International Journal of Computing and Digital Systems
ISSN (2210-142X)
Int. J. Com. Dig. Sys. 16, No.1 (Aug-24)

http://dx.doi.org/10.12785/ijcds/160147

A Combined Deep Cnn-Lstm Network For Sketch Recognition

Lale EL Mouna[1], Hassan Silkan[1], Stéphane Cédric Koumetio Tekouabou[2], Youssef Hanyf[3,4], Mohamedou Cheikh tourad[5] and Mohamedade Farouk Nanne[5]

[1]*Laroseri Laboratory, Chouaib Doukkali University,El Jadida,Morocco*
[2]*Center of Urban Systems (CUS), Mohammed VI Polytechnic University (UM6P), Benguérir, Morocco*
[3]*Research laboratory in management and decision support, AI Data SEED team, Ibn Zohr University, Dakhla, Morocco*
[4]*National School of Commerce and Management, Ibn Zohr University, Dakhla, Morocco*
[5]*Scientific Computing, Computer Science and Data Science Research Unit (CSIDS), University of Nouakchott, Nouakchott, Mauritania*

Received 5 Nov. 2023, Revised 19 Apr. 2024, Accepted 26 Apr. 2024, Published 1 Aug. 2024

Abstract: Automating freehand sketches is a complex process due to their diverse and abstract characteristics. Recently, there has been significant interest among researchers in machine learning algorithms, owing to their emergence. Nevertheless, many utilized models are either inadequate or overly complex, featuring processes that lack clarity and consistency, which hinders their ability to accurately depict real-world scenarios. In this study, we introduce an approach that applies deep learning methods involving a combination of Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) to enhance sketch recognition performance. In the initial phase of our approach, a CNN was employed to extract features that were subsequently forwarded to an LSTM network for classification. We evaluated the efficacy of our method by utilizing the QuickDraw dataset offered by Google, and the results demonstrated that our approach outperformed both CNN and LSTM, as well as other state-of-the-art methods. Our method attained an accuracy of 95%, with precision and recall reaching 95%, while also achieving an F1 score of 94%.

Keywords: Sketch recognition, Convolution neural network, Recurrent neural network, LSTM, features

1. Introduction

Freehand sketching is a prevalent daily practice used to convey thoughts, record occurrences, and communicate with others [1]. The increasing use of touch screens in wearable devices has heightened interest in the ability to recognize sketches. However, the requirements for automatic interpretation of hand-drawn sketches are substantially higher. Several factors contribute to this distinction. One notable contrast lies in the abstract nature of sketches, which offer minimal information about shapes, whereas natural images are characterized by their abundance of color and texture details. The various painting styles employed by different individuals can make it challenging for computers to develop accurate representations of objects for tasks such as sketch recognition. Most existing methods for sketch recognition follow the traditional approach of extracting sketch descriptors, such as color, texture, and shape, and then training a classifier with these descriptors. However, these features are optimized for natural images and are not specifically tailored to freehand sketches. Sketches tend to be highly iconic and abstract, featuring fewer visual cues than their natural-image counterparts.Deep-learning (DL) approaches, such as visual recognition on large-scale challenging datasets, have proven successful in various areas of computer vision in recent years [2]. DL approaches have been shown to enhance sketch-based recognition and generate relevant feature representations by examining large sketch datasets, including TU-Berlin and QuickDraw datasets [3][4]. DL can produce more distinct characteristics from sketch photos and can be used for sketch classification or recognition. Deep features were first used for sketch recognition by [5], who created a unique neural-network model. However, these methods based on deep learning outperform the traditional approaches in terms of efficiency. In this study, we propose an approach to improve sketch recognition through the utilization of deep learning methods. More precisely, we applied a CNN to extract valuable features from the sketches, and an LSTM network to classify them, which marks a significant retreat from the conventional feature-based techniques.Our contributions to the sketch recognition field can be summarized as follows:
- Integration of DL: We harness the power of DL, specifically a CNN, to extract noteworthy features from freehand sketches. This enables us to capture the unique and abstract qualities of sketches, which are markedly different from conventional natural images.
- Sequential Analysis with LSTM: We utilized an LSTM network to analyze the sequential nature of the extracted

E-mail address: laleelmouna@gmail.com, silkan_h@yahoo.fr, ctekouaboukoumetio@gmail.com, youssef.hanyf@gmail.com, Cheikhtouradmohamedou@gmail.com, mohamedade@gmail.com

sequence and progression of strokes in a sketch, thereby improving the ability of our model to recognize complex drawings.
- QuickDraw Dataset Evaluation: We thoroughly evaluated our proposed approach using the QuickDraw dataset, which contains a broad range of sketch classes and is one of the most extensive and comprehensive resources available for sketch-recognition research. This ensures the robustness of the proposed approach. By combining the power of deep learning, CNN, and LSTM, our approach strives to significantly enhance the accuracy of sketch recognition, thereby providing a prospective solution for overcoming the difficulties presented by the abstract and symbolic qualities of freehand sketches. The results achieved in this investigation highlight the efficacy of the proposed approach, underscoring its superior performance compared to CNN, LSTM, and other state-of-the-art methods. Section 2 provides an overview of the relevant literature, detailing previous research efforts in the field. Section 3 elaborates on the approach employed for sketch recognition, outlining the utilization of CNN and LSTM networks. Section 4 describes and analyzes the experimental results obtained using the QuickDraw dataset. Finally, in section 5, we present the conclusions drawn from this research.

2. Related Work

Several studies have focused on recognizing the sketches. These inquiries can generally be categorized into two primary lines of investigation: traditional sketch recognition [6] and approaches rooted in artificial intelligence [5]. It is evident that traditional methods, having a longer history, prove to be less efficient compared to artificial intelligence-based methods when it comes to addressing this issue. In Table 1, we present a comprehensive summary of diverse research efforts within the domain of sketch recognition, organizing them by publication year, dataset utilized, employed methodologies, and performance metrics applied.

