessing techniques. Following this, feature vectors are computed using VGG16. Subsequently, K-means clustering is employed to identify clusters. Finally, a customized CNN along with ResNet50, ResNet50V2, and DenseNet are utilized for image classification, producing the final output.

About dataset

In this study ChartVQA dataset was selected due to its diverse large-scale and realistic data from real-world articles and reports charts images for visual question answering (VQA) tasks. Datset consists of 20,882 chart images with four major chart types such as 2,882

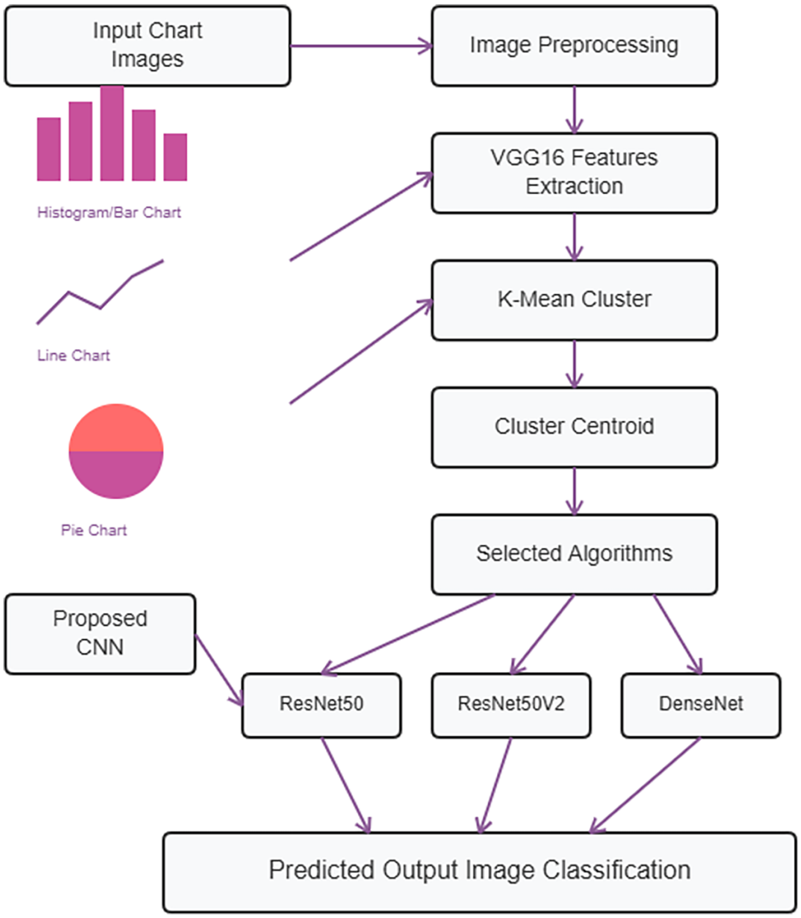


Figure 1 **Proposed methodology diagram.** Full-size DOI: 10.7717/peerj-cs.3148/fig-1

histogram, 4,000 pie, 8,000 bar images and 6,000 line chart images. Multi-class chart classification tasks are well-suited for it.

Figure 2 contains some sample images from a dataset that is used for training our model. The actual dimensions of images are 224 × 224 pixels which are reduced during preprocessing step.

Data preprocessing

The original chart images in PNG format are resized to 200 × 200 pixels in RGB format to maintain consistency across the dataset. This resizing is essential for ensuring compatibility with the VGG16 model used for feature extraction. Additionally, the pixel values are scaled to the range [0, 1] by dividing each value by 255. Stabilizing the training process through normalization during preprocessing is crucial, as it prevents gradients from becoming too large or too small during backpropagation. Figure 3 shows the sample resize image of bar chart, similarly all images are resized.

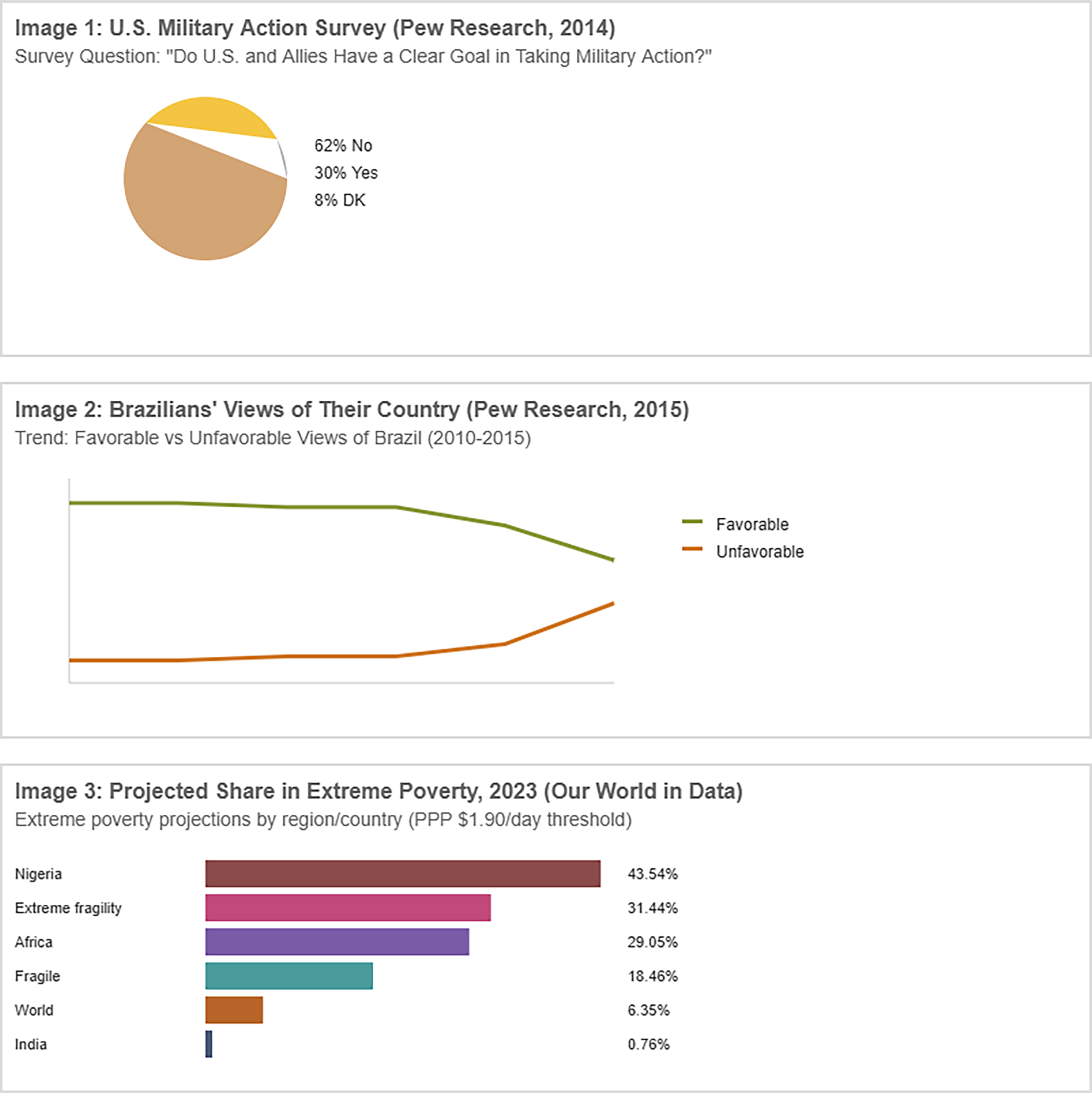


Figure 2 Sample of training images from dataset.

