Attention-Net: An Ensemble Sketch Recognition Approach using Vector Images

Gaurav Jain*, Shivang Chopra*, Suransh Chopra*, and Anil Singh Parihar*

Abstract– **For the past few decades, machines have replaced humans in several disciplines. However, machine cognition still lags behind the human capabilities. We address the machines' ability to recognize human drawn sketches in this work. Visual representations such as sketches have long been a medium of communication for humans. For artificially intelligent systems to effectively immerse in interactive environments, it is required that machines understand such notations. The abstract nature and varied artistic styling of these sketches make automatic recognition of drawings more challenging than other areas of image classification. In this paper, we use sketches represented as a sequence of strokes, i.e., as vector images, to effectively capture the long-term temporal dependencies in hand-drawn sketches. The proposed approach combines the self-attention capabilities of Transformers while effectively utilizing the long-term temporal dependencies through Temporal Convolution Networks (TCN) for sketch recognition. The confidence scores obtained from the two techniques are combined using triangular-norm (T-norm). Attention heat-maps are plotted to isolate the discriminating parts of a sketch that contribute to sketch classification. The extensive quantitative and qualitative evaluation confirms that the proposed network performs favorably against state-of-the-art techniques.**

Index Terms—**Sketch recognition, Transformers, Attention-mechanism and Development, Temporal convolution network, Visual Cognition**

I. Introduction

Human beings demonstrate an inherent intellectual capability to draw and apprehend wide-ranging ideas through sketches. The utility of sketches has merely evolved over time, from sculptures and carvings in ancient times to design blueprints in the modern era. Hence, to build artificially intelligent systems that are all-pervasive and create an immersive experience for humans, it is required that abstract concepts such as sketches must be understood. Sketch analysis has gained importance in recent years, with implications in understanding academic setups such as mathematics [1], chemistry [2]–[4], and electronics [5]. This has paved the way for further development in the field with applications like sketch segmentation [6]–[9], image retrieval [10]–[12], and sketch recognition [13]–[15]. Succeeding in these tasks would further extend the capabilities of machines to reach human perception capabilities, such as understanding human behaviour and actions [16].

In this work, we focus on *sketch recognition*, which aims to identify human drawings and classify them into their respective categories. Although existing methods provide satisfactory accuracy performance for image recognition [17], only a few approaches have addressed the issue of identifying human drawn illustrations [18]–[20]. Sketch recognition was first introduced by Sutherland *et al.* [21]. Since then, several approaches have been introduced for better sketch comprehension in this active area of research. Conventional studies [18], [19] consider sketches as nothing but images, converting them to binary representations and extracting features for training classifiers like Support Vector Machines (SVM). With the success of deep learning-based approaches in recent years for image classification tasks [17], these techniques have gained importance in sketch analysis as well. In particular, convolutional neural network-based approaches [14], [22] have outperformed the conventional techniques in sketch recognition with significantly dominant accuracy performances. However, most of the techniques often overlook the characteristic properties of sketches, and hence leave room for improvement.

Achieving artificial visual perception of sketches is particularly challenging due to (1) heterogeneous representations, and (2) level of abstraction. Representation of an object is contingent upon an individual's own interpretation of that object. This leads to a high intra-class and inter-class variation. Unlike images, sketches cross the boundary of what the object *actually* is, to what the object *may be interpreted* by a human. Hence, learning such uncertain, yet acceptable, depiction of an object proves to be a non-trivial task. Representation of sketches as images is inconsistent with the way sketches are drawn by humans. Hence, a more credible depiction of sketches is using the *vector image format*. At this abstract level, sketches constitute a sequence of short pen strokes that render a drawing over time. This representation proved to be an apt choice for sketch generation [23].

In order to address the aforementioned issues, an ensemble approach is proposed for sketch recognition in this work. The major contributions of the proposed approach are as follows:

- **Attention-based classification of sketches:** This is the first approach to the best of our knowledge that employs Transformers for sketch recognition. We leverage the attention mechanism of Transformers to identify objects. For this, sketches are interpreted as sequence of strokes. We isolate parts of sketches that contribute towards classification of an object.
- **Long-term temporal dependencies modeling of strokes:** Recurrent networks fail to process long-term correlations. Hence, we incorporate TCN in our network to trace long-term dependencies through dilated convolutions for vector images.
- **Ensemble learning:** Amalgamation of self-attention mechanism using Transformers and modeling of long-term sequential information using TCN is achieved using

*The authors contributed equally to this work.

The authors are with the Machine Learning Research Laboratory, Department of Computer Science and Engineering, Delhi Technological University, New Delhi 110042, India. (e-mail: gauravjain13298@gmail.com, shivangchopra11@gmail.com, suransh2008@gmail.com, parihar.anil@gmail.com)

T-norms. The score-level fusion ensures the two techniques to learn independent features, while combining scores to make a collective decision.

- **Parallel Architecture:** The proposed architecture independently trains the two modules. Despite having lots of trainable parameters, the individual components of our architecture are able to leverage parallel processing to enable simultaneous training thereby leading to a reduction in training complexity.

The rest of the paper is organized as follows: In Section II, related approaches are discussed. Section III presents the problem formulation and the mathematical notations used throughout the paper. In Section IV, the proposed approach is presented. Section V evaluates the proposed approach through a qualitative and quantitative analysis. Section VI discusses the ablation studies of the proposed approach, which is followed by concluding remarks in Section VII.

II. Related Work

Extensive research has been directed towards image recognition tasks [17], however, limited work has been done in the field of sketch recognition. The very first work in sketch recognition was proposed by [21]. Following which, several approaches have been proposed. These can be loosely classified into the following categories:

A. *Traditional Classifiers*

These methods process pixel-level images, or raster images, to extract local features that are used to train classifiers or guide learning strategies. For example, [13] extracted bag-of-features from sketches, which trains a Support Vector Machine (SVM) classifier. [18] represented sketches as Fisher Vectors. A Gaussian Mixture Model (GMM) was used to encode raster images of sketches. The deviation from the GMM, in terms of Fisher Vectors, was used to finally train an SVM classifier. Due to the absence of visual cues in sketches, simple classifiers fail to achieve reasonable recognition rates. To address this, [24] employed a star graph representation. In this, varying set of features were extracted and fused using a multi-kernel framework.

Conventional approaches have proved to be nothing more than a first step in the field of sketch recognition when compared with human recognition rates. These methods fail to model the casually drawn sketches that exhibit a wide-ranging diversity in styles and presentations. The complexity involved in sketch recognition is beyond the scope of simple classifiers, and thus researchers resort to deep learning based approaches.