In traditional sketch recognition, the user draws a request. The request sketch is described by a set of features including color, texture, and shape [8]. The similarity between image sketches was calculated using distance or similarity measures. Eitz et al. [9] introduced the BOW representation for freehand sketches and subsequently applied multiclass SVMs to to recognize input samples. In [7][10], the concept of local features was explored by researchers who employed a comprehensive structure, known as a star graph, is used to depict a sketch. In a different study, Schneide et al. [11] employed the Fisher vector as a sketch descriptor, which resulted in significant improvements in classification outcomes. Li et al. [12] introduced a novel approach in their research by proposing a criterion that aimed to enhance the average value of trace ratios through the use of linear discriminant analysis in a manner that maximized the harmonic weighted mean, this method demonstrates versatility and can be applied to various classification problems. Chang et al. [13] introduced an innovative semantic pooling strategy that excelled in addressing complex analytical tasks, specifically in the realms of case detection, recognition, and event recounting. One of the disadvantages of conventional approaches is the decline in performance as the size of the dataset grows [14]. In recent years, inspired by the accomplishments of deep learning in elevating the capabilities of computer vision, many researchers have worked towards enhancing specialized deep models for the recognition of sketches. Zhang et al. [15] presented a hybrid CNN architecture that combines two components: A-Net, which is dedicated to capturing appearance details, and S-Net, which focuses on the shape characteristics. The researchers evaluated the performance of this hybrid CNN model for several tasks, including image classification and sketch retrieval, utilizing multiple datasets: TU-Berlin, Flickr15k, and Sketchy. The results were remarkable, as The model achieved an accuracy of 80% on the Sketchy dataset and 83.5% on the TU-Berlin dataset. Zhang [16] introduced a technique that employs a convolutional dual channel neural network. The approach started with the refinement of the sketch to give it a sleek and polished appearance. Subsequently, contours were extracted using a dedicated contour extraction algorithm. The CNN then processes the input by integrating the refined sketch with the extracted contours through a fully connected layer, enabling feature fusion. The recognition rate achieved using a softmax classifier to generate classification results was 73.24%. Zhu et al. [17] introduced a network, combining both attention and dense elements, with a specific focus on optimizing sketch classification. This network handles the unique sparse characteristics of sketches by incorporating large-scale overlapping pooling. Dense blocks were strategically integrated into the central convolutional layer to foster feature reuse, and these blocks leveraged mixed attention mechanisms to effectively capture both localized and intricate details. Furthermore, the inclusion of center loss in combination with softmax cross-entropy loss was employed to enhance the overall classification performance. Li et al.[18] introduced a novel network architecture called Sketch-R2CNN, with a focus on optimizing vector sketch recognition. This unique architecture leverages a rasterized one-branch RNN to effectively harness vector sketches. The network analyzes a vector-based sketch and uses an RNN to detect the important feature points in the vector space.Subsequently, a neural module equipped with a line network is employed to transform the vector features and points into a multichannel point feature map. This map is then fed into a CNN to extract convolutional features within the pixel space. A line-network neural module was designed to facilitate end-to-end learning. To evaluate the efficacy of their approach, the authors conducted experiments on two datasets: the TU Berlin dataset and QuickDraw. Wu et al.[19] proposed a solution for the problem of segmenting sketches at the stroke level. Their strategy views this issue as an assignment involving the generation of one sequence to another. employing a method called SketchSegNet that relies on an RNN. This method was used to transform stroke sequences into the corresponding semantic labels of the in-

TABLE I. Machine Learning Methods and Performance Metrics

Ref.	Year	Methods	Dataset	Performance Metrics
[4]	2020	CNN	Quickdraw	Accuracy
[5]	2015	DNN	TU-Berlin	mAP
[7]	2013	Star graph, KNN, SVM	TU-Berlin	Accuracy
[8]	2009	ECSS map approach	SQUID database	Precision and recall
[9]	2012	KNN, SVM	TU-Berlin	Accuracy
[10]	2015	Multiple Kernel Learning (MKL)	TU-Berlin	Accuracy
[11]	2014	Stargraph, MKL	TU-Berlin	Accuracy
[12]	2017	Linear discriminant analysis (LDA)	Coil20 dataset, Umist, JAFFE, YaleB, FERET, PIE and ORL	Average classification error rate and standard deviation
[13]	2016	NI-SVM	MED14, MED13, CCVsub datasets	mAP
[14]	2016	Shape-similarity-retrieval-method	SQUID, MPEG-7, COIL-100 databases	Precision and recall
[15]	2020	Hybrid CNN	TU-Berlin, Sketchy, Flickr15k	Accuracy, MAP
[16]	2021	CNN	TU-berlin, COAD dataset	Accuracy
[17]	2021	CNN	TU-Berlin	Accuracy
[18]	2020	Sketch-R2CNN	TU-Berlin, QuickDraw datasets	Accuracy
[19]	2018	SketchSegNet (RNN)	QuickDraw dataset	Accuracy
[20]	2021	MGT	QuickDraw dataset	Accuracy
[21]	2018	CNN, RNN	QuickDraw dataset	Accuracy, precision, recall, MAP
[22]	2019	LSTM with Attention Mechanism and Minimum	Quick Draw Dataset	MAP@3, Accuracy
[23]	2018	KNN, CNN	Quick Draw Dataset	MAP@3, Accuracy
[24]	2022	ResNet	Tu-Berlin	Accuracy
[25]	2023	Light-SRNet	TU-Berlin, Sketchy, QuickDraw datasets	Accuracy

dividual components. The authors also presented a full-scale dataset for the segmentation of sketches at the stroke level consisting of 57,000 annotated freehand human sketches obtained from QuickDraw. Their experimental findings showed that their approach attained an average accuracy of over 90% for feature labeling on this newly proposed dataset. Xu et al. [20] presented an innovative multigraph neural network (GNN) that was designed to explore how sketches are represented across various graphs, The image captured both temporal information and the general and local geometric contours at the same time. To demonstrate the efficiency of the proposed method, the authors performed comprehensive numerical assessment of the sketch-recognition challenge. Specifically, MGT was applied to 414k sketches generated from Google QuickDraw. Xu et al. [21] introduced a deep hashing technique for sketch retrieval. They investigated sketch features, which have been under-explored in previous work, by utilizing a large dataset of human sketches (the QuickDraw dataset[23]) containing 3.8 million samples. They addressed the challenge of abstraction by introducing an innovative hashing loss function based on Hamming space.This loss function was designed to produce more condensed feature sets for the individual sketch categories. They also emphasized the investigation of temporal feature ordering by employing a two-branch network that combines both a CNN and RNN. Nguyen et al [22] presented a novel sequential model that incorporates a combination of CNN and LSTM along with attentional mechanisms [14]. This model has been applied in the field of image analysis, where researchers have employed multiple pretrained models in ImageNet to identify doodles. Furthermore, they implemented a multimodel integration strategy employing a minimum-cost flow algorithm. They then assessed its effectiveness on the QuickDraw dataset, which comprises millions of sketches organized into 340 distinct categories. The dataset was generated using the popular online game, QuickDraw! where players are tasked with sketching objects belonging to specific categories within a 20-second time frame. The sketches are stored as time-stamped vectors, with metadata showing the geographic locations of the cued object and player. Guo et al.[23] employed CNNs alongside k-nearst neighbors(KNN) algorithms for the classification of hand drawn sketches. The CNN architecture features three convolutional layers, followed by three fully connected layers, ultimately resulting in a softmax layer. This CNN model was trained on a subset of the ”Quick,Draw!” dataset and demonstrated remarkable performance, obtaining a top-1 accuracy of 76.7% and a top-5 accuracy of 92.9% on an independent test set demonstrates outstanding performance. Simultaneously, researchers applied a KNN classifier to the same dataset subset, utilizing features extracted from the final fully connected layer of the CNN. Interestingly, the KNN classifier yielded a top-1 accuracy of 64.8% and a top-5 accuracy of 85.1% on the identical test set. These findings underscore the superiority of CNN over the KNN classifier in terms of classification accuracy. Nonetheless, it is essential to acknowledge that the KNN classifier offers the advantage of interpretability, providing insights into the key features that influence the classification process [26].Wang et al.[24] introduced a novel ’hierarchical residual network’ alongside a concise triplet-center loss mechanism for the identification of sketches. Their research comprises of three primary contributions. First, they designed a multiscale residual block that outperformed other residual blocks in capturing multi-scale information, exceeding the capabilities of traditional basic residual blocks, while also requiring fewer learning parameters. Second, they constructed a hierarchical residual system by stacking these multiscale residual blocks, which yielded more comprehensive features than a single-level residual system. Finally, they attended to a concise triplet-based loss specifically formulated to focus on the sketch recognition issue, aiming to resolve the problems of limited interclass distance and excessive intraclass similarity. Their proposed approach underwent comprehensive testing using the Tu-Berlin benchmark dataset,which encompasses 20,000 instances distributed across 250 categories representing daily objects. The experimental outcomes emphasize the superiority of their technique over previously established methods. It is particularly worth noting that their approach achieved a remarkable accuracy rate of 88.2% and successfully demonstrated the effectiveness of their innovative hierarchical residual network accompanied by the concise triplet-center loss for the accurate recognition of sketches. Hou et al. [25] presented Light-SRNet, a convolutional neural network aimed at achieving precise sketch recognition while retaining its lightweight nature. It incorporates a mechanism of dual attention, harnessing both spatial and channel-focused attention mechanisms within the feature-extraction process to increase the discibing potential of extracted features. The authors conducted assessments of Light-SRNet across three datasets– TU Berlin, Sketchy, and QuickDrawExtended– and reported an experimental recognition accuracy of 73.14%. In the next section, we present our method for improving recognition performance by combining the CNN and LSTM models.