Full-size DOI: 10.7717/peerj-cs.3148/fig-2

Features extraction and clustering

The pretrained VGG16 model is used for feature extraction from the block5_pool layer, offering both computational efficiency and strong performance, particularly in

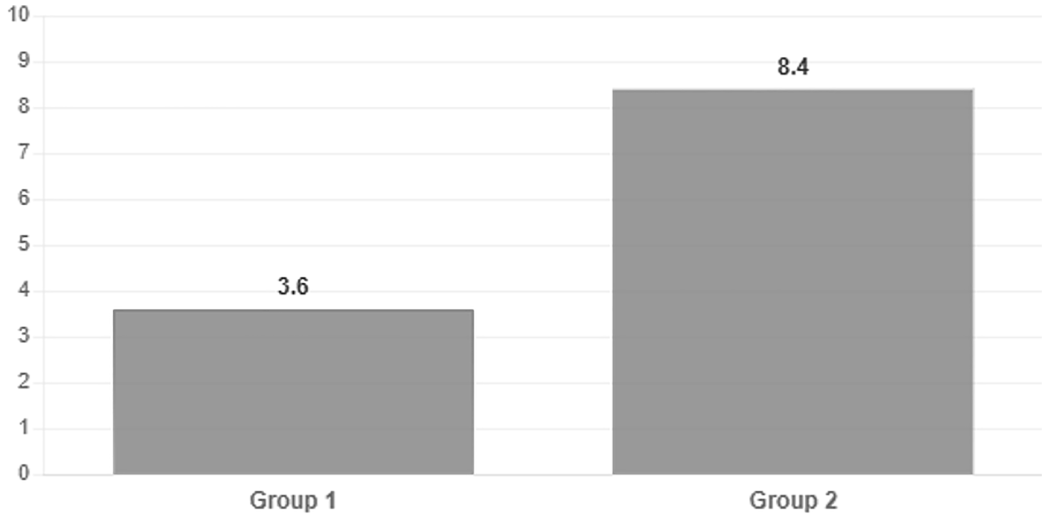


Figure 3 **Sample size.** Full-size DOI: 10.7717/peerj-cs.3148/fig-3

unsupervised learning scenarios. It is more interpretable and easier to manage when extracting visual features such as edges, textures, and spatial patterns from the final convolutional block, due to its use of 3 × 3 convolutional layers and 2 × 2 max-pooling layers compared to more complex architectures like Xception, EfficientNet, or Inception.

Clustering with K-means and PCA

Extracted features from VGG16 from the block5_pool layer of VGG16 were first reduced using PCA. Approximately 200–300 dimensions were reduced and retained the top components that explained 95% of the total variance which helpful for K-means algorithm by removing noise and redundant features. K = 4 clusters was chosen due to match with chart types in dataset to validate visual separability among these four types. An unsupervised pipeline that was streamlined and interpretable without excessive computational overhead was made possible by the combination of VGG16, PCA, and K-means.

Figure 4 shows the actual data points of images computed by VGG16 for and Fig. 5 is four group cluster generated by k-means.

Selected algorithms

CNN architecture is used for the image classification task due to its ability to autonomously learn hierarchical features from data. All selected models are designed to capture both low-level and high-level attributes from input images for accurate classification into chart types (histogram, bar chart, line chart, pie chart). The proposed custom CNN model uses 32 filters of size 3 × 3 and includes a max-pooling layer to classify chart images. Additionally, models such as ResNet50 and ResNet50V2, both pretrained on ImageNet, are employed to learn complex patterns in chart images through their deep

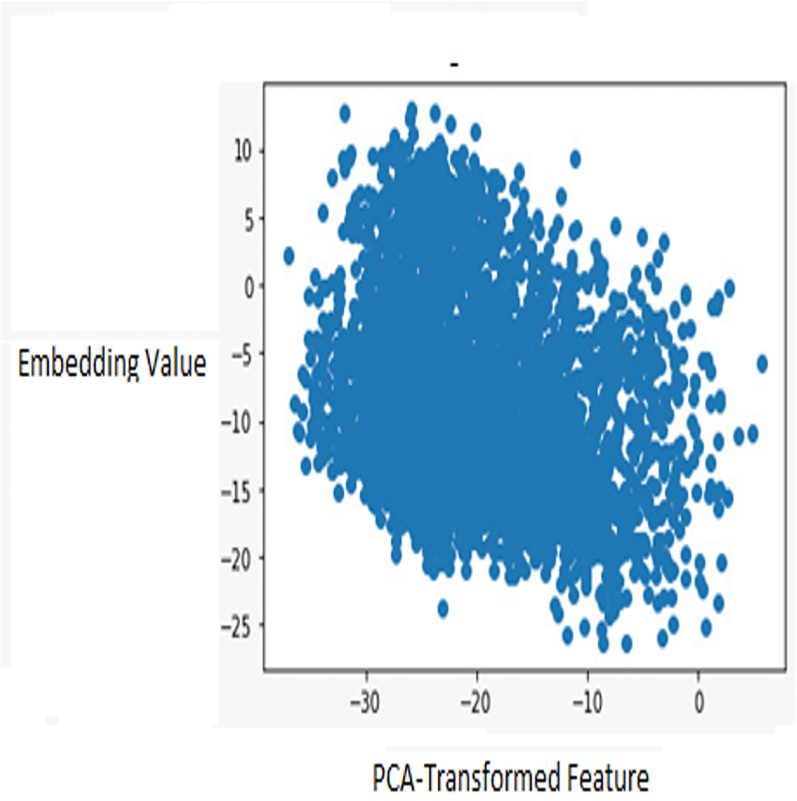


Figure 4 **Featured DataPoint.** Full-size DOI: 10.7717/peerj-cs.3148/fig-4

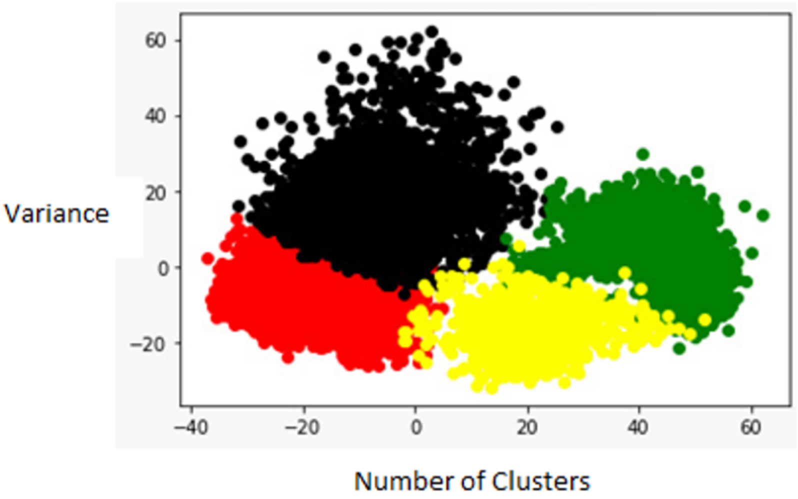


Figure 5 **KMean_Cluster.** Full-size DOI: 10.7717/peerj-cs.3148/fig-5

residual learning capabilities. Finally, DenseNet improves classification efficiency by enabling feature reuse and reducing the number of parameters. The parameters used to train and validate the models are summarized in Table 2.

Table 2 **Parameters table for all selected models.**

Model	Input size	Conv layer	Activation function	Pooling layer	Dropout rate	Output layer	Learning rate	Epochs	Batch size	Optimizer	Loss function
Custom CNN	224 × 224 × 3	32 filters, 3 × 3 kernel	ReLU (Dense), SoftMax	Max Pooling (2 × 2)	0.25	SoftMax	0.001	30	32	RMSProp	Categorical Crossentropy
ResNet50	200 × 200 × 3	Pre-trained on ImageNet	ReLU, SoftMax	–	0.5	SoftMax	0.001	30	32	RMSProp	Categorical Crossentropy
ResNet50V2	200 × 200 × 3	Pre-trained on ImageNet	ReLU, SoftMax	–	0.5	SoftMax	0.001	30	32	RMSProp	Categorical Crossentropy
DenseNet	200 × 200 × 3	Pre-trained on ImageNet	ReLU, SoftMax	–	0.5	SoftMax	0.001	30	32	RMSProp	Categorical Crossentropy

Implementation details

Hardware

All experiments were conducted using Google Colab Pro, which provided the following computational resources:

- GPU: NVIDIA Tesla T4
- CPU: Single-core Intel Xeon
- RAM: 13 GB

Software

The implementation environment was based on:

- Operating system: Ubuntu-based Colab virtual environment
- Programming language: Python 3.9
- Deep learning libraries: TensorFlow 2.12, Keras
- Supporting libraries: NumPy, Matplotlib, Scikit-learn

RESULTS AND DISCUSSION

To ensure reliable model evaluation and prevent overfitting, the dataset was divided into 60% training, 20% validation, and 20% test sets. This 60/20/20 split is widely accepted in machine learning practice, especially when the dataset size is moderately large as is the case with our dataset of 20,882 chart images. The 60% training set provides sufficient data for learning, the 20% validation set allows for hyperparameter tuning and early stopping decisions, and the 20% test set enables robust evaluation of model generalization. To maintain class balance across all split's, stratified sampling was applied during data partitioning. This ensured that each of the four classes (bar chart, line chart, pie chart, histogram) retained approximately the same proportion in the training, validation, and test sets as in the original dataset. The original dataset has a natural imbalance (*e.g*., bar charts = 8,000, histograms = 2,882), so preserving these proportions during splitting was essential to avoid training bias or skewed performance evaluation.