B. *Convolutional Neural Networks (CNN)*

Similar to the conventional approaches, these methods process raster images for recognition. [25] employed a residual network for freehand sketch recognition. The authors conducted experiments with varying depth of networks and validated that deeper networks perform significantly better for sketch classification. In another work, [26] proposed *DeepSketch 3*, which employs CNN for partial sketch recognition, and sketch retrieval using query-by-example. Efforts to exploit the power of well-established CNN architectures, such as Alexnet [17] and Google net [27], have resulted in outperforming the classical methods. The very first approach to surpass human recognition accuracy was a CNN-based method [14]. However, the raster image representation required for these approaches fail to characterize these sketches at the right level of abstraction. Recently, [28] explored transfer learning approach for sketch classification. While [29] proposed to use both shape and appearance of sketches. For this, a hybrid-CNN was proposed to extract two sets of deep features for shape and appearance respectively. An SVM was trained using these sets of features to classify sketches.

Although CNN-based approaches have comfortably outperformed existing classifiers, there is scope for lots of improvement. There have been concentrated efforts towards extrapolating the power of existing deep learning models for new domains [30]. However, the thin stroke and absence of color channels leaves very little information for a CNN to process. Furthermore, while people can agree on what an object looks like, how they ultimately render the object can vary significantly, thereby leading to lots of artistic variation. This calls for an approach that can extract more than just the spatial information available in a raster image. To mitigate this, sequential information in sketches is exploited, which cannot be used for a photograph.

C. *Recurrent Neural Networks (RNN)*

Sequential information that sketches constitute is what makes their handling different from images. To exploit this piece of information, recurrent neural network offer a favourable solution for sequence modeling. *Sketch-RNN* [23] is the very first approach to corroborate the effectiveness of vector images for sketch generation. In this, an RNN-based encoder-decoder architecture models the temporal correlation between the previous pen strokes to generate the next pen stroke. These vector representations overcome the barrier of representation but display a considerable drop in performance when the number of object categories increase. Most RNN-based approaches constitute the Long Short Term Memory (LSTM) units. While LSTMs have been employed exhaustively for language modeling tasks, they fail to model the underlying distribution for sketches. This is mainly attributed to the inability of these units to model complex and large sequences of data. To this end, attention mechanism is used in tandem with LSTMs [31] and convolutional neural networks as well [32]. This ensures that the model focuses on certain parts of the input that form the most distinguishing features of an object class. However, LSTMs still fail to encode long term dependencies and adapt to variable-length inputs.

In sum, a vector image based sketch recognition approach is required, which can model long term dependencies along with both the temporal information effectively. To address this research gap, an ensemble approach for sketch recognition is proposed in this work. The details of which are described in the following section.

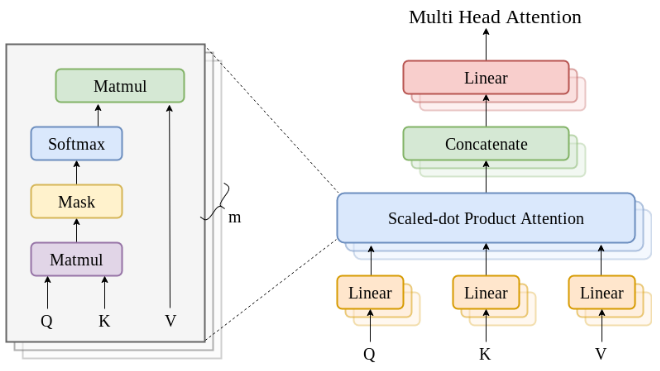

Fig. 1. Internal structure of the Multi-Headed Self-Attention Mechanism in a Transformer block explaining the various operations involved in obtaining the multi-head attention heat-maps .

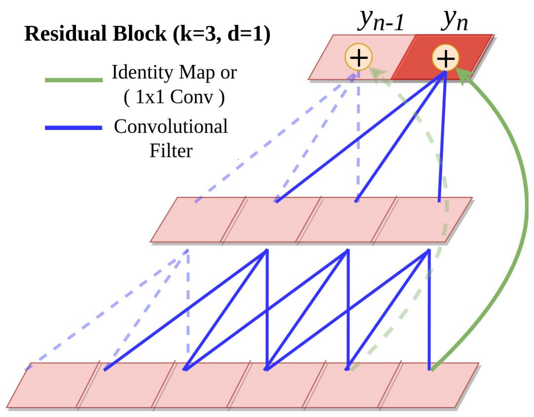

Fig. 2. Residual connections and dilated convolutions in TCN

III. Problem Formulation and Notations

The *Quick, Draw!* [23] dataset contains drawings in timestamped vector format. The main goal of our experiments is to leverage the temporal information in the timestamped data in order to classify sketches without rasterizing them into the corresponding image data.

Each sketch, $\mathcal{S}$, is represented as sequence of strokes s_i i.e. $\mathcal{S} = \{s_1, s_2, ..., s_n\}$ where n denotes length of sequence. Each stroke s_i is transformed to a 3-point format given by:

$$s_i = \{\Delta x_i, \Delta y_i, p_i\}, \quad \forall i \in \{1, 2, ..., n\} \tag{1}$$

Where (Δx_i, Δy_i) is the change in the x and y co-ordinates from the $(i-1)^{th}$ to the i^{th} timestamp, and p_i is a binary variable which denotes PenState.

$$p_i = \begin{cases} 0, & \text{if pen is lifted and moving to the next point} \\ 1, & \text{if pen is in contact with surface} \end{cases} \tag{2}$$

IV. Proposed Methodology

A. Preliminaries

1) Transformer

Previously, LSTMs were the only way to effectively capture the temporal nature of data. However, [33] proposed a novel architecture Transformer, which replaced the complex recurrent computations with attention mechanism. Transformers play a pivotal role in our architecture and have the added advantage of concurrent computations over LSTMs. Fig. 1 shows detailed view of self attention mechanism in a transformer. The attention function is a mapping from the d-dimensional input Values, V, onto the output $\mathcal{A}$, which is parameterised by a set of Key - Value (K, V) pairs. The Attention can essentially be defined as a weighted sum of Values $V = \{v_i | i \in \{1, 2, ..., d\}\}$ such that,

$$Attention(K, Q, V) = \text{SOFTMAX}\left(\frac{QK^T}{\sqrt{d}}\right)V \tag{3}$$

where $K, Q, V \in \mathbb{R}^d$. Furthermore, in order to capture multiple perspectives, instead of averaging the attention scores, we use multi-head self-attention mechanism. Linear projections of Queries, Keys and Values are taken in order to incorporate the different perspectives of the m attention heads which are then concatenated to form the final output.