3. Proposed Approach

The proposed model analyzes the combination of a CNN for the extraction of deep features and an LSTM for sketch recognition by incorporating the extracted features. We begin by providing a brief overview of CNN and LSTM.

A. cnn

In the late 1980s, Yann LeCun was the innovator behind the introduction of Convolutional Neural Networks (CNNs) [27], representing a specialized class of neural network architectures renowned for their remarkable effectiveness in a large variety of computer vision approaches, particularly in classification and image recognition tasks [28]. CNNs are widely employed in various applications, including face recognition [29], video description [16], and 3D object retrieval through 3D sketching [10]. The CNN architecture generally comprises three main layers: the input layer, the hidden layer, and the output layer. The input layer is typically represented as a three-dimensional array and forwarded to the convolutional layer for further processing, in which the array dimensions are defined by the height, width, and channel count. The input $x = (x_t)_{t=0}^{N-1}$ is one-dimensional and without initial zero padding. When it is fed into a convolutional layer with a collection of M_1 three-dimensional filters (denoted as w_1^h for $h = 1, \ldots, M_1$), it produces a feature map. These filters are used on all the input channels [30]:

$$a^1(i,h) = (w_h^1 \times x)(i) = \sum_{j=-\infty}^{\infty} w_h^1(j) \times (i-j) \quad (1)$$

Where $w_h^1 \in R^{1\times k\times 1}$ and $a^1 \in \mathbb{R}^{1\times(N-k+1)\times M_1}$. The output of the first layer is generated by applying a nonlinear activation function $h()$ to the output of the input channel, yielding $f^1 = h(a^1)$. The hidden layer consists of a convolutional layer, a pooling layer, and a fully connected layer. Features from raw or intermediate feature maps are extracted by the convolutional layer using learnable filters [31]. In the convolutional layer, neurons initially establish connections with a reduced set of neurons in the subsequent layer, and the filter employs a weight matrix divided into segments to execute a convolution operation. Importantly, the weights undergo updates throughout the learning procedure [32]. The pooling layer introduces a transformation, in which all values within the pooling window are consolidated into a singular value. This operation encompasses maximum pooling, which involves selecting the highest value from each subfield within the preceding layer [31]. This layer not only decreases the size of the input layer but also plays a significant role in minimizing the computational burden during the learning phase and countering overfitting issues [32].

In the hidden layer $l = 2, \ldots, L$, the input feature map $f^{(l-1)} \in \mathbb{R}^{1\times N_{l-1}\times M_{l-1}}$ is denoted, where $1 \times N_{l-1} \times M_{l-1}$ represents the dimensions of the output filter map from the previous convolutional layer. This output filter map has dimensions $N_{l-1} = N_{l-2} - k + 1$. It undergoes convolution with a collection of M_1 filters, represented as $w_h^1 \in \mathbb{R}^{1\times k\times M_{l-1}}$, for $h = 1, \ldots, M_1$. This convolution results in the creation of a feature map designated as $a^1 \in \mathbb{R}^{1\times N_l\times M_l}$, following the process outlined in [30].

$$a^1(i,h) = (w_h^l \times f^{(l-1)})(i) = \sum_{j=-\infty}^{\infty} \sum_{m=1}^{M_{(l-1)}} w_h^l(j,m)\, f^{(l-1)}(i-j,m) \quad (2)$$

The final layer of a CNN is a fully connected layer that compresses the high-level features extracted by the convolutional layers into a single vector and generates the output. The feature values generated in the previous steps are then subjected to nonlinear activation functions to yield the final output, represented as $f^1 = h(a^1)$. The dimensions of the output matrix, f^L, are determined by both the filter size and the number of filters used in the final layer [30]. In Figure 3, we can observe the basic structure of a CNN, which mainly consists of a convolutional layer, a max pooling layer, and a fully connected layer.

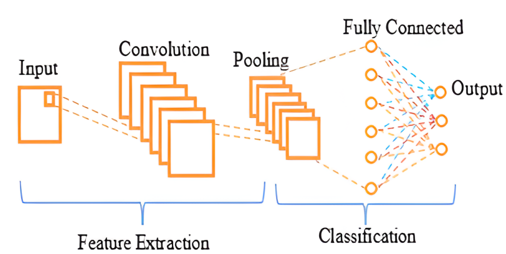


Figure 1. Basic structure of the convolutional network [2]

B. lstm

LSTM neural networks [20] represent a specific variant of recurrent neural networks (RNNs). These networks excel at understanding long-term dependencies by utilizing feedforward connections. While conventional RNNs aim to address the deficiency in memory found in typical feedforward neural networks—a shortcoming responsible for lackluster outcomes in tasks involving sequences and time series—they implement cyclic connections within their hidden layers. This architecture allows them to retain short-term memory, enabling the assimilation of information from sequential and time-based data.However, RNNs are plagued by the widely recognized issue of vanishing gradient, a predicament that curtails the model's capability to learn distant relationships within the data. LSTMs have been devised to overcome this challenge. They achieve this by conserving pertinent information within memory cells while purging extraneous data. This fundamental modification generally leads to enhanced performance compared with traditional RNNs. An LSTM unit consists of essential components: a memory cell, input gate, output gate, and forget gate (refer to Figure 2 for an illustration).

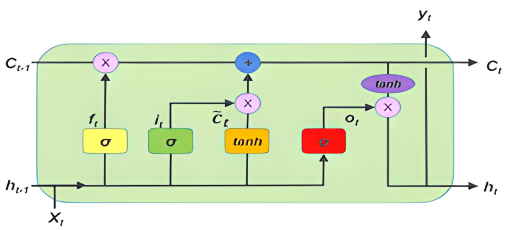


Figure 2. The anatomy of an LSTM cell [33]

Through this architecture, the LSTM achieves the ability to manage the flow of information in a deliberate manner, determining what should be "forgotten" and what should be "remembered." This capability enables LSTMs to grasp long-term relationships. Specifically, the input gate, denoted as i_t, and the subsequent gate c_t^*, regulate the introduction of fresh information into the memory state c_t at time t. Mean while, the forget gate f_t governs the treatment of past data in the memory cell at time t, deciding whether it should be retained or discarded. Simultaneously, the output gate σ_t directs the utilization of information for the memory cell's output. In summary, Equations (2)–(5) concisely describe the operations executed by the LSTM unit.

$$i_t = \sigma(U_i x_t + W_i h_{t-1} + b_i) \tag{3}$$

$$f_t = \sigma(U_f x_t + W_f h_{t-1} + b_f) \tag{4}$$

$$c_t^* = \tanh(U_c x_t + W_c h_{t-1} + b_c) \tag{5}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot c_t^* \tag{6}$$

$$\sigma_t = \sigma(U_o x_t + W_o h_{t-1} + b_o) \tag{7}$$

where x_t represents the input, W and U are the weight matrices, and b_* denotes the bias term vectors. The sigmoid function is denoted as σ, and the $\odot$ operator represents element-wise multiplication. Finally, the hidden state h_t, which captures the memory cell output, is derived as follows:

$$h_t = \sigma_t \odot \tanh(c_t) \tag{8}$$

When multiple LSTM layers are organized in a sequence, the memory state c_t and hidden state h_t from each LSTM layer are transmitted as inputs to the subsequent LSTM layer in the stack.