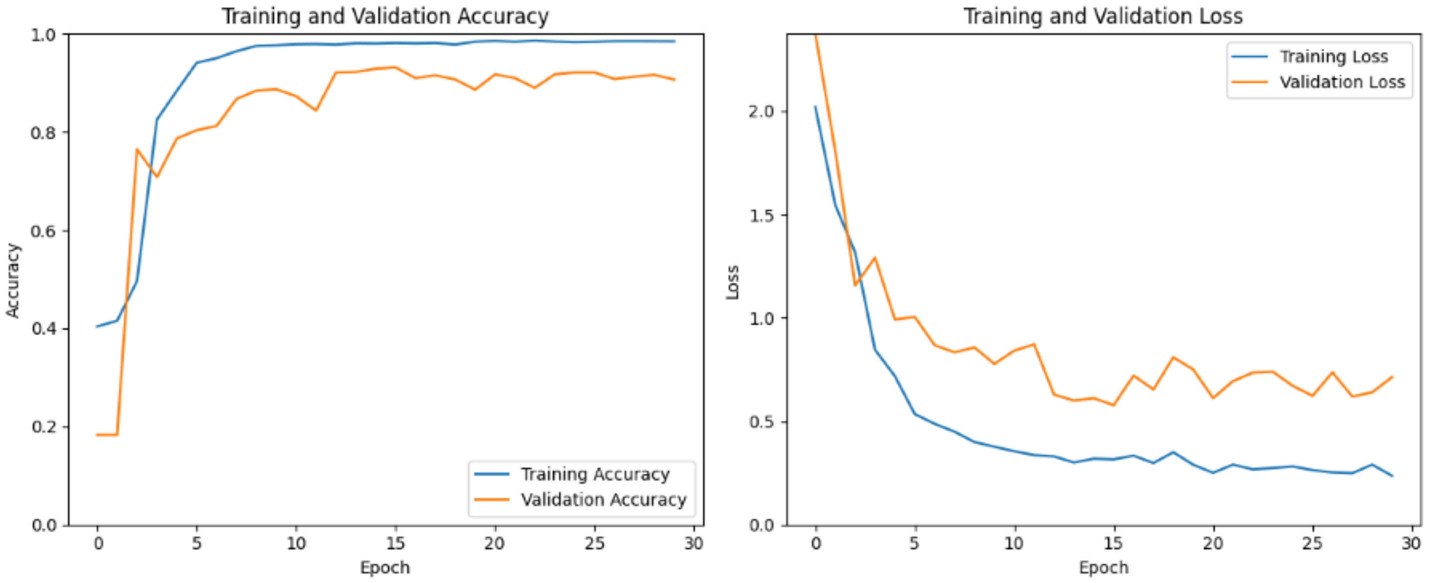


Figure 6 **Custom CNN train and valid loss and accuracy.** Full-size DOI: 10.7717/peerj-cs.3148/fig-6

Models training and validation accuracies and losses

All selected models, custom CNN, RestNet50, RestNet50V2 and DenseNet performance are measured with training and validation accuracies and losses. In this scenario, training is conducted over 30 epochs, meaning all models will go through the entire training dataset 30 times.

Model training CNN custom layer

Model training in CNNs involves optimizing the model's boundaries to minimize a defined misfortune function while maximizing performance measurements. Figure 6 shows training and validation accuracy and loss over multiple 25–30 epochs. Training accuracy increases steadily and reaches a high level, surpassing 0.9825 (98.25%) by the final epochs. This consistent rise suggests that the model effectively learn patterns in the training data and fit it well. The validation accuracy starts around 0.0198 and fluctuates over the different epochs and reaches significant upward trend then finally it stabilizes around 0.90–0.92. On the other hand, at initial epochs (Epochs 0–3), both training and validation loss decrease initially, indicating that the model is learning and reducing errors on both the training and validation sets. The decrease in training loss is sharper than that in validation loss, which is typical as the model starts to fit the training data. The training loss continues to decrease steadily, indicating ongoing improvement in fitting the training data. However, validation loss fluctuates, showing noticeable peaks around different epochs. These fluctuations suggest that the model is experiencing some challenges in generalizing to the validation set, potentially due to slight overfitting or an unoptimized learning rate.

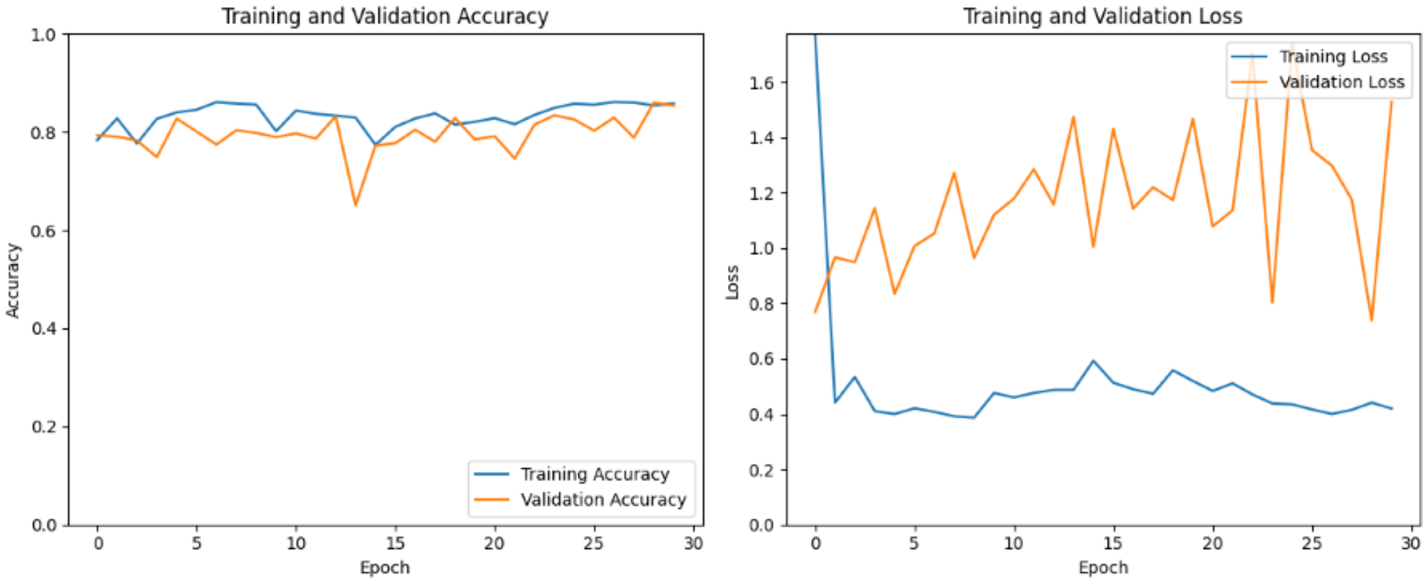


Figure 7 **ResNetV1 train and valid loss and accuracy.** Full-size DOI: 10.7717/peerj-cs.3148/fig-7

Model training ResNet50 V1

By selecting ResNet50, the top layers of the pretrained model, commonly comprising dense layers responsible for classification, are disposed of, retaining the convolutional base. This stage involves iteratively adjusting the model's boundaries to minimize a defined misfortune function while maximizing performance measurements. In Fig. 7, the training and validation accuracy graphs start at around 80% of the initial epochs. The accuracy continues to increase during training but experiences a dip between the 12th and 15th epoch before rising again to nearly 85%. The validation accuracy curve exhibits some fluctuations but eventually converges with the training accuracy in the final epochs. However, the training and validation loss graphs reveal a significant amount of validation loss, indicating that the current model is not adequately trained under the same parameters provided to other models. This suggests that the model may require further tuning or adjustments to improve its performance and reduce the validation loss.