$$MultiHead(Q, K, V) = \text{Concat}(\text{head}_1, \text{head}_2, ..., \text{head}_m)\mathbf{W}^O \tag{4}$$

where $\text{head}_l = \mathcal{A}(Q\mathbf{W}_l^Q, K\mathbf{W}_l^K, V\mathbf{W}_l^V)$, such that the project matrices $\mathbf{W}_l^Q, \mathbf{W}_l^K, \mathbf{W}_l^V \in \mathbb{R}^{\frac{d}{m} \times d}$, $\forall l \in \{1, 2, ..., m\}$, and $\mathbf{W}^O \in \mathbb{R}^{d \times d}$. The proposed network uses multi-head self-attention with $m = 8$ attention heads, along with $d = 128$. The parallel computation of these heads is analogous to performing a single attention. Hence, allowing the model to capture positional and temporal information without increasing time complexity.

2) Temporal Convolution Networks (TCN)

Convolution-based architectures have been shown to outperform recurrent neural networks in various sequence modelling tasks [34]. In their discussions, [34] propose a novel Temporal Convolution Network (TCN) which comprises of a 1-D fully convolutional network used in conjugation with casual convolutions in order to capture long term dependencies in the temporal data. Here, we attempt to describe the basics of TCN thereby giving the intuition behind choosing such an architecture for our experiments.

The receptive field of simple casual convolution is linearly dependent on the depth of the network. Dilated convolutions however allow an exponentially large receptive fields. Formally, for the sequence of strokes $\mathcal{S} = \{s_1, s_2, ..., s_n\}$, and a filter $f : \{0, ..., k-1\} \rightarrow \mathbb{R}$, the dilated convolution operation $\mathcal{F}$ on the i^{th} element s_i of the sequence is defined as:

$$\mathcal{F}(s_i) = (\mathcal{S} *_d f)(s_i) = \sum_{j=0}^{k-1} f(j) \cdot s_{i-d\cdot j} \tag{5}$$

where d is the dilation factor, k is the filter size and $s - d \cdot j$ accounts for the direction of the past. Furthermore, the large depths of TCNs induce instability in the architecture which is effectively tackled by addition of residual connection in the architecture as shown in Fig. 2.

3) Triangular Norm (T-norm)

Another central aspect of our research revolved around the conjunction of scores obtained through the Transformer and TCN based architectures. As demonstrated in the ablation study in Section V, end-to-end training of these architectures simultaneously affects the performance of the individual components. Hence, each of these components was trained individually on the data and the scores from these layers were finally combined using score fusion techniques. Here, we give a basic introduction of the various score fusion techniques used in our experiments.

As discussed in [37], T-norms are functions that map the unit squares onto the unit intervals i.e. $\mathcal{T}(A, B) : [0, 1] \times [0, 1] \rightarrow [0, 1]$. The T-norms provide infimum of the scores obtained from the various models, thereby leading to an improvement in the overall performance of our proposed architecture. The T-norms used as a part of our experiments are as follows:

- `Einstein Product`: $\mathcal{T}(A, B) = \frac{AB}{(2-A+B-AB)}$
- `Hamacher`: $\mathcal{T}(A, B) = \frac{AB}{(A+B-AB)}$
- `Yager` $(r > 0)$:
  $\mathcal{T}(A, B) = max(1 - ((1 - A)^r + (1 - B)^r)^{\frac{1}{r}}, 0)$
- `Frank` $(r > 0)$: $\mathcal{T}(A, B) = \log_r(1 + \frac{(r^A-1)(r^B-1)}{r-1})$

These methods prove fruitful in finding the interrelations between the scores obtained from the different models, all the while preserving the individual traits captured by these models.

B. Attention-Net: Overall Pipeline

Fig. 3 illustrates the proposed framework which constitutes multiple modules, (1) Feature Augmenter, (2) Transformer blocks, (3) Temporal convolution network module, and (4) Score fusion. Details of each module are as follows:

Feature Augmenter: The low feature dimensionality of the input data posed a significant problem while experimenting with the Transformer-based architectures. An important condition that needs to be satisfied while working with Transformers is that the number of input features d should be divisible by the number of heads in the multi-head attention module of the Transformer. The initial attempts at using Fully connected and Convolutional layers to increase the input dimensionality was met with limited success. Therefore, a convolution-based Augmenter was used in order to effectively capture the input representation and render a better-performing pipeline. The Augmenter module has two-fold functionality, (1) Extracting a latent vector representation $Z \in \mathbb{R}^{128 \times n}$ of our input data, and (2) Reshape the input to feed a larger context vector to the Transformer. Finally, the latent representation of the given input (Z) is derived from the middle layer (Bottleneck) of the Augmenter. This augmenter is trained independently on the input data and then the frozen weights of the encoder are used in our overall pipeline.

Transformer: Traditionally, Transformers [33] have been extensively used in language modeling tasks. We propose to leverage the proven ability of Transformers to capture long-term dependencies for our task of sketch recognition. First, the embedding layer is removed since semantic information is not required to be preserved by our model as opposed to the domain of Natural Language Processing. As proposed in the official implementation, we induce positional characteristics of our data into our model by using *coordinate embedding* [38]. In this embedding, a tuple with (position, time) is computed, instead of calculating just the position. $\mathcal{P}^t \in \mathbb{R}^{x \times d}$ is obtained by generating a sinusoidal position embedding with $1 \leq i \leq h$ positions, and $1 \leq t \leq n$ time steps for each vector-dimension $1 \leq j \leq d$:

$$\mathcal{P}^t_{i,2j} = sin\left(\frac{i}{10000^{2j/d}}\right) + sin\left(\frac{t}{10000^{2j/d}}\right) \tag{6}$$

$$\mathcal{P}^t_{i,2j+1} = cos\left(\frac{i}{10000^{2j/d}}\right) + cos\left(\frac{t}{10000^{2j/d}}\right) \tag{7}$$

Our overall pipeline consists of 10 blocks of Transformers with 8 attention heads each connected via Residual Connections as shown in Fig. 3. After passing the input through the coordinate embedding, the encoded input $\mathcal{P}$ is passed through the Transformer to obtain the point-wise attention heat-maps $\mathcal{A}_k, \forall k \in \{1, 2, ..., m\}$. Finally, to perform a classification task, we added a fully-connected layer on top of the final Transformer Block's output for the Start Token (s_0). The function of the Extract layer is to fetch features from the output corresponding to this Start token. In other words, if the output of the transformer block is of the dimensions $n \times d$, then the Extract layer's output is a vector of dimension d. This output is then passed through a fully connected layer with Softmax activation to obtain class probabilities $\Psi_A = \{\psi_{A_1}, \psi_{A_2}, ..., \psi_{A_c}\}$, where c represents the number of classes.