C. cnn-lstm combination

CNN exhibit strong capabilities in intelligently extracting features from data, while remaining unaffected by frequency variations within the data. However, it is constrained by the presumption of input-output independence during processing. This disregard for inherent interfeature information leads to performance deterioration when handling time series data. On the other hand, LSTM demonstrated superiority in handling time-series data, particularly in capturing long-term dependencies within data sequences, which improved recognition accuracy. This advantage comes at the cost of a longer learning period compared to CNN, as LSTM must grasp the nonlinear relationships present in the data. To harness the strengths of both algorithms, we introduced an innovative approach that combines two models: a CNN is utilized for feature extraction from sketches, whereas an LSTM network serves as a classifier. Our study is distinguished as a pioneer in the field of sketch recognition by introducing a CNN-LSTM combination. Prior studies on sketch recognition have mainly relied on either CNNs or LSTMs individually, or alternative deep learning models [34][35] . Notably, no previous study has employed a CNN-LSTM combination for sketch recognition. We introduce a novel integration method in which the flattened output of the CNN serves as the input sequence for LSTM. This innovative approach enables the capture of both spatial and temporal information inherent in sketches, including the stroke sequence and progression. Our approach contrasts with existing approaches that utilize the CNN-LSTM combination, where the CNN functions as a feature extractor and the LSTM acts as a classifier for each time step. Our approach streamlines the process, reducing parameters and computations while avoiding redundancy and inconsistency in classification. Furthermore, our study presents an efficient fusion process that enhances the effectiveness of our approach compared with previous studies. Figure 3 illustrates the visual representation of the combined network proposed for sketch recognition. It showcases the integration of CNN for feature extraction and LSTM for classification in the context of sketch recognition. In the section dedicated to Convolutional Neural Networks (CNNs), the architecture begins with a convolutional layer consisting of 16 filters of size 3x3 and a rectified linear unit (ReLU) activation function. This is followed by a max-pooling layer with a 2x2 pool size to reduce the spatial dimensions. Next, a second convolutional layer is introduced with 32 filters of size 3x3 and ReLU activation. The second max-pooling layer further reduces the feature maps. Continuing the architecture, two additional convolutional layers are added, each containing 128 filters of size 3x3

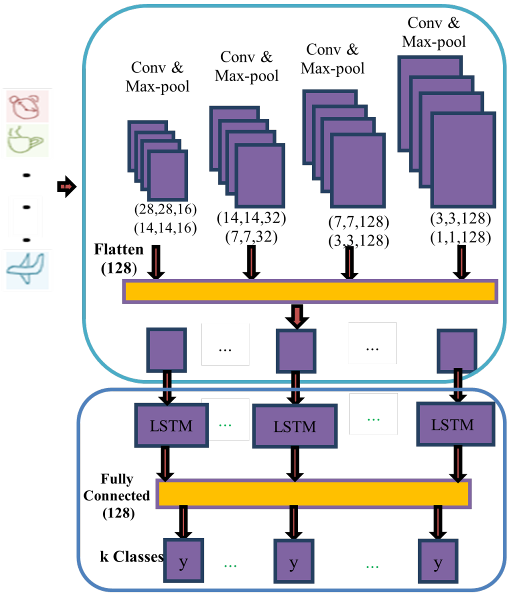


Figure 3. Visual representation of the combined network proposed for sketch recognition

and ReLU activation, to extract more complex features from the input sketches.After each of these convolutional layers, a max-pooling layer with a 2x2 pool size is applied to further down-sample the feature maps. Subsequently, a Long Short-Term Memory (LSTM) network is employed, where the flattened and expanded feature map from the CNN is fed into an LSTM layer comprising 32 units. This layer utilizes a hyperbolic tangent (tanh) activation function, which is critical for learning the long-term dependencies and sequential patterns in the data. Following LSTM, the architecture incorporates a fully connected layer comprising 128 units with a tanh activation function, further refining the extracted features for classification tasks. Finally, the model concludes with an output layer featuring a Softmax activation function. This layer is essential for transforming the feature representations into a probabilistic distribution across various sketch classes.The softmax function ensures that the sum of the probabilities is equal to one, with each probability representing the likelihood of the input sketch belonging to a specific class. The softmax function is defined as [36].

$$softmax(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \tag{9}$$

Where z_i represents the output score for the i-th class, and K is the total number of classes. This function calculates the probability of each class based on the scores in the output vector z. It exponentiates each score and divides it by the sum of all exponentiated scores to normalize the probabilities, ensuring they sum up to one. This normalization process transforms the output scores into a probability distribution across all classes. In the following section, we present and detail the experimental results, providing a comprehensive analysis of the effectiveness of our proposed approach in sketch recognition tasks.

4. Experiment Results

In this section, we describe our dataset, explain our implementation protocol, and present the experimental results.

A. data

The QuickDraw dataset, introduced by Dey et al. [37], is a substantial collection comprising over 50 million drawings grouped into 345 distinct categories. It consists of doodles created by millions of people worldwide as part of an online game developed by Google. In the game, users are prompted to draw a specific object or concept within a 20-second time limit, and these doodles are then used to train machine learning models. To evaluate our models, we selected 10 categories (cloud, sun, pants, umbrella, table, ladder, eyeglasses, clock, scissors, and cup) and randomly chose 10,000 sketches for each category. Figure 4 shows examples from the QuickDraw dataset.

Figure 4. Sample sketches from the QuickDraw dataset

The decision to use the QuickDraw dataset was driven by several factors specific to our research objectives and methodology. First, the QuickDraw dataset provides a diverse and extensive collection of sketches that encompass a wide range of object categories and drawing styles. This breadth aligns well with the aims of our study, which sought to develop a robust and generalizable sketch recognition model that is capable of handling various sketch types and complexities. Moreover, the QuickDraw dataset is much larger and more diverse than other datasets, containing 50 million sketches across 345 categories. This extensive dataset offers a more comprehensive and challenging resource for sketch-recognition research. In addition, the QuickDraw dataset covers a broader range of sketch classes and styles drawn by millions of people from various countries and cultures. This diversity enhances the realism and variability of the sketches, contributing to the robustness and generalizability of our proposed approach. Furthermore, the availability of such a large-scale dataset enables us to train our model comprehensively, ensuring its efficacy across diverse sketch categories.