Model training ResNet50 V2

ResNet50 v2 incorporates several enhancements, such as batch normalization after each convolution and the use of identity mappings, which help in mitigating the vanishing gradient problem and improving gradient flow. In Fig. 8 the training accuracy remains nearly constant at a high value (close to 1.0), indicating the model is performing extremely well on the training dataset. The validation accuracy fluctuates and remains significantly lower than the training accuracy, with no clear upward trend. The large gap between training and validation accuracy suggests the model is overfitting to the training data. The lack of improvement in validation accuracy indicates the model struggles to generalize to

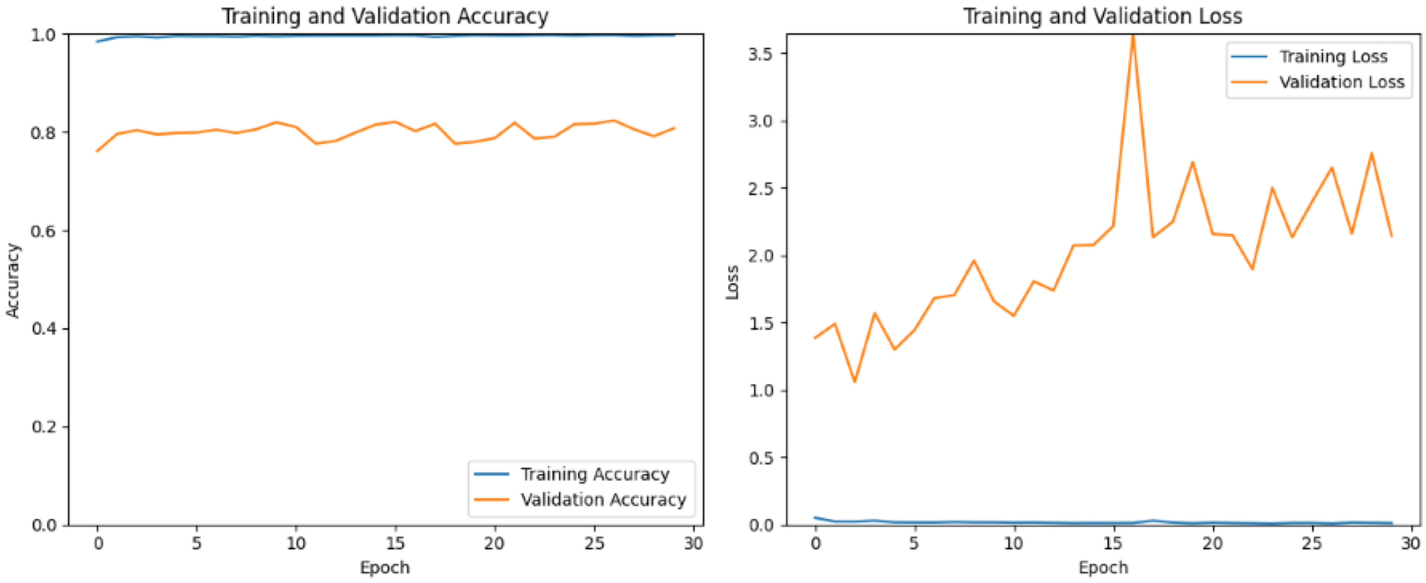


Figure 8 ResNetV2 training and validation loss and accuracy. Full-size DOI: 10.7717/peerj-cs.3148/fig-8

unseen data, possibly due to insufficient regularization or a lack of diverse features in the training set. The training loss is consistently very low, which aligns with the high training accuracy. Validation loss is much higher and fluctuates significantly, with a notable spike around epoch 15 and continued instability thereafter. The divergence between training and validation loss further confirms overfitting. The spike in validation loss could indicate instability during training, possibly caused by a learning rate that is too high or by overfitting to the training data. The model fits the training data almost perfectly but fails to generalize with the validation data.

Model training DenseNet

DenseNet, short for dense convolutional network, is chosen for its unique architecture described by dense connectivity patterns between layers. Unlike traditional CNNs where each layer is connected only to the subsequent layer, DenseNet lays out direct connections between all layers within a dense block. Figure 9 shows the training process has yielded a perfect training accuracy of 100%, however, the validation accuracy stands at approximately 80%. This discrepancy suggests that while the model has learned to classify the training data flawlessly, it may be overfitting, meaning it has memorized the training data rather than learning to generalize from it. The 80% validation accuracy indicates that the model's performance drops when applied to new, unseen data. The training process has shown a consistent training loss of nearly 0% across all 30 epochs, indicating that the model has effectively learned the training data. However, the validation loss presents a different story, initially increasing steadily from 0.8 to 3 over various epochs, suggesting that the model was struggling to generalize to unseen data and was likely overfitting. Interestingly, from epoch 25 to 30, the validation loss began to decrease, reaching 1.3 by the 30th epoch.

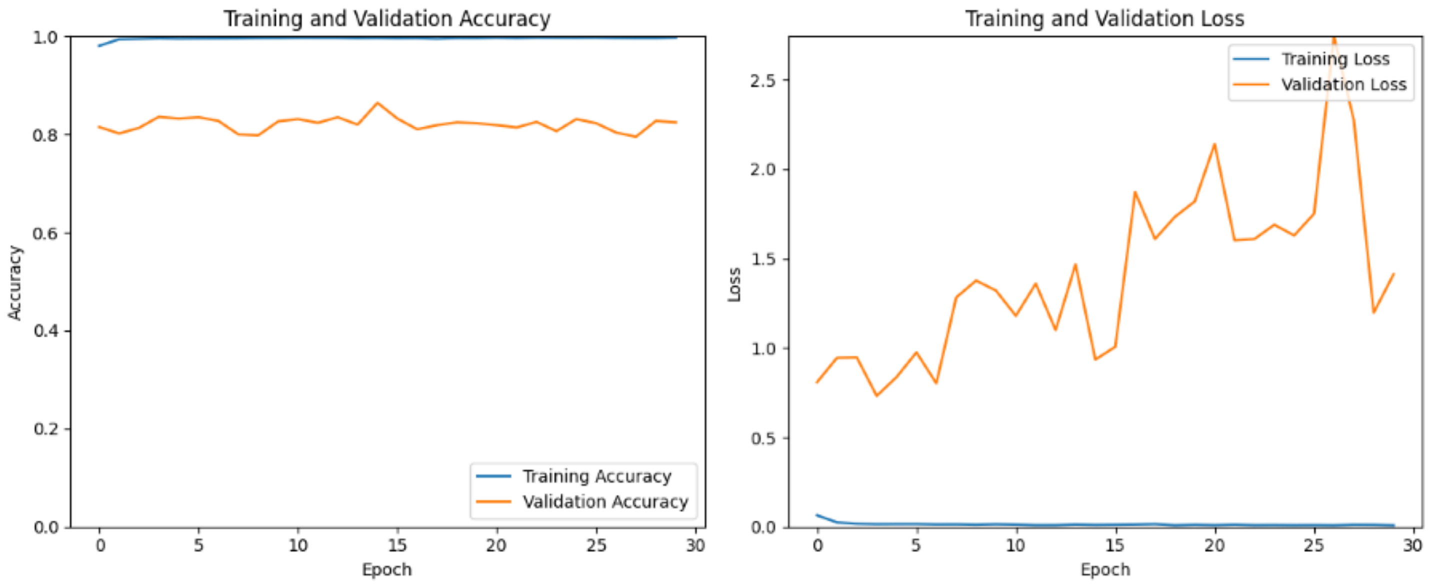
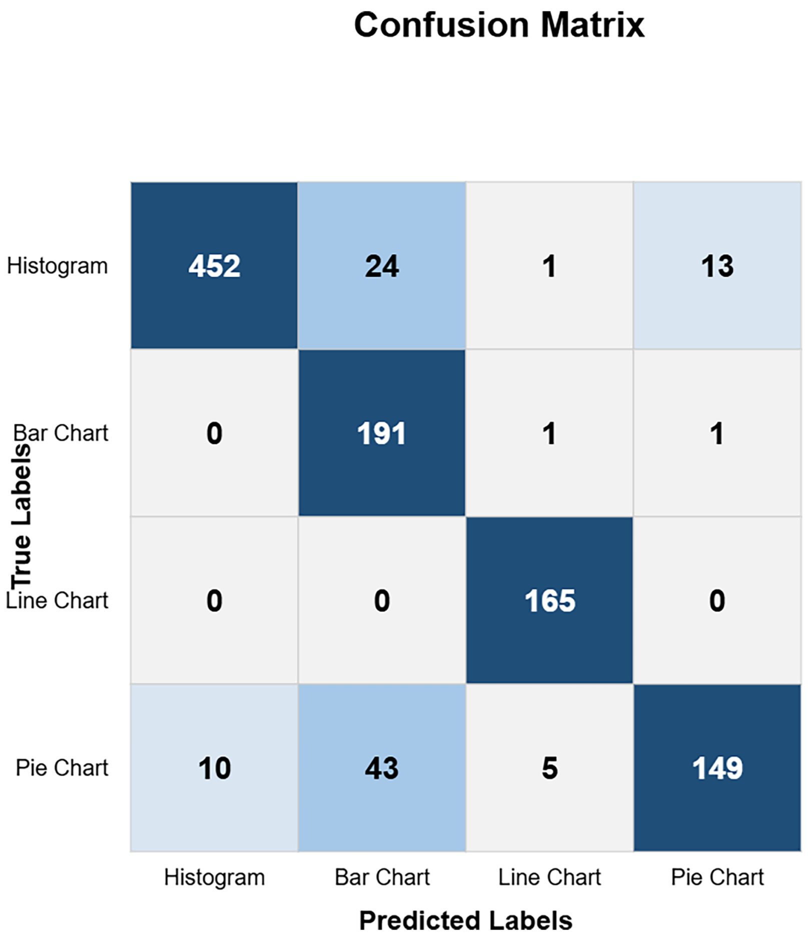
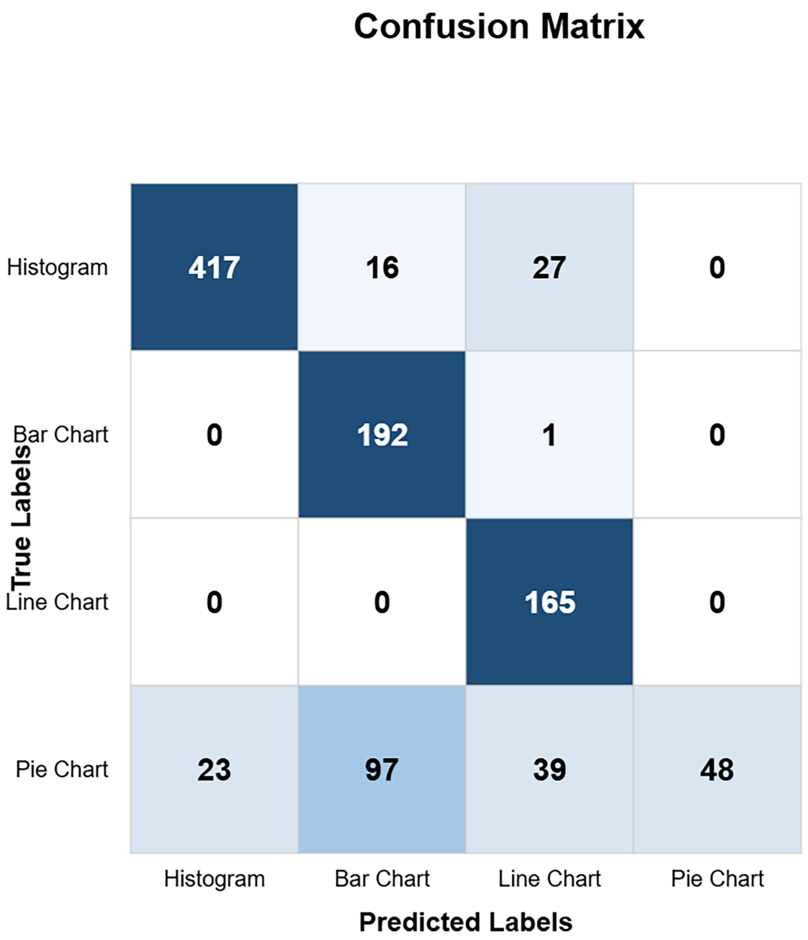