Temporal Convolution Network: In many of the temporal modelling tasks, Convolution Layers tend to outperform the conventional Recurrent-based architectures, including LSTMs and GRUs, due to their excellent ability to capture short as well as long-term dependencies in terms of n-grams. As described in [34], we propose to use a Temporal Convolution Network (TCN) alongside our Transformer-based architecture to enhance the performance of our overall pipeline. As shown in Fig. 3, the input $\mathcal{S}$ is passed through 3 layers of TCN. These layers are internally comprised of 4 Dilated Convolution layers connected via residual connections. The 4 Dilated Convolutional sub-layers have receptive fields of size $d = [1, 2, 4, 8, 16]$ respectively, thereby capturing temporal dependencies of upto 2^{16} timesteps. For each timestep i and layer j, the output of the individual residual blocks is as follows:

$$\mathcal{S}^j_i = ReLU(\mathcal{S}^{j-1}_i + \mathcal{F}(\mathcal{S}^{j-1}_i)) \tag{8}$$

For each layer j, we use a filters of size $k = 5$ and 128 kernels each. Finally, the output of the last cell of the final convolution layer $\mathcal{S}^3_n$ is picked and passed through a fully connected layer with Softmax activation to obtain class probabilities $\Psi_B = \{\psi_{B_1}, \psi_{B_2}, ..., \psi_{B_c}\}$, where c represents the number of classes.

Score Fusion: In the last phase of our pipeline, the class-wise probability distributions Ψ_A and Ψ_B obtained from the Transformer and TCN architectures are combined using various score fusion techniques. The two modules were initially

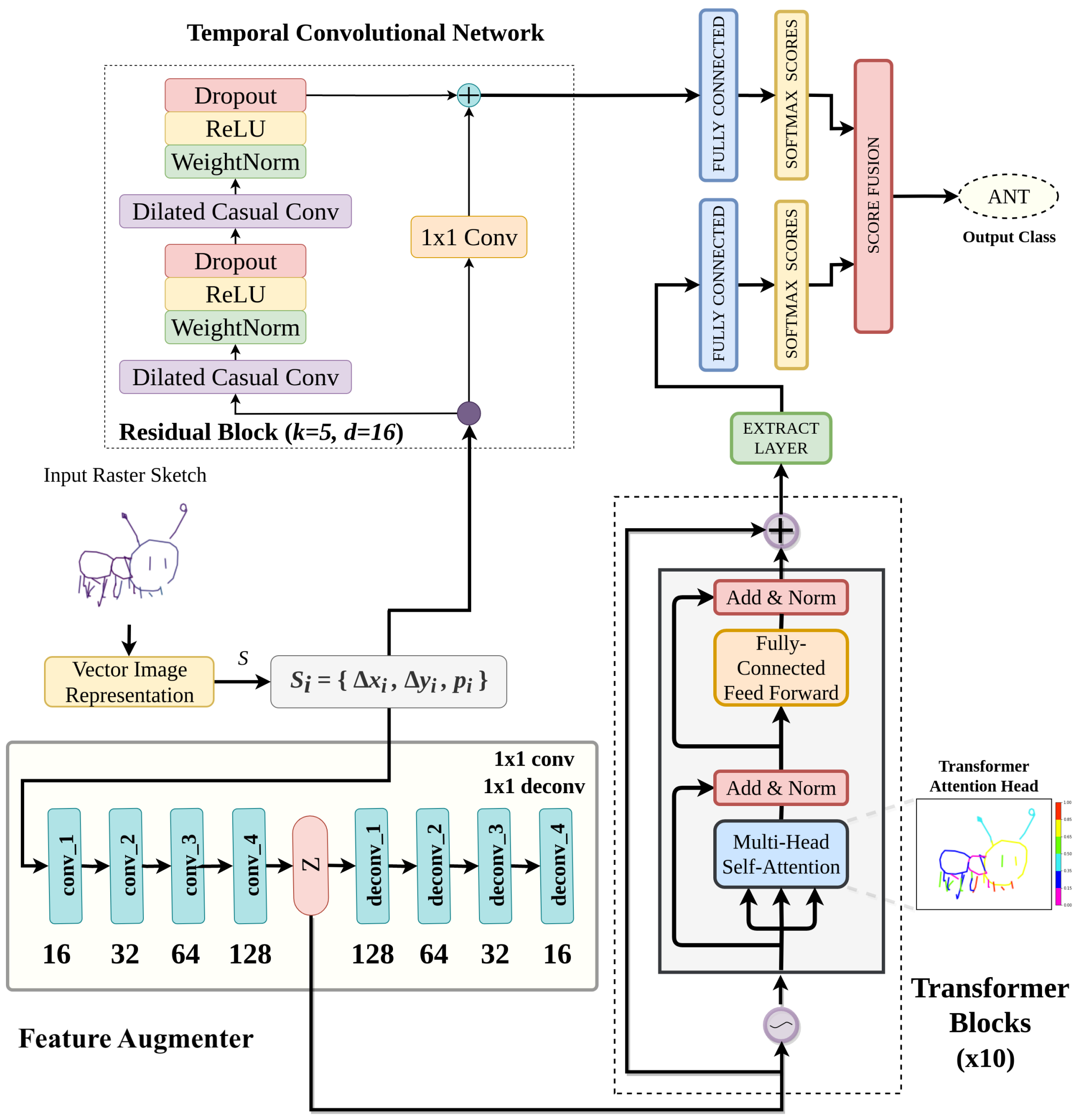

Fig. 3. Attention-Net: proposed architecture for ensemble sketch recognition using transformers and temporal convolution network.

trained from end-to-end. However, this led to a degradation in the performance of the overall pipeline. The proposed architecture therefore trains the individual modules independently and then combines the scores using various score fusion techniques. This led to the assignment of respective weights to the scores of individual modules and thereby leading to a far superior algorithm as compared to the individual modules. The final prediction probabilities were obtained as follows:

$$y = \mathcal{T}(\Psi_A, \Psi_B) \tag{9}$$

C. Implementation Details

The three component of our pipeline were trained separately and finally the trained models were combined together using various score fusion techniques. All modules shared the same training configuration. Early stopping on validation loss configured with $\delta_e = 0.00001$ and patience$= 10$ is used. Learning Rate Scheduler was used for dynamic training which defines the learning rate (lr_i) for i^{th} epoch as:

$$lr_i = ((lr_{beg} - lr_{min}) * \delta_{lr}^i) + lr_{min} \tag{10}$$

where $\delta_{lr} = 0.9$ is the decay rate, $lr_{beg} = 0.0001$ represents the initial learning rate, and $lr_{min} = 0.00001$ is the minimum learning rate. The hyper-parameters used in the network are fine-tuned for optimal performance using grid search.