B. implementation details

We evaluated the performance of three models - a simple CNN, LSTM models, and our proposed CNN-LSTM model

- using four performance metrics (Accuracy, Precision, Recall, and F1 Score) on the QuickDraw dataset, which was split into 80% for training and 20% for testing. We set the learning rate to 0.001 and trained the models for 50 epochs using Python and the Keras package in TensorFlow within the Google Colab environment

C. evaluation metric

We used four metrics to evaluate our models, which are defined by the number of true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN) in the predictions of the categorical model. Accuracy was calculated to evaluate the performance of the proposed approach, representing the degree to which the classifier can accurately categorize the data. The calculation is as follows:

$$Accuracy = \frac{TP + TN}{FP + FN + TP + TN} \tag{10}$$

Recall measures the ability of the model to identify all positive instances.

$$Recall = \frac{TP}{TP + FN} \tag{11}$$

Precision refers to the ability of the classifier to correctly identify positive instances without labeling negative instances as positive.

$$Precision = \frac{TP}{TP + FP} \tag{12}$$

The $F1$ score, which quantifies the balanced combination of precision and recall, yields results within the range of [0,1].

$$F1score = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{13}$$

D. results analysis

cnn results

The results of the performance evaluation of the CNN model are depicted in Figure 5. The evaluation process involved a comprehensive examination of the accuracy and cross-entropy (loss) metrics during both the training and validation stages. At the 50th epoch, the CNN achieved notable results, with training and validation accuracy rates of 90.6% and 90.5%, respectively. Impressively, the CNN architecture also yielded matching training and validation loss values of 0.3.

lstm results

Figure 6 illustrates the evaluation of LSTM performance during 50 epochs of training and testing. At epoch 50, the training accuracy was 92%, and the validation accuracy was 90%. Additionally, the training loss was 0.24, while the validation loss was 0.29.

cnn-lstm results

The performance of CNN-LSTM in sketch recognition is depicted in Figure 7, where the accuracy and loss (cross-entropy) curves for the training and test datasets are plotted.

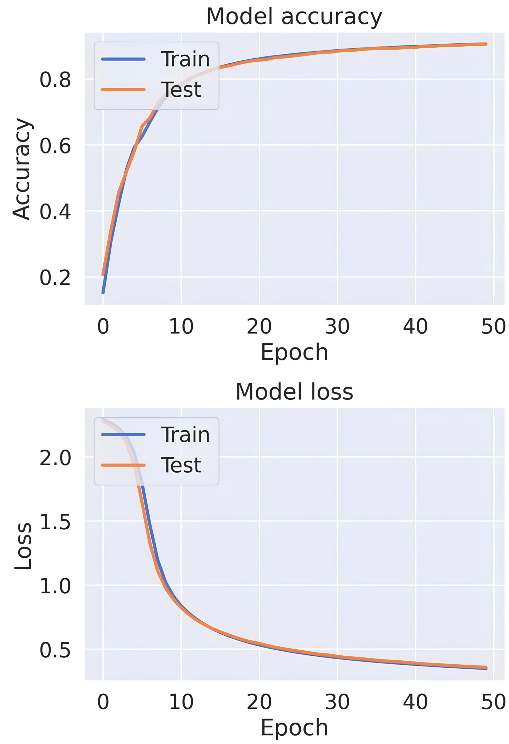


Figure 5. Performance curve on training and validation datasets of CNN model

The model demonstrated remarkable accuracy, achieving scores of 99% for the training set and 94% for the validation set after 50 epochs. The corresponding losses were 0.02 and 0.27, respectively. The experiment's findings unequivocally demonstrate the significant benefits of combining these two models to improve sketch recognition accuracy. The CNN-LSTM model, which integrates CNN and LSTM, outperforms either model individually in terms of training and validation accuracy.

The experimental results illustrated in the three figures (Figures 8, 9, and 10) provide insightful comparisons between the CNN, LSTM, and combined CNN-LSTM models regarding their responses to increasing training set sizes in terms of sketch classification accuracy. As depicted in Figure 8, the CNN model demonstrates a typical learning curve of machine learning models, with a sharp increase in accuracy initially, which then tapers off, showing diminishing returns on accuracy as the dataset size increases beyond 10K. The accuracy levels off after reaching approximately 40K, indicating that further data addition yields minimal improvement, reflecting a saturation point in learning from additional data. The LSTM model, shown in Figure 9, exhibits a gentler slope in accuracy improvement with increasing data size, suggesting that it may not leverage additional data as efficiently as the CNN model to enhance its predictive accuracy. The plateauing of accuracy starts earlier, at approximately 20K, indicating a potentially

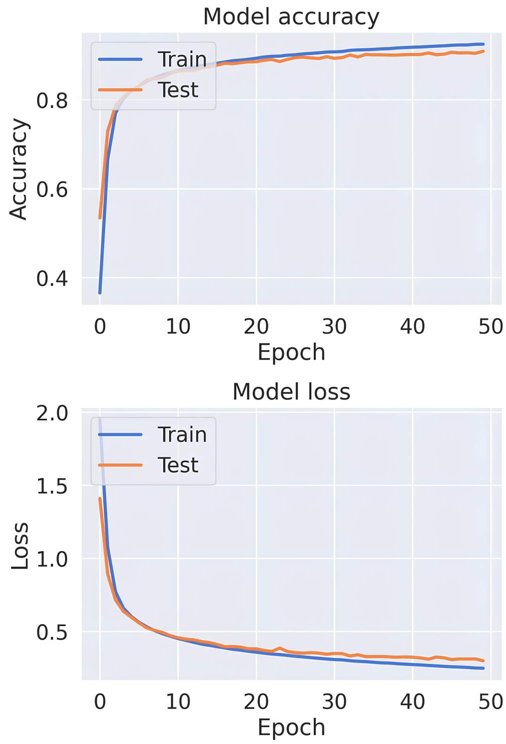


Figure 6. Performance curve on training and validation datasets of LSTM model

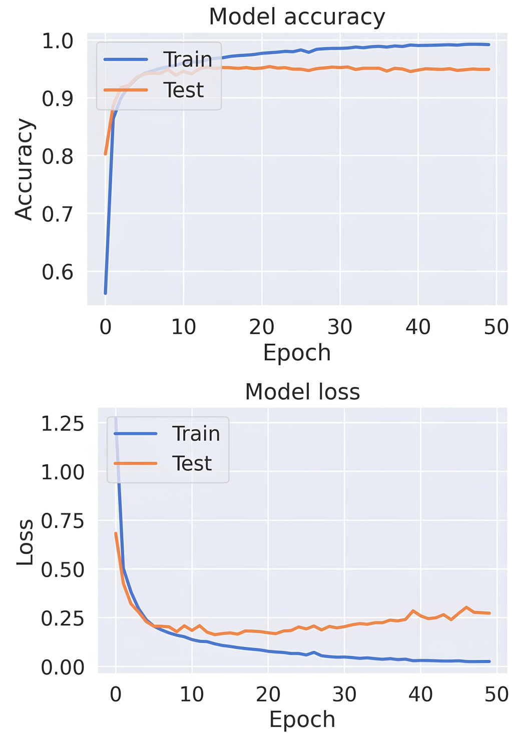


Figure 7. Performance curve on training and validation datasets of CNN-LSTM combination

quicker reach to its capacity limit for learning from more data compared to the CNN model. In contrast, the CNN-LSTM model, presented in Figure 10, illustrates an initial sharp rise in accuracy, similar to the CNN model, but begins to plateau at approximately 20K, akin to that of the LSTM model. This pattern suggests that while the combined model efficiently capitalizes on smaller datasets, akin to the CNN, its capacity to continuously learn from added data is more in line with LSTM's performance, reaching a plateau relatively early. These comparative insights highlight the importance of choosing the appropriate model based on the available training data size and the inherent capacity of the model to benefit from additional data. Although CNNs exhibit robustness in learning from larger datasets, LSTMs, and by extension, CNN-LSTMs may offer optimal performance with smaller to medium datasets, informing strategic decisions on model selection and data utilization to maximize sketch recognition accuracy.