Figure 9 DenseNet train and valid loss and accuracy. Full-size DOI: 10.7717/peerj-cs.3148/fig-9



	Histogram	Bar Chart	Line Chart	Pie Chart
Histogram	452	24	1	13
Bar Chart	0	191	1	1
Line Chart	0	0	165	0
Pie Chart	10	43	5	149

Figure 10 Custom CNN confusion matrix (CM). Full-size DOI: 10.7717/peerj-cs.3148/fig-10



	Histogram	Bar Chart	Line Chart	Pie Chart
Histogram	417	16	27	0
Bar Chart	0	192	1	0
Line Chart	0	0	165	0
Pie Chart	23	97	39	48

Figure 11 RestNetV1 confusion matrix (CM). Full-size DOI: 10.7717/peerj-cs.3148/fig-11

Models confusion matrix

The confusion matrix in Fig. 10 shows the custom CNN model's high classification accuracy with 452 histograms, 191 bar charts, 165 line charts, and 149 pie charts as true positives, indicating its robustness and effectiveness. The model produced very few misclassified values, indicating that it performs exceptionally well overall. This high level of accuracy across different chart types demonstrates the model's robustness and effectiveness in correctly identifying various visual data representations. Figure 11 highlights the model ResNet50V1 high accuracy for histograms (460), bar charts (192), and line charts (161), but it struggles with pie charts correctly identifying only 88 and frequently misclassifying other charts as pie charts. This indicates a notable weakness in the model's ability to accurately distinguish pie charts from other types of visual data representations. In Fig. 12, the ResNet50V2 model shows high accuracy for histograms (447) with few misclassifications (16 bar charts, 27 line charts, 0 pie charts), but it struggles with pie charts, correctly identifying only 48 and often misclassifying other charts as pie charts. The confusion matrix for the DenseNet model in Fig. 13 shows it correctly identified 12 histograms but misclassified 467 as bar charts, failed to correctly predict any

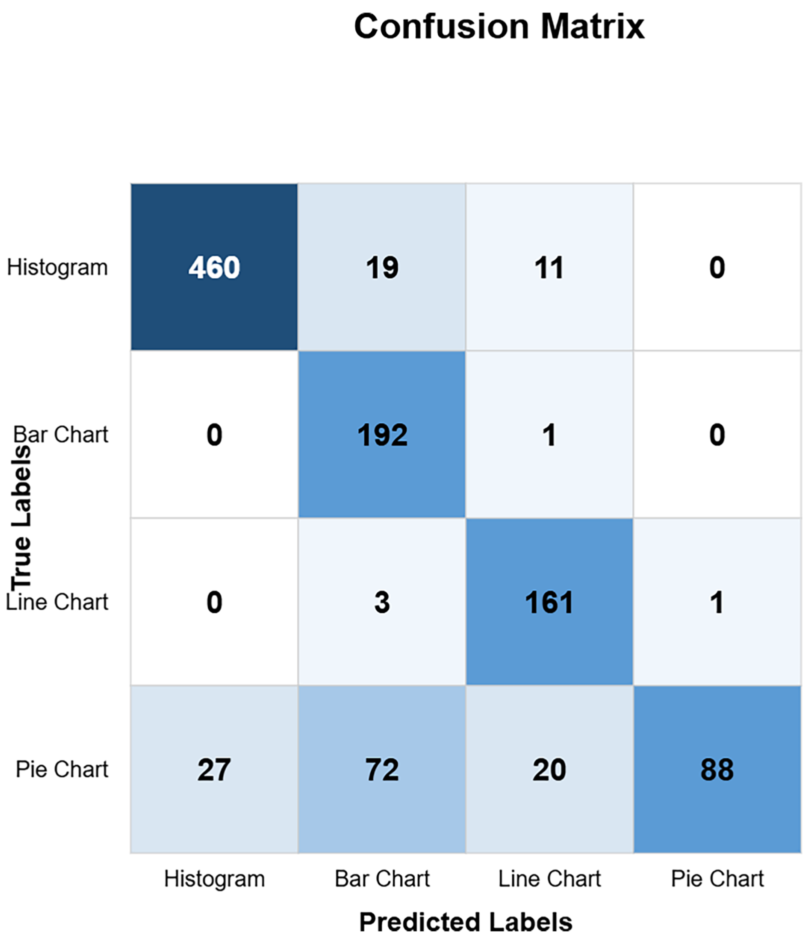


Figure 12 ResNetV2 confusion matrix (CM). Full-size DOI: 10.7717/peerj-cs.3148/fig-12

bar charts, misclassifying 192 as histograms, performed well for line charts (165 correct), but struggled with pie charts, correctly identifying only 55 and misclassifying many as other types.

Standard evaluation metrics, including accuracy, precision, recall, F1-score, and area under the curve (AUC), were computed to assess the performance of all models. Table 3 presents the highest accuracy (90.71%) and AUC (0.94) achieved by custom CNN, indicating its strong ability to balance sensitivity across all chart classes. ResNet50 achieved an accuracy of 85.40%, suggesting a tendency to miss certain chart types. Other models, such as ResNet50V2 and DenseNet, showed overall lower performance, particularly struggling with the identification of pie charts. These findings are consistent with the confusion matrix analysis (Figs. 10–13), where the custom CNN architecture effectively classified images compared to the other models, which frequently misclassified charts due to their complex internal representations.

Figure 14 illustrates the custom CNN architecture, starting with an input size of 200 × 200 × 3. The input is processed through an initial convolutional layer (Conv2D) with 32 filters, followed by three sequential convolutional blocks to extract hierarchical features.