V. Experiments

A. Dataset and Evaluation Metrics.

The *Quick, Draw!* dataset [23] is used to evaluate the recognition performance of the proposed approach. The dataset constitutes over 50 Million sketches spread around 345 object categories. Recognition accuracy is reported to validate the performance of the proposed method. A random train-test split of 80-20% is done, with 50,000 samples per class during

[1]Official TensorFlow implementation of RNN for *Quick, Draw!*

	Method	**Top-1**	**Top-5**	**Top-10**
Traditional Classifiers	HOG-SVM [13]	50.25 ± 2.05%	62.51 ± 0.64%	64.22 ± 1.64%
	Fisher-Vectors [18]	56.11 ± 1.34%	74.28 ± 1.00%	79.19 ± 0.33%
CNN-based Classifiers	AlexNet [17]	63.25 ± 0.95%	80.83 ± 2.07%	85.02 ± 1.36%
	ResNet50-CNN [35]	78.86 ± 2.18%	**91.10 ± 0.86%**	94.14 ± 0.97%
	Sketch-a-Net v2 [14]	74.84 ± 1.01%	87.19 ± 1.13%	91.25 ± 2.00%
	Graph Neural Networks [36]	73.20 ± 0.77%	85.35 ± 1.22%	89.05 ± 0.63%
RNN-based Classifiers	RNN-Tensorflow [1]	75.59 ± 2.04%	90.14 ± 1.27%	94.13 ± 2.08%
	Transformer	76.35 ± 0.96%	89.02 ± 0.84%	93.95 ± 1.38%
	Temporal Convolution Network	75.17 ± 2.22%	87.99 ± 1.20%	91.70 ± 1.54%
	Concatenate (TCN, Transformer)	73.21 ± 1.67%	88.02 ± 1.00%	90.41 ± 2.70%
Ensemble (Transformer + TCN)	Ensemble (`Max`)	77.98 ± 2.34%	88.12 ± 1.66%	94.21 ± 1.76%
	Ensemble (`Min`)	76.32 ± 1.90%	86.04 ± 1.22%	93.67 ± 1.67%
	Ensemble (`Average`)	78.14 ± 2.05%	88.44 ± 0.70%	94.17 ± 1.09%
	Proposed Ensemble (`Hamacher`)	78.24 ± 1.22%	88.93 ± 1.96%	94.44 ± 1.27%
	Proposed Ensemble (`Einstien`)	78.54 ± 2.02%	89.01 ± 2.65%	94.57 ± 1.30%
	Proposed Ensemble (`Yager`) r=1.6	78.88 ± 2.10%	90.33 ± 1.42%	**95.52 ± 1.57%**
	Proposed Ensemble (`Frank`) r=1.4	**79.33 ± 1.56%**	90.46 ± 0.90%	95.27 ± 1.71%

TABLE I
COMPARATIVE EVALUATION OF RECOGNITION ACCURACY ON THE *Quick, Draw!* DATASET, WITH (A) TOP-1, (B) TOP-5, (C) TOP-10 ACCURACY.

	Method	**5**	**20**	**50**
Traditional Classifiers	HOG-SVM [13]	75.21 ± 1.74%	66.79 ± 2.10%	63.22 ± 0.95%
	Fisher-Vectors [18]	79.53 ± 1.31%	75.80 ± 1.72%	72.90± 0.89%
CNN-based Classifiers	AlexNet [17]	77.18 ± 1.55%	75.22 ± 1.05%	73.06 ± 2.30%
	ResNet50-CNN [35]	96.47 ± 1.66%	90.06 ± 2.22%	86.20 ± 1.82%
	Sketch-a-Net v2 [14]	94.78 ± 1.25%	88.64 ± 1.63%	85.19 ± 0.78%
	Graph Neural Networks [36]	94.71 ± 1.00%	89.13 ± 1.34%	82.50 ± 1.25%
RNN-based Classifiers	RNN-Tensorflow [1]	95.58 ± 2.15%	87.33 ± 1.52%	83.98 ± 2.00%
	Transformer	96.21 ± 1.37%	90.31 ± 1.82%	88.72 ± 2.09%
	Temporal Convolution Network	95.98 ± 1.22%	90.21 ± 1.77%	86.54 ± 1.32%
	Concatenate (TCN, Transformer)	94.81 ± 1.67%	89.79 ± 1.27%	84.41 ± 1.93%
Ensemble (Transformer + TCN)	Ensemble (`Max`)	95.11 ± 2.22%	90.77 ± 2.36%	88.16 ± 1.16%
	Ensemble (`Min`)	94.71 ± 2.63%	87.21 ± 2.77%	85.77 ± 2.33%
	Ensemble (`Average`)	95.89 ± 2.01%	90.55 ± 1.52%	89.48 ± 2.16%
	Proposed Ensemble (`Hamacher`)	96.57 ± 1.11%	91.34 ± 1.33%	90.48 ± 1.45%
	Proposed Ensemble (`Einstien`)	96.66 ± 1.69%	90.73 ± 1.21%	89.13 ± 1.41%
	Proposed Ensemble (`Yager`) r=1.6	97.15 ± 2.13%	**92.32 ± 1.63%**	90.22 ± 1.33%
	Proposed Ensemble (`Frank`) r=1.4	**97.33 ± 1.65%**	91.24 ± 1.67%	**90.77 ± 1.01%**

TABLE II
COMPARATIVE EVALUATION OF RECOGNITION ACCURACY ON SUBSETS OF *Quick, Draw!* DATASET, WITH (A) 5 CLASSES, (B) 20 CLASSES, (C) 50 CLASSES.

training and 10,000 samples per class for testing. Average recognition accuracy is reported using a 5-fold cross-validation approach.

B. Comparative Evaluation.

An exhaustive quantitative comparison is performed to validate the performance of the proposed approach against (1) Conventional classifiers, (2) CNN-based, and (3) RNN-based techniques. Further, baseline score fusion techniques such as `Max`, `Min`, and `Average` are compared as well. The compared approaches are trained using the same dataset for fair evaluation. Table I reports the recognition accuracy of the proposed approach in comparison with the performance of the state of the art methods. We evaluate top-1, top-5, and top-10 recognition rates.