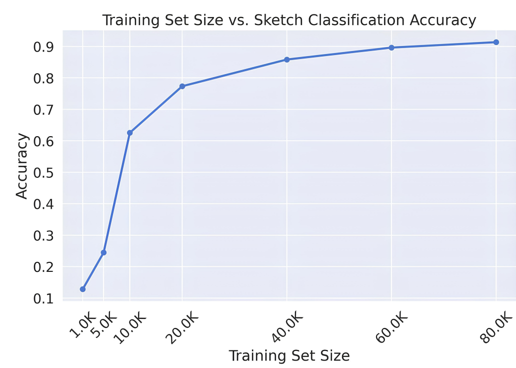


Figure 8. Training Set Size vs. Accuracy for the CNN model

TABLE II. Results of different models on QuickDraw dataset

Model	Accuracy	Precision	Recall	F1 score
CNN	90.56%	90.49%	90.44%	90.44%
LSTM	90.90%	90.94%	90.90%	90.90%
CNN-LSTM	94.94%	95.02%	94.99%	94.99%

Table 2 summarizes the results obtained by the different models on the QuickDraw dataset, and we found that the CNN-LSTM combination performs better than simple CNN and LSTM, which elucidates the impact of the combination on the results.

Table 3 presents the classification accuracy, precision, recall, and F1 score of the CNN-LSTM model in comparison with the CNN and LSTM models for each class, namely cloud, sun, pants, umbrella, table, ladder, eyeglasses, clock, scissors, and cup. We note that the performance of the CNN-LSTM model on the test set is superior to that of the CNN and LSTM models for all classes.

The CNN-LSTM model achieved the highest result of

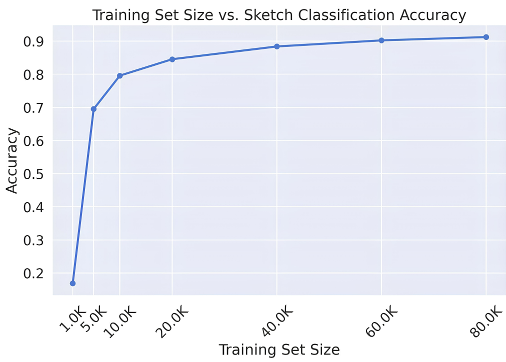


Figure 9. Training Set Size vs. Accuracy for the LSTM model

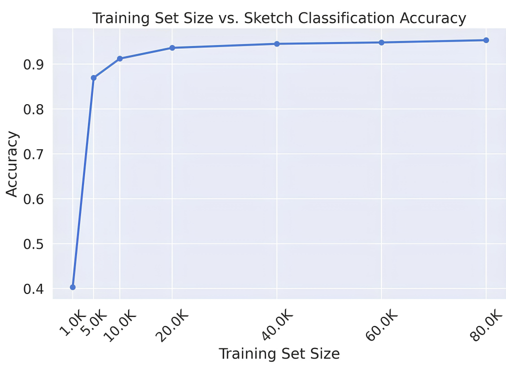


Figure 10. Training Set Size vs. Accuracy for the CNN-LSTM model

97.74% for the clock class. For the CNN model, the highest result was 94.64% in the umbrella class, whereas for the LSTM model, it was 94.82% in the same class.

The lowest results for CNN and CNN-LSTM were found in the cup class, at 83.70% and 91%, respectively, while for the LSTM model, it was 84.61% in the cloud class.

We compared our methods with the most recent and advanced methods on the QuickDraw database to evaluate their performances, as presented in Table 4. These approaches encompass a range of sketch recognition methods, which are extensively discussed in the second section of this article, titled "Related Works." Our methods, including CNN, LSTM, and the combined CNN-LSTM model, outperformed all other methods, exhibiting accuracy rates of 90.56%, 90.90%, and 94.94%, respectively.

TABLE III. Performance metrics of different models for each class with model comparison

Class	Metric	CNN Model	LSTM Model	CNN-LSTM Model
Cloud	Accuracy	86.63%	84.61%	93.31%
	Precision	87.51%	90.57%	96.01%
	Recall	86.62%	84.60%	93.30%
	F1 score	87.07%	87.49%	94.64%
Sun	Accuracy	93.49%	92.54%	97.63%
	Precision	88.83%	91.42%	94.98%
	Recall	93.48%	92.53%	97.63%
	F1 score	91.10%	91.97%	96.29%
Pants	Accuracy	93.85%	90.92%	94.39%
	Precision	93.98%	93.51%	96.35%
	Recall	93.84%	90.91%	94.38%
	F1 score	93.91%	92.19%	95.35%
Umbrella	Accuracy	94.64%	94.82%	96.77%
	Precision	92.90%	94.44%	96.18%
	Recall	94.63%	94.81%	96.76%
	F1 score	93.76%	94.63%	96.47%
Table	Accuracy	91.11%	92.30%	94.07%
	Precision	92.35%	89.95%	97.34%
	Recall	91.10%	92.29%	94.07%
	F1 score	91.72%	91.11%	95.68%
Ladder	Accuracy	89.85%	92.98%	96.23%
	Precision	91.75%	91.48%	95.10%
	Recall	89.84%	92.97%	96.23%
	F1 score	90.79%	92.22%	95.66%
Eyeglasses	Accuracy	89.04%	90.87%	92.22%
	Precision	83.63%	86.01%	90.85%
	Recall	89.04%	90.87%	92.21%
	F1 score	86.25%	88.37%	91.53%
Clock	Accuracy	94.18%	93.49%	97.74%
	Precision	94.82%	95.57%	98.17%
	Recall	94.17%	93.48%	97.73%
	F1 score	94.50%	94.51%	97.95%
Scissors	Accuracy	87.78%	88.27%	95.69%
	Precision	88.97%	88.90%	91.56%
	Recall	87.78%	88.27%	95.69%
	F1 score	88.37%	88.58%	93.58%
Cup	Accuracy	83.70%	88.10%	91.91%
	Precision	90.21%	87.52%	93.75%
	Recall	83.70%	88.09%	91.91%
	F1 score	86.83%	87.81%	92.82%

TABLE IV. Results of the state of the art on the Quickdraw dataset

Method	Accuracy
Xu et al [20]	70%
Li et al [18]	84.4%
Xu et al [21]	80.5%
Nguyen-Xuan [22]	90.4%
Kothawade et al [4]	65.57%
Guo et al [23]	60%
Our CNN	90.56%
Our LSTM	90.90%
Our CNN-LSTM	94.94%

5. Conclusions And Future Work

In this study, we proposed a novel approach that combines CNN and LSTM architectures for sketch recognition. The architecture employs a CNN to extract features and an

LSTM network as the classifier. Our constructed CNN-LSTM model includes several layers of convolution and max pooling, along with a single LSTM layer. We evaluated this method on a QuickDraw dataset by selecting ten categories, each containing 10,000 sketches. Our approach achieved a 94% accuracy rate for sketch recognition by leveraging the feature extraction capabilities of the CNN combined with the classification power of the LSTM. We also compared our proposed architecture with individual CNN and LSTM models to demonstrate its superiority. Additionally, compared with state-of-the-art techniques on the QuickDraw dataset, our approach outperformed them. Furthermore, we envision implementing decision-making systems based on this approach [38] [39]. Moving forward, we plan to extend this approach to address other sketch-related tasks, including sketch retrieval, sketch synthesis, and sketch simplification.