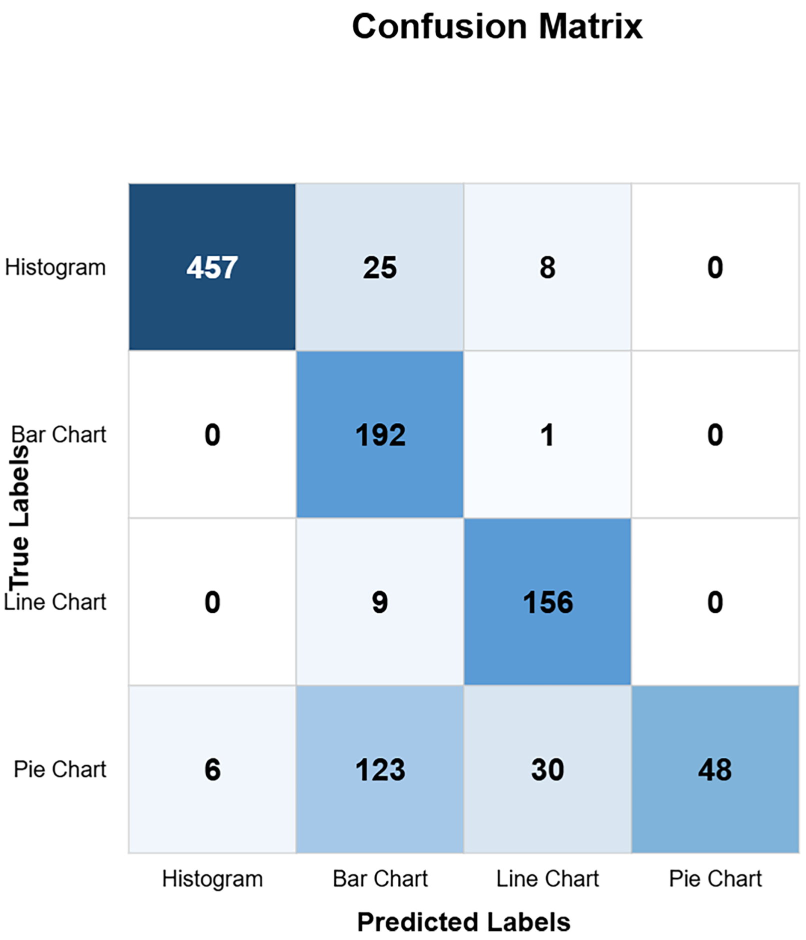


Figure 13 **DensNet confusion matrix (CM).** Full-size DOI: 10.7717/peerj-cs.3148/fig-13

Table 3 **Performance comparison of custom and pretrained models on chart classification task.**

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)	AUC (macro)
Custom CNN	90.71	91.2	90.1	90.6	0.94
ResNet50	85.40	86.3	83.5	84.8	0.89
ResNet50V2	80.76	81.0	79.2	80.1	0.86
DenseNet	82.46	84.5	78.4	81.3	0.87

The output is then passed through two dense layers with 32 units each, leading to the classification head where 2D feature representations are finalized. Complete architecture showcases a systematic approach to image classification through progressive feature abstraction and dimensionality reduction.

Model performance

Among all evaluated models, our proposed custom CNN architecture consistently achieved the highest overall accuracy, demonstrating strong generalization across all four chart types. In contrast, DenseNet, while powerful in general-purpose vision tasks, showed

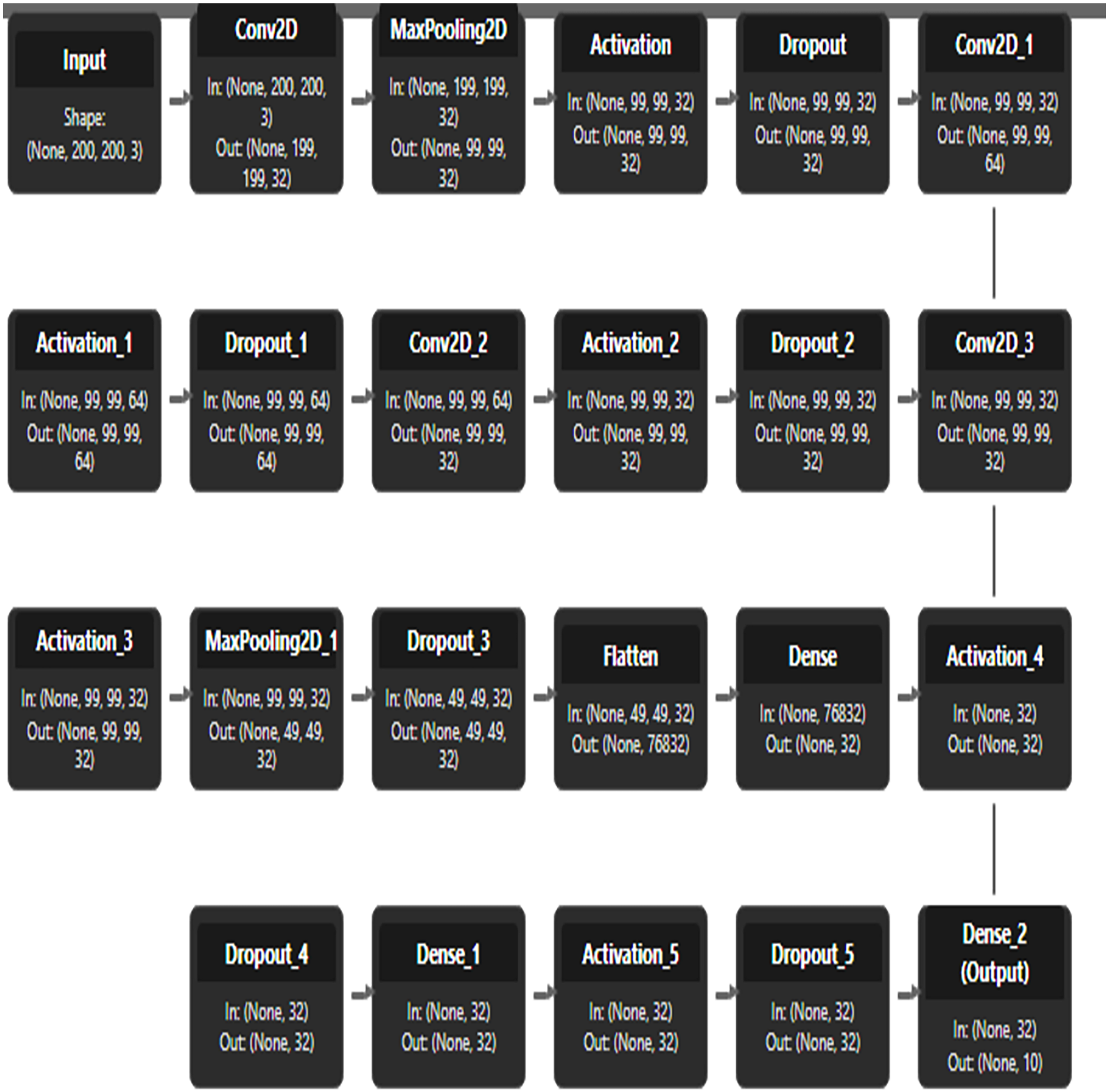


Figure 14 **CNN architecture.** Full-size DOI: 10.7717/peerj-cs.3148/fig-14

the lowest performance, particularly on pie charts. DenseNet often misclassified pie charts as bar or donut charts, likely due to its sensitivity to overlapping visual features and complex internal representations that are not well-suited to the stylistic variability in pie charts. On the other hand, custom CNN designed with domain-specific simplicity

Table 4 Execution time of custom and pretrained models across 30 training epochs.

Model	Training time (30 epochs)	Validation time (Per epoch)	Total execution time	Notes
Custom CNN	4 min 30 s	8 s	~5 min	Lightweight model with fewer parameters
ResNet50 (Pretrained)	12 min 10 s	10 s	~13 min	Deeper architecture, slower convergence
ResNet50V2 (Pretrained)	13 min 40 s	12 s	~15 min	Improved gradient flow, higher cost
DenseNet (Pretrained)	15 min 30 s	11 s	~17 min	High feature reuse, heavier model

effectively captured geometric and structural distinctions (*e.g.*, segment shapes, axis lines) with fewer parameters and lower risk of overfitting. This contrast highlights the advantage of lightweight, task-specific architectures for structured visual data like chart images.

Table 4 presents a comparative analysis of the execution times for four deep learning models used in the chart image classification task. The custom CNN model demonstrates the fastest overall training and validation time, completing 30 epochs in approximately 5 min due to its lightweight architecture and fewer parameters. In contrast, the pretrained models ResNet50, ResNet50V2, and DenseNet exhibit longer training and validation times, ranging from approximately 13 to 17 min in total.

CONCLUSIONS

In conclusion, this study on chart image classification using clustering techniques and deep feature extraction demonstrated strong and competitive results across multiple models. Initially, features are extracted from chart images using a pre-trained VGG16 model and grouped using K-means clustering in an unsupervised manner. These clusters are then used to guide training of supervised models, including ResNet50, ResNet50V2, DenseNet, and custom CNN. Among these, custom CNN outperformed the others, achieving a classification accuracy of 90.71%, compared to 85.40% (ResNet50), 82.46% (DenseNet), and 80.76% (ResNet50V2).