The traditional classifiers fail to model the complex and diverse features present in the information-deficient sketches. This exhibits a clear parallel with the poor performance of these conventional approaches. On the other hand, CNN-based approaches perform considerably better due to the large amount of trainable parameters that suffice to model the diversity in features and classes present in the challenging *Quick, Draw!* dataset. However, CNN-based approaches tend to fall short of the performance of RNN-based approach. This is because of the ability of RNN-based approaches to model the 'way' a sketch is rendered, i.e., the sequential information. The proposed approach, on the contrary, outperforms all the proposed classes of approaches due to the ability of extracting both temporal and spatial information.

The ensemble approach achieves higher accuracy than its individual counterparts as well, i.e., the Transformer and the TCN only networks. This validates our hypothesis as the efficacy of combining the attention mechanism and long-term temporal dependencies clearly translates into higher recognition rates.

To further evaluate the performance, three subsets of the *Quick, Draw!* dataset are formed containing 5, 20, and 50 classes respectively. For each subset, classes are selected randomly and average recognition accuracy for a 5-fold cross validation are reported. Table II reports the comparison results. As the number of classes increase, Attention-Net consistently outperforms the baseline approaches. Quantitative evaluation

Model Configurations			Recognition Accuracy		
No. of Blocks	Skip-connections	Feature-Augmenter	5	20	50
5	✗	✗	93.66 ± 1.34%	84.44 ± 2.04%	78.90 ± 1.06%
5	✓	✗	94.56 ± 2.11%	86.38 ± 1.00%	81.60 ± 1.54%
10	✗	✗	92.91 ± 2.15%	85.32 ± 1.88%	75.03 ± 1.72%
10	✓	✗	94.60 ± 1.45%	86.55 ± 0.96%	77.44 ± 1.29%
5	✗	✓	93.76 ± 2.10%	87.11 ± 1.56%	82.87 ± 2.24%
5	✓	✓	92.73 ± 1.11%	83.14 ± 1.98%	79.55 ± 1.62%
10	✗	✓	95.71 ± 1.41%	89.13 ± 1.79%	87.21 ± 2.11%
10	✓	✓	**96.21 ± 1.37%**	**90.31 ± 1.82%**	**88.72 ± 2.09%**

TABLE III
ABLATION ANALYSIS AND COMPARATIVE RESULTS FOR DIFFERENT MODELS ON THE *Quick, Draw!* [23] DATASET IN TERMS OF ACCURACY REPORTED ON THREE SUBSETS WITH 5, 20 AND 50 CLASSES RESPECTIVELY. VALUES IN **BOLD** DEPICT THE HIGHEST ACCURACY ACHIEVED IN EACH SUBSET.

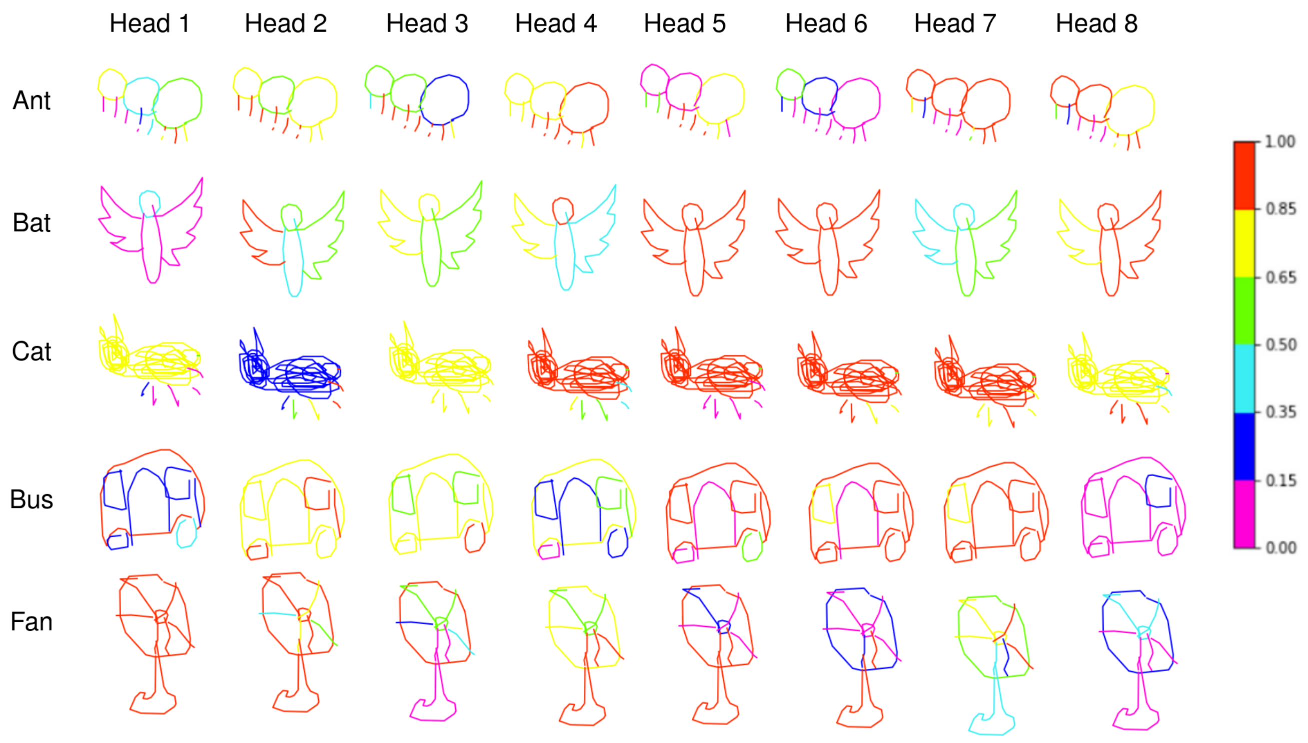

Fig. 4. Visualization of the attention heat-maps of a few classes in the *Quick, Draw!* dataset depicting the relative importance of the parts of sketch during inference.

confirms the effectiveness of representing sketches as vector images.

VI. DISCUSSIONS

A. Ablation Analysis

The proposed Transformer-module majorly constitutes two modules, the feature-augmenter, and the transformer blocks. In this section, we evaluate the effectiveness of these propositions. For this, we compare the recognition accuracy of different models by tweaking three model configurations. First, we vary the number of transformer blocks. Second, we add skip-connections between each transformer block to exploit the power of residual learning. Finally, experiments that corroborate the effectiveness of the proposed auto-encoder is performed. Table III reports the observed performance.