References

[1] W. Lu and E. Tran, "Free-hand sketch recognition classification," 2017, https://api.semanticscholar.org/CorpusID:28314302

[2] L. El Mouna, H. Silkan, Y. Haynf, A. Tmiri, and A. Dahmouni, "Comparative study of deep learning models for detection and classification of intracranial hemorrhage," in *International Conference on Business Intelligence*. Springer, 2022, pp. 122–131, DOI: 10.1007/978-3-031-06458-6_10.

[3] S. Hayat, K. She, Y. Yu, and M. Mateen, "Deep cnn-based features for hand drawn sketch recognition via transfer learning approach," *Editorial Preface From the Desk of Managing Editor*, vol. 10, no. 9, 2019, dOI: 10.14569/IJACSA.2019.0100958.

[4] D. Kothawade, I. Kazi, J. Mhatre, K. M. Nair, and D. P. Nitnaware, "Recognition of labels for hand drawn images," *International Research Journal of Engineering and Technology (IRJET)*, vol. 7, no. 5, pp. 1819–1823, 2020. [Online]. Available: https://www.irjet.net/archives/V7/i5/IRJET-V7I5351.pdf

[5] Y. Yang and T. M. Hospedales, "Deep neural networks for sketch recognition," *ArXiv*, vol. 1, no. 2, p. 3, 2015. [Online]. Available: https://api.semanticscholar.org/CorpusID:195345953

[6] Y. Li and W. Li, "A survey of sketch-based image retrieval," *Machine Vision and Applications*, vol. 29, no. 7, pp. 1083–1100, 2018, dOI: 10.1007/s00138-018-0953-8.

[7] Y. Li, Y.-Z. Song, S. Gong *et al.*, "Sketch recognition by ensemble matching of structured features," in *British Machine Vision Conference (BMVC)*, 2013, p. 1, dOI: 10.5244/C.27.35.

[8] H. Silkan, S. Ouatik, A. Lachkar, and M. Meknassi, "A novel shape descriptor based on extreme curvature scale space map approach for efficient shape similarity retrieval," in *2009 Fifth International Conference on Signal Image Technology and Internet Based Systems*. IEEE, 2009, pp. 160–163, dOI: 10.1109/SITIS.2009.35.

[9] M. Eitz, J. Hays, and M. Alexa, "How do humans sketch objects?" *ACM Transactions on Graphics (TOG)*, vol. 31, no. 4, pp. 1–10, 2012, dOI: 10.1145/2185520.2185540.

[10] Y. Li, T. M. Hospedales, Y.-Z. Song, and S. Gong, "Free-hand sketch recognition by multi-kernel feature learning," *Computer Vision and Image Understanding*, vol. 137, pp. 1–11, 2015, dOI: 10.1016/j.cviu.2015.02.003.

[11] R. G. Schneider and T. Tuytelaars, "Sketch classification and classification driven analysis using fisher vectors," *ACM Transactions on Graphics (TOG)*, vol. 33, no. 6, pp. 1–9, 2014, dOI: 10.1145/2661229.2661231.

[12] Z. Li, F. Nie, X. Chang, and Y. Yang, "Beyond trace ratio: weighted harmonic mean of trace ratios for multiclass discriminant analysis," *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 10, pp. 2100–2110, 2017, dOI: 10.1109/TKDE.2017.2728531.

[13] X. Chang, Y.-L. Yu, Y. Yang, and E. P. Xing, "Semantic pooling for complex event analysis in untrimmed videos," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 8, pp. 1617–1632, 2016, dOI: 10.1109/TPAMI.2016.2608901.

[14] H. Silkan, S. E. A. Ouatik, and A. Lachkar, "Extreme curvature scale space for efficient shape similarity retrieval," *International Arab Journal of Information Technology*, vol. 13, no. 6A, pp. 791–800, 2016. [Online]. Available: https://api.semanticscholar.org/CorpusID:5755386

[15] X. Zhang, Y. Huang, Q. Zou, Y. Pei, R. Zhang, and S. Wang, "A hybrid convolutional neural network for sketch recognition," *Pattern Recognition Letters*, vol. 130, pp. 73–82, 2020, dOI: 10.1016/j.patrec.2019.01.006.

[16] L. Zhang, "Hand-drawn sketch recognition with a double-channel convolutional neural network," *EURASIP Journal on Advances in Signal Processing*, vol. 2021, no. 1, pp. 1–12, 2021, dOI: 10.1186/s13634-021-00752-4.

[17] M. Zhu, C. Chen, N. Wang, J. Tang, and C. Zhao, "Mixed attention dense network for sketch classification," *Applied Intelligence*, pp. 1–8, 2021, dOI: 10.1007/s10489-021-02211-x.

[18] L. Li, C. Zou, Y. Zheng, Q. Su, H. Fu, and C.-L. Tai, "Sketch-r2cnn: An rnn-rasterization-cnn architecture for vector sketch recognition," *IEEE Transactions on Visualization and Computer Graphics*, vol. 27, no. 9, pp. 3745–3754, 2020, dOI: 10.1109/TVCG.2020.2987626.

[19] X. Wu, Y. Qi, J. Liu, and J. Yang, "Sketchsegnet: A rnn model for labeling sketch strokes," in *2018 IEEE 28th International Workshop on Machine Learning for Signal Processing (MLSP)*. IEEE, 2018, pp. 1–6, dOI: 10.1109/MLSP.2018.8516988.

[20] P. Xu, C. K. Joshi, and X. Bresson, "Multigraph transformer for free-hand sketch recognition," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 10, pp. 5150–5161, 2021, dOI: 10.1109/TNNLS.2021.3069230.

[21] P. Xu, Y. Huang, T. Yuan, K. Pang, Y. Z. Song, T. Xiang *et al.*, "Sketchmate: Deep hashing for million-scale human sketch retrieval," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8090–8098, dOI: 10.1109/CVPR.2018.00844.

[22] B. Nguyen-Xuan and G. S. Lee, "Sketch recognition using lstm with attention mechanism and minimum cost flow algorithm," *International Journal of Contents*, vol. 15, no. 4, pp. 8–15, 2019, dOI: 10.5392/IJoC.2019.15.4.008.

[23] K. Guo, J. WoMa, and E. Xu, "Quick, draw! doodle recognition," 2018, https://api.semanticscholar.org/CorpusID:155095704.

[24] L. Wang, S. Zhang, H. He, X. Zhang, and Y. Sang, "A hierarchical residual network with compact triplet-center loss for sketch recognition," *Multimedia Tools and Applications*, vol. 81, no. 11, pp. 15 879–15 899, 2022, dOI: 10.1007/s11042-022-12431-z.

[25] X. Hou, X. Rong, and X. Yu, "Light-srnet: a lightweight dual-attention feature fusion network for hand-drawn sketch recognition," *Journal of Electronic Imaging*, vol. 32, no. 1, p. 013005, 2023, dOI:10.1117/1.JEI.32.1.013005.

[26] L. El Mouna, H. Silkan, Y. Haynf, M. F. Nann, and S. C. K. Tekouabou, "A comparative study of urban house price prediction using machine learning algorithms," in *E3S Web of Conferences*, vol. 418. EDP Sciences, 2023, p. 03001, dOI: 10.1051/e3sconf/202341803001.

[27] Y. LeCun, B. Boser, J. Denker, D. Henderson, R. Howard, W. Hubbard, and L. Jackel, "Handwritten digit recognition with a backpropagation network," in *Advances in Neural Information Processing Systems*, D. Touretzky, Ed., vol. 2. Morgan-Kaufmann, 1989, https://api.semanticscholar.org/CorpusID:2542741.