Custom CNN's effectiveness is attributed to its lightweight architecture, appropriate regularization, and task-specific design, which allowed it to generalize better, especially for visually diverse and underrepresented chart types like pie charts. In contrast, deeper models such as DenseNet showed signs of overfitting and struggled particularly with curved and abstract visual structures. Despite these promising results, several limitations exist. The dataset used, while comprehensive, consists mostly of web-extracted or synthetic charts and may not fully represent real-world chart variability.

A limitation of this research is that the model is trained on only four fixed chart types with labels in a single language, and it does not address embedded text within the charts. These aspects can be explored and enhanced in future work.

Future direction

The proposed model can be further trained on larger and more diverse datasets, including additional chart types such as radar charts, area charts, and Gantt charts. It can also be evaluated using benchmark datasets like PlotQA, LEAF-QA, DocVQA, and FigureQA.

Future enhancements may include handling low-resolution images, embedded annotations, and multilingual labels within charts. The Friedman aligned rank test or the Wilcoxon signed-rank test will be used for further performance evaluation.

ACKNOWLEDGEMENTS

The authors would like to thank all of the individuals and institutions that contributed to this research.

ADDITIONAL INFORMATION AND DECLARATIONS

Funding

This work was supported by Universiti Teknikal Malaysia Melaka (UTeM) and the Ongoing Research Funding program, (ORF-2025-395), King Saud University, Riyadh, Saudi Arabia. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Grant Disclosures

The following grant information was disclosed by the authors:
Universiti Teknikal Malaysia Melaka (UTeM).
Ongoing Research Funding program: ORF-2025-395.
King Saud University, Riyadh, Saudi Arabia.

Competing Interests

The authors declare that they have no competing interests.

Author Contributions

- Syed Muhammad Hassan Zaidi conceived and designed the experiments, analyzed the data, authored or reviewed drafts of the article, and approved the final draft.
- Abdul Hafeez Khan conceived and designed the experiments, analyzed the data, prepared figures and/or tables, and approved the final draft.
- Syed Azeem Inam conceived and designed the experiments, performed the experiments, authored or reviewed drafts of the article, and approved the final draft.
- Abdullah Ayub Khan conceived and designed the experiments, performed the experiments, prepared figures and/or tables, and approved the final draft.
- Jamil Abedalrahim Jamil Alsayaydeh performed the computation work, authored or reviewed drafts of the article, and approved the final draft.
- Ahmad Ali AlZubi performed the computation work, prepared figures and/or tables, and approved the final draft.
- Benny Ogunshola performed the experiments, prepared figures and/or tables, and approved the final draft.
- Safarudin Gazali Herawan analyzed the data, prepared figures and/or tables, and approved the final draft.

Data Availability

The following information was supplied regarding data availability:

The code is available at Zenodo:

- hassandit1992. (2025). hassandit1992/Charts-Classification: Alpha Version (V1.00). Zenodo. https://doi.org/10.5281/zenodo.15709641

- Syed Muhammad Hassan Zaidi, Abdul Hafeez Khan, Syed Azeem Inam, Abdullah Ayub Khan, Jamil Abedalrahim Jamil Alsayaydeh, Ahmad Ali AlZubi, Benny Ogunshola, & Safarudin Gazali Herawan. (2025). Unsupervised Classification of Multi-Class Chart Images: A Comparison of Customized CNNs and Transfer Learning Techniques [Data set]. Zenodo. https://doi.org/10.5281/zenodo.14882564.

REFERENCES

Al-Kady Y, Abdel-Wahab N, Hassanein F, Al-Alfy R, El-Malatawy L, Ammar R, Al-Shetairy M, Amin S. 2023. Chartlytics: chart image classification and data extraction. In: *2023 Eleventh International Conference on Intelligent Computing and Information Systems (ICICIS).* Piscataway: IEEE, 515–522.

Alomari MA, Al-Andoli MN, Ghaleb M, Thabit R, Alkawsi G, Alsayaydeh JAJ, Abdulguddoos SAG. 2025. Security of smart grid: cybersecurity issues, potential cyberattacks, major incidents, and future directions. *Energies* **18(1)**:141 DOI 10.3390/en18010141.

Alsayaydeh JAJ, Yusof MF, Abdul Halim MZ, Zainudin MNS, Herawan SG. 2023. Patient health monitoring system development using ESP8266 and arduino with IoT platform. *International Journal of Advanced Computer Science and Applications* **14(4)**:617–624 DOI 10.14569/IJACSA.2023.0140467.

Alsayaydeh JAJ, Zainon M, Baskaran H, Herawan SG. 2022. Intelligent interfaces for assisting blind people using object recognition methods. *International Journal of Advanced Computer Science and Applications* **13(5)**:472–479 DOI 10.14569/IJACSA.2022.0130584.

Araujo T, Chagas P, Alves J, Santos C, Sousa Santos B, Serique Meiguins B. 2020. A real-world approach on the problem of chart recognition using classification, detection and perspective correction. *Sensors* **20(16)**:4370 DOI 10.3390/s20164370.

Attique H, Shah S, Jabeen S, Khan FG, Khan A, ELAffendi M. 2022. Multiclass cancer prediction based on copy number variation using deep learning. *Computational Intelligence and Neuroscience* **2022** DOI 10.1155/2022/4742986.

Bajic F, Job J, Nenadić K. 2020. Data visualization classification using simple convolutional neural network model. *International Journal of Electrical and Computer Engineering Systems* **11(1)**:43–51.

Bajić F, Job J. 2021. Chart classification using siamese CNN. *Journal of Imaging* **7(11)**:220 DOI 10.3390/jimaging7110220.

Bansal M, Kumar M, Sachdeva M, Mittal A. 2021. Transfer learning for image classification using VGG19: caltech-101 image data set. *Journal of Ambient Intelligence and Humanized Computing* **14(4)**:3609–3620 DOI 10.1007/s12652-021-03488-z.

Bauerle A, Albus P, Störk R, Seufert T, Ropinski T. 2022. exploRNN: teaching recurrent neural networks through visual exploration. *The Visual Computer* **39(9)**:4323–4338 DOI 10.1007/s00371-022-02593-0.

Birant D, Akça A, Bozkurt B, Bağlan M. 2022. Classification of scatter plot images using deep learning. *Dokuz Eylul Üniversitesi Mühendislik Fakültesi Fen ve Mühendislik Dergisi* **24(71)**:631–642 DOI 10.21205/deufmd.2022247126.

Breiman L. 2001. Random forests. *Machine Learning* **45(1)**:5–32 DOI 10.1023/a:1010933404324.

Chagas P, Akiyama R, Meiguins A, Santos C, Saraiva F, Meiguins B, Morais J. 2018. Evaluation of convolutional neural network architectures for chart image classification. In: *2018 International Joint Conference on Neural Networks (IJCNN)*.

Chapelle O, Haffner P, Vapnik VN. 1999. Support vector machines for histogram-based image classification. *IEEE Transactions on Neural Networks* **10(5)**:1055–1064 DOI 10.1109/72.788646.

Cheng C-S, Chen P-W, Ho Y. 2022. Control chart concurrent pattern classification using multi-label convolutional neural networks. *Applied Sciences* **12(2)**:787 DOI 10.3390/app12020787.

Cortes C, Vapnik V. 1995. Support-vector networks. *Machine Learning* **20(3)**:273–297 DOI 10.1007/BF00994018.

Del Bimbo D, Gemelli A, Marinai S. 2022. Data augmentation on graphs for table type classification. In: *Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR)*, 242–252.

Dietterich TG. 2000. Ensemble methods in machine learning. *Multiple Classifier Systems* **1857(3)**:1–15 DOI 10.1007/3-540-45014-9_1.

Elzer S, Schwartz E, Carberry S, Chester D, Demir S, Wu P. 2007. A browser extension for providing visually impaired users access to the content of bar charts on the web. In: *WEBIST (2)*, 59–66.

Eslami A, Heidarie Golafzani S, Naghibi MH. 2023. Developed triangular charts; deltaic CPTu-based soil behavior classification using AUT: CPTu-Geo-Marine Database. *Probabilistic Engineering Mechanics* **71(3)**:103380 DOI 10.1016/j.probengmech.2022.103380.

Fawagreh K, Gaber MM, Elyan E. 2014. Random forests: from early developments to recent advancements. *Systems Science & Control Engineering* **2**:602–609 DOI 10.1080/21642583.2014.956265.

Hastie T, Tibshirani R, Friedman J, Franklin J. 2005. The elements of statistical learning: data mining, inference and prediction. *The Mathematical Intelligencer* **27(2)**:83–85.