Varying the number of transformer blocks alters the number of parameters. The proposed network with 10 blocks performs significantly better than the 5-block variant. This enables the model to address the complexity of sketch recognition. Increasing the number of blocks further provided an insignificant increase in accuracy with a relatively large training overhead, such as time and space complexity.

Further, inclusion of residual learning observes improvement in model performances for CNNs [35]. In our network, we add skip-connections between the transformer blocks. This is because gradients often get lost when passed through multiple layers. In order to address this issue, skip-connections ensure that lower layers receive information, even if the network gets deeper.

The proposed architecture with feature-augmenter displays significantly superior recognition accuracy, as compared to other ablations. This is due to the fact that the offsets $(\Delta x_i, \Delta y_i)$, contain fairly low amount of information for computationally intensive architectures like transformers. Hence,

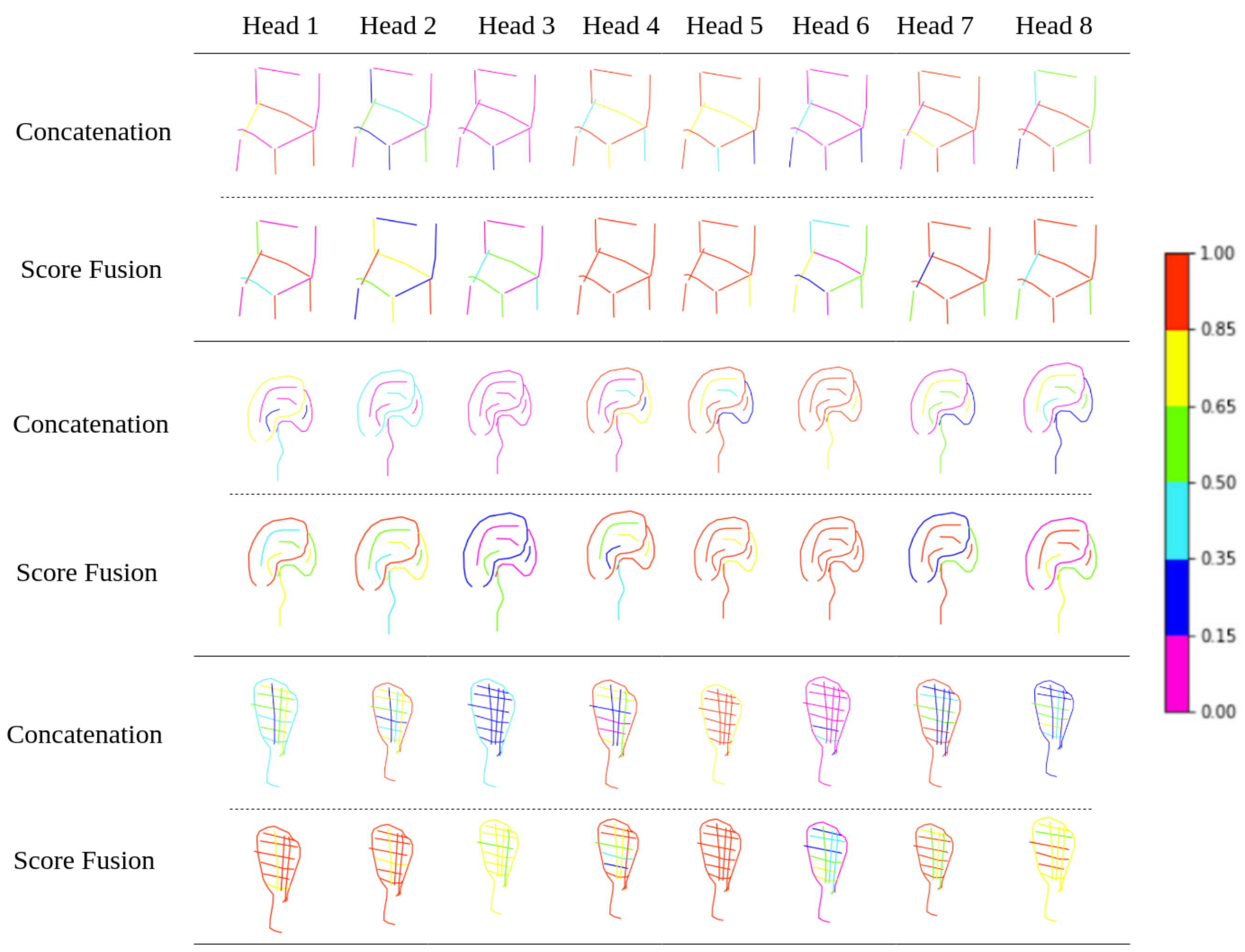

Fig. 5. Comparison of attention heat-maps obtained after (1) using score fusion, (2) concatenating the outputs of the Transformer and TCN modules.

mapping the offsets to a larger latent vector enables the transformer to tweak the attention heads with a lot more available input data.

B. Attention Visualization

Fig. 4 plots the attention heat-maps for all the 8 heads from the final block of the Transformer. Different heads focus on varying set of features in each sketch. For instance, in the class `fan`, most of the heads pay attention on the circular rim of a fan while the others focus on the spokes. For the `bat` class, most of the structures contain the same amount of attention weights like the central body, wings and face. Similar observations can be made for the other classes.

C. Order of Strokes

The usage of vector image raises another important question about the importance of *order of strokes*, which was previously immaterial in the case of raster images. Since the input is a real-time set of strokes, unlike images that are fed into the network after completion, it is important to assess the effect of the same for classification results. Fig. 6 displays two different ways to render the sketch of a `ant`. From the attention heat-map, it can be inferred that there is no effect of sequence of strokes for the final classification or attention. This is attributed to the fact that the *Quick, Draw!* dataset is acquired from a large number of people, maintaining the randomness in the order of strokes used to draw a sketch. This bias is completely removed from the training dataset itself.

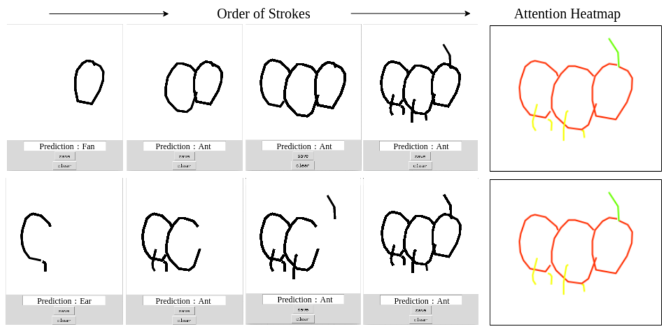

Fig. 6. Experiments to evaluate the effect of order of strokes on attention heat-maps and classification (drawn through a User Interface in real-time).

D. Score Fusion v/s Concatenation

For combining the outputs of the Transformer network and the TCN, an obvious candidate appears to be the concatenation of the two outputs. In this case, an end-to-end network is trained. The attention heat-maps in the two cases are plotted in Fig. 5. The attention heat-maps in the case of concatenation reveals a fairly inconsistent view of attention across different heads and classes, when compared with the score fusion case. This behaviour is mainly attributed to the fact that when both the networks are trained in an end-to-end fashion, the two tend to learn dissimilar feature sets. Hence, an effective way of combining the two approaches is to train them independently and fuse them at score level, rather than feature level. This is further corroborated with the recognition accuracy reported in Table II for the two cases.

E. Computational Complexity Analysis

In this subsection, we discuss the computational cost of the proposed approach in comparison to existing techniques. Traditional and CNN-based approaches can classify the sketch only once the complete sketch has been rendered. On the contrary, sequence-based models possess the ability to classify partial sketches, which facilitates real-time inference. Therefore, the use of sequence-based sketch classifiers over the traditional and CNN based classifiers yields a significant improvement in terms of the inference times. This extended functionality of sequence-based classifiers establishes dominance over these traditional and CNN-based approaches in terms of computational efficiency.

Amongst the sequence-based classifiers, the Transformer-based architectures significantly outperform the RNN-based classifiers in terms of computational complexity. This is primarily due to capability of Transformers to utilize parallel processing. In order to process an input with n time-steps, the Transformers and TCN both require $\mathrm{O}(1)$ sequential operations, while RNN-based architectures require a much greater $\mathrm{O}(n)$ sequential operations [33]. Further, we compare the average inference times per sample averaged across $10,000$ samples for the various architectures, reported in Table IV. A significant improvement in inference times can clearly be observed for the proposed Transformer-based architecture as compared to existing RNN-based implementations, such as RNN-Tensorflow[1].

Model	**Inference Time**
RNN-Tensorflow[1]	112.29 ± 10 ms
Transformer	80.38 ± 6 ms
Temporal Convolution Network	93.71 ± 3 ms
Ensemble (TCN + Transformer)	95.22 ± 7 ms

TABLE IV
AVERAGE PER-SAMPLE INFERENCE TIMES FOR THE VARIOUS ARCHITECTURES FOR THE SEQUENCE-BASED MODELS.

VII. Conclusion

The proposed approach provides an ensemble approach for combining the attention mechanism in Transformers, and the ability of TCN to extract long-term temporal dependencies information from the vector images for recognizing human drawn sketches. An extensive quantitative and qualitative comparison of the same supports the validity of the proposed method. Further, ablation analysis, along with attention maps reveal that the proposed approach makes prediction based on certain parts of sketches, that are intuitive in nature.

In the future, transformer-based networks can be adapted to solve challenging problems in the domain of computer vision. Potential applications include tasks such as sketch retrieval or sketch synthesis, which require descriptive line-drawing features, could be interesting to explore. The proposed approach opens avenues to enhance the cognition capabilities of machines to process sequential visual information. Alternate domains of computer vision that employ such sequential data include human action recognition for complex motion sequences like, dance, martial arts etc.

References

[1] J. J. LaViola Jr and R. C. Zeleznik, "Mathpad 2: a system for the creation and exploration of mathematical sketches," *ACM Transactions on Graphics (TOG)*, vol. 23, no. 3, pp. 432–440, 2004.

[2] T. Y. Ouyang and R. Davis, "Chemink: a natural real-time recognition system for chemical drawings," in *Proceedings of the 16th international conference on Intelligent user interfaces*. ACM, 2011, pp. 267–276.

[3] F. Keyrouz, L. Tauk, and E. Feghali, "Chemical structure recognition and prediction: A machine learning technique," in *2018 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)*. IEEE, 2018, pp. 1–7.

[4] ——, "Enhanced chemical structure recognition and prediction using bayesian fusion," in *2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC)*. IEEE, 2019, pp. 0611–0616.

[5] T. Lu, C.-L. Tai, F. Su, and S. Cai, "A new recognition model for electronic architectural drawings," *Computer-Aided Design*, vol. 37, no. 10, pp. 1053–1069, 2005.

[6] Z. Sun, C. Wang, L. Zhang, and L. Zhang, "Free hand-drawn sketch segmentation," in *European Conference on Computer Vision*. Springer, 2012, pp. 626–639.

[7] Z. Huang, H. Fu, and R. W. Lau, "Data-driven segmentation and labeling of freehand sketches," *ACM Transactions on Graphics (TOG)*, vol. 33, no. 6, p. 175, 2014.

[8] K. Li, K. Pang, J. Song, Y.-Z. Song, T. Xiang, T. M. Hospedales, and H. Zhang, "Universal sketch perceptual grouping," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 582–597.

[9] L. Li, H. Fu, and C.-L. Tai, "Fast sketch segmentation and labeling with deep learning," *IEEE computer graphics and applications*, vol. 39, no. 2, pp. 38–51, 2018.

[10] P. Xu, Y. Huang, T. Yuan, K. Pang, Y.-Z. Song, T. Xiang, T. M. Hospedales, Z. Ma, and J. Guo, "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.

[11] F. Wang, L. Kang, and Y. Li, "Sketch-based 3d shape retrieval using convolutional neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1875–1883.

[12] M. Eitz, R. Richter, T. Boubekeur, K. Hildebrand, and M. Alexa, "Sketch-based shape retrieval," *ACM Transactions on Graphics (TOG)*, vol. 31, no. 4, p. 31, 2012.

[13] M. Eitz, J. Hays, and M. Alexa, "How do humans sketch objects?" *ACM Trans. Graph.*, vol. 31, no. 4, pp. 44–1, 2012.

[14] Q. Yu, Y. Yang, F. Liu, Y.-Z. Song, T. Xiang, and T. M. Hospedales, "Sketch-a-net: A deep neural network that beats humans," *International journal of computer vision*, vol. 122, no. 3, pp. 411–425, 2017.

[15] J. Zhang, Y. Chen, L. Li, H. Fu, and C.-L. Tai, "Context-based sketch classification," in *Proceedings of the Joint Symposium on Computational Aesthetics and Sketch-Based Interfaces and Modeling and Non-Photorealistic