[28] P. Kamencay, M. Benco, T. Mizdos, and R. Radil, "A new method for face recognition using convolutional neural network," *Advances*

in Electrical and Electronic Engineering, vol. 15, no. 4, pp. 663–672, 2017, dOI: 10.15598/aeee.v15i4.2389.

[29] H. Bi, Z. Liu, L. Yang, K. Wang, and N. Li, "Face sketch synthesis: a survey," *Multimedia Tools and Applications*, vol. 80, pp. 18 007–18 026, 2021, dOI: 10.1007/s11042-020-10301-0.

[30] A. Borovykh, S. M. Bohté, and C. W. Oosterlee, "Conditional time series forecasting with convolutional neural networks," *arXiv: Machine Learning*, 2017. [Online]. Available: https://api.semanticscholar.org/CorpusID:36367927

[31] R. Zuo, Y. Xiong, J. Wang, and E. J. M. Carranza, "Deep learning and its application in geochemical mapping," *Earth-science reviews*, vol. 192, pp. 1–14, 2019, dOI: 10.1016/j.earscirev.2019.02.023.

[32] E. Hoseinzade and S. Haratizadeh, "Cnnpred: Cnn-based stock market prediction using a diverse set of variables," *Expert Systems with Applications*, vol. 129, pp. 273–285, 2019, dOI: 10.1016/j.eswa.2019.03.029.

[33] A. A. Ismail, T. Wood, and H. C. Bravo, "Improving long-horizon forecasts with expectation-biased lstm networks," *ArXiv*, vol. abs/1804.06776, 2018, dOI: 10.48550/arXiv.1804.0677.

[34] Q. Yu, Y. Yang, Y. Song, T. Xiang, and T. Hospedales, "Sketch-a-net that beats humans," in *British Machine Vision Conference*, 2015, dOI: 10.1007/s11263-016-0932-3.

[35] D. Ha and D. Eck, "A neural representation of sketch drawings," in *International Conference on Learning Representations*, 2018, dOI: 10.48550/arXiv.1704.03477.

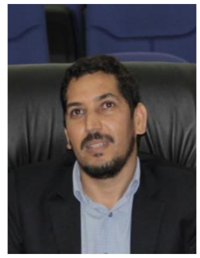
[36] F. Es-Sabery, A. Hair, J. Qadir, B. Sainz-De-Abajo, B. García-Zapirain, and I. De La Torre-Díez, "Sentence-level classification using parallel fuzzy deep learning classifier," *IEEE Access*, vol. 9, pp. 17 943–17 985, 2021, dOI: 10.1109/ACCESS.2021.3053917.

[37] S. Dey, P. Riba, A. Dutta, J. Llados, and Y. Z. Song, "Doodle to search: Practical zero-shot sketch-based image retrieval," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2179–2188, dOI: 10.1109/CVPR.2019.00228.

[38] M. C. Tourad and A. Abdali, "An intelligent similarity model between generalized trapezoidal fuzzy numbers in large scale," *International Journal of Fuzzy Logic and Intelligent Systems*, vol. 18, no. 4, pp. 303–315, 2018, dOI: 10.5391/IJFIS.2018.18.4.303.

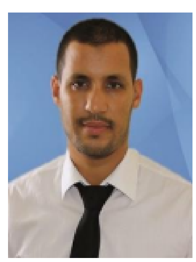
[39] M. C. Tourad, A. Abdali, and A. Outfarouin, "On a new index of publish/subscribe system in the context of big data," in *2016 IEEE/ACS 13th International Conference of Computer Systems and Applications (AICCSA)*. IEEE, 2016, pp. 1–2, dOI: 10.1109/AICCSA.2016.7945756.

Lale EL Mouna is currently a doctoral candidate in Computer Science, specializing in deep learning and computer vision, at Chouaib Doukkali University in El Jadida, Morocco. She holds a Master's degree in Data Science from the same university. Her research focuses on advancing deep learning models for image recognition and prediction. Lale is deeply passionate about applying state-of-the-art machine learning techniques to tackle real-world challenges in fields such as prediction modeling. She has previously contributed to research projects on image classification and neural network optimization. Email: laleelmouna@gmail.com



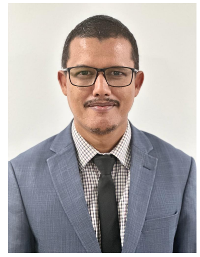
Hassan Silkan receives the PhD in computer sciences from Sidi Mohamed Ben Abdellah University, FSDM, Morocco. Currently, he is a professor in Chouaib Doukkali University, Department of Computer Science, Faculty of sciences El Jadida, Morocco. He published more than 32 papers in international journals and conferences in the fields of Shape Representation and Description, Similarity search, Content based images retrieval, Database Indexing, Multimedia Databases, and others. He can be contacted at email:silkan$_h@yahoo.frorsilkan.h@ucd.ac.ma$

Stéphane C. K. Tékouabou is currently an Assistant Professor in the department of Computer Sciences and Educational Technologies of the Higher Teaching Training College of the University of Yaoundé 1 in Cameroon. He was a Research scientist and a PostDoc Researcher at the Center of Urban System (CUS) of the Mohamed VI Polytechnic University (UM6P), Ben Guerir 43150, Morocco from 2020 to 2023. He is also an associate professor at the EIGSI Engineering school of Casablanca since 2020. He received his PhD degree in September 2020 from the Faculty of Sciences, Chouaib Doukkali University, 24000, El Jadida, Morocco. He obtained his State Engineering degree in 2016 from the National School of Applied Sciences of El Jadida (ENSAJ) of the same university. He is an associate editor of the African cities journal and reviewer of many journals including Sustainable Cities and Society (Elsevier, Q1, IF 10.696, SJR 2.02, CiteScore 14.4). He has also participated in the Scientific Committee of many international conferences. His research interests include computer vision (pattern recognition, detection, and classification problems); artificial intelligence; machine learning, deep learning, modeling and optimization, Urban Engineering, educational technologies



Youssef Hanyf holds a PhD degree in computer science from Chouaib Doukkali University, Morocco in 2017. He also received his B.Sc. (Mathematics and informatics) and M.Sc. (Software quality) from the same University, in 2009 and 2011, respectively.Currently, He is a professor of computer science at National School of Commerce and Management of Dakhla, Ibn Zhor University, Dakhla, Morocco. His research includes high dimensional data processing, Data structure, Image processing, information retrieval,similarity search, machine learning, and recommendation systems. He has published over 13 papers in international journals and conferences. He can be contacted at email: youssef.hanyf@gmail.com or y.hanyf@uiz.ac.ma.

Mohamedou El Ghotob Cheikh Tourad his Ph.D. in Computer Science at Cadi Ayyad University in Marrakesh, Morocco, in 2020. Presently, he serves as a Research Professor specializing in Artificial Intelligence and Data Science within the Faculty of Science and Technology at Nouakchott University. His scholarly pursuits revolve around various aspects of Artificial Intelligence, encompassing Natural Language Processing (NLP), Deep Learning, Similarity Learning, and the analysis of Big Data. He can be contacted at email Email: cheikhtouradmohamedou@gmail.com



Mohamedade Farouk Nanne is a teacher researcher in the Mathematics and Computer Science department at the Faculty of Sciences and Techniques, University of Nouakchott, Nouakchott, Mauritania. His area of research of interest includes artificial intelligence, conversational agents, network security, bioinformatics, and internet of things. He can be contacted at email: mohamedade@gmail.com.