Hossain Z, Hassim NB, Alsayaydeh JAJ, Hasan MK, Islam MR. 2021. A tree-profile shape ultra wide band antenna for chipless RFID tags. *International Journal of Advanced Computer Science and Applications* **12(4)**:546–550 DOI 10.14569/IJACSA.2021.0120469.

Huang W, Zong S, Tan C. 2007. Chart image classification using multiple-instance learning. In: *Proceedings*, 27.

Jiang Z-P, Liu Y-Y, Shao Z-E, Huang K-W. 2021. An improved VGG16 model for pneumonia image classification. *Applied Sciences* **11(23)**:11185 DOI 10.3390/app112311185.

Jung D, Kim W, Song H, Hwang JI, Lee B, Kim B, Seo J. 2017. ChartSense: interactive data extraction from chart images. In: *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, 6706–6717.

Khang AWY, Elias SJ, Zulkifli N, Indra WA, Alsayaydeh JAJ, Manap Z, Gani JAM. 2020. Qualitative based QoS performance study using hybrid ACO and PSO algorithm routing in MANET. *Journal of Physics: Conference Series* **1502**:012004 DOI 10.1088/1742-6596/1502/1/012004.

Kim HJ, Lell N, Scherp A. 2024. Text role classification in scientific charts using multimodal transformers. In: *International Conference on Applications of Natural Language to Information Systems*, 47–61.

Kosemen C, Birant D. 2020. Multi-label classification of line chart images using convolutional neural networks. *SN Applied Sciences* **2(7)**:1250 DOI 10.1007/s42452-020-3055-y.

Kusiak A. 2019. Convolutional and generative adversarial neural networks in manufacturing. *International Journal of Production Research* **58(5)**:1594–1604 DOI 10.1080/00207543.2019.1662133.

Lavangnananda K, Piyatumrong A. 2005. Image processing approach to features extraction in classification of control chart patterns. In: *Proceedings of the 2005 IEEE Midnight-Summer Workshop on Soft Computing in Industrial Applications, 2005. SMCia/05*. Vol. 41, Piscataway: IEEE, 85–90.

LeCun Y, Bengio Y, Hinton G. 2015. Deep learning. *Nature* **521(7553)**:436–444 DOI 10.1038/nature14539.

Mishra P, Kumar S, Chaube MK. 2022. Classifying chart based on structural dissimilarities using improved regularized loss function. *Neural Processing Letters* **54**:2385–2411 DOI 10.1007/s11063-021-10735-z.

Mnih V, Kavukcuoglu K, Silver D, Rusu AA, Veness J, Bellemare MG, Graves A, Riedmiller M, Fidjeland AK, Ostrovski G, Petersen S, Beattie C, Sadik A, Antonoglou I, King H, Kumaran D, Wierstra D, Legg S, Hassabis D. 2015. Human-level control through deep reinforcement learning. *Nature* **518(7540)**:529–533 DOI 10.1038/nature14236.

Nakano K, Nochioka K, Yasuda S, Tamori D, Shiroto T, Sato Y, Takaya E, Miyata S, Kawakami E, Ishikawa T, Ueda T, Shimokawa H. 2023. Machine learning approach to stratify complex heterogeneity of chronic heart failure: a report from the CHART-2 study. *ESC Heart Failure* **10(3)**:1597–1604 DOI 10.1002/ehf2.14288.

Ng AY, Jordan MI. 2001. On discriminative vs. generative classifiers: a comparison of logistic regression and Naive Bayes. *Neural Information Processing Systems* **14(3)**:841–848 DOI 10.1007/s11063-008-9088-7.

Olivas ES, Guerrero JDM, Martinez-Sober M, Magdalena-Benedito JR, Serrano LE. 2009. *Handbook of research on machine learning applications and trends: algorithms, methods, and techniques*. Pennsylvania, USA: IGI Global.

Pereira S, Pinto A, Amorim J, Ribeiro A, Alves V, Silva CA. 2019. Adaptive feature recombination and recalibration for semantic segmentation with fully convolutional networks. *IEEE Transactions on Medical Imaging* **8(12)**:2914–2925 DOI 10.1109/TMI.2019.2918096.

Quinlan JR. 1986. Induction of decision trees. *Machine Learning. Symposium on the Theory of Computing* **1(1)**:81–106 DOI 10.1023/a:1022643204877.

Ren Y, Li X, Jin F, Li C, Liu W, Li E, Zhang L. 2024. Extracting regular building footprints using projection histogram method from UAV-based 3D models. *ISPRS International Journal of Geo-Information* **14(1)**:6 DOI 10.3390/ijgi14010006.

Santur Y. 2022. Candlestick chart-based trading system using ensemble learning for financial assets. *Sigma Journal of Engineering and Natural Sciences—Sigma Mühendislik ve Fen Bilimleri Dergisi* **40(2)**:370–379 DOI 10.14744/sigma.2022.00039.

Scharr H. 2004. Optimal operators in digital image processing. In: *Proceedings of SPIE 5299, Computational Imaging II*, 106–117.

Shaha M, Pawar M. 2018. Transfer learning for image classification. In: *2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA)*, 656–660.

Shaheen SM, Alhalawani S, Alnabet N, Alhenaki D. 2019. Analytical experiments on the utilization of data visualizations. In: Alfaries A, Mengash H, Yasar A, Shakshuki E, eds. *Communications in Computer and Information Science*. Vol. 1097, Springer Nature Switzerland AG, 148–161 DOI 10.1007/978-3-030-36365-9_12.

Simonyan K, Zisserman A. 2014. Very deep convolutional networks for large-scale image recognition. ArXiv DOI 10.48550/arXiv.1409.1556.

Singh M, Kanroo MS, Kawoosa HS, Goyal P. 2023. Towards accessible chart visualizations for the non-visuals: research, applications, and gaps. *Computer Science Review* **48**:100555 DOI 10.1016/j.cosrev.2023.100555.

Thiyam J, Singh SR, Bora PK. 2023. Chart classification: a survey and benchmarking of different state-of-the-art methods. *International Journal on Document Analysis and Recognition* **27(1)**:19–44 DOI 10.1007/s10032-023-00443-w.

Tran PH, Ahmadi Nadi A, Nguyen TH, Tran KD, Tran KP. 2022. Application of machine learning in statistical process control charts: a survey and perspective. In: Tran KP, ed. *Control Charts and Machine Learning for Anomaly Detection in Manufacturing*. Cham, Switzerland: Springer, 7–42 DOI 10.1007/978-3-030-83819-5_2.

Tufte ER, Graves-Morris PR. 1983. *The visual display of quantitative information.* Vol. 2. Cheshire, CT: Graphics Press.

Vijithananda SM, Jayatilake ML, Hewavithana B, Gonçalves T, Rato LM, Weerakoon BS, Kalupahana TD, Silva AD, Dissanayake KD. 2022. Feature extraction from MRI ADC images for brain tumor classification using machine learning techniques. *Biomedical Engineering Online* **21(1)**:52 DOI 10.1186/s12938-022-01022-6.

Wang X. 2024. Application of multimodal fusion in automatic image classification: combining CNN and RNN model. *Intelligent Decision Technologies* **19(2)**:928–942 DOI 10.1177/18724981241299605.

Wang P, Fan E, Wang P. 2020. Comparative analysis of image classification algorithms based on traditional machine learning and deep learning. *Pattern Recognition Letters* **141**:61–67 DOI 10.1016/j.patrec.2020.07.042.

Weiss K, Khoshgoftaar TM, Wang D. 2016. A survey of transfer learning. *Journal of Big Data* **3(1)**:9 DOI 10.1186/s40537-016-0043-6.

Wu YH, Gao SH, Mei J, Xu J, Fan DP, Zhang RG, Cheng MM. 2021. JCS: an explainable COVID-19 diagnosis system by joint classification and segmentation. *IEEE Transactions on Image Processing* **30**:3113–3126 DOI 10.1109/TIP.2021.3058783.

Yang C, Jiang W, Guo Z. 2019. Time series data classification based on dual path CNN-RNN cascade network. *IEEE Access* **7**:155304–155312 DOI 10.1109/ACCESS.2019.2949287.

Yang X, Song Z, King I, Xu Z. 2022. A survey on deep semi-supervised learning. *IEEE Transactions on Knowledge and Data Engineering* **35(9)**:8934–8954 DOI 10.1109/TKDE.2022.3220219.

Zan T, Su Z, Liu Z, Chen D, Wang M, Gao X. 2020. Pattern recognition of different window size control charts based on convolutional neural network and information fusion. *Symmetry* **12(9)**:1472 DOI 10.3390/sym12091472.

Zhou T, Brown M, Snavely N, Lowe DG. 2017. Unsupervised learning of depth and ego-motion from video. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Piscataway: IEEE, 1851–1858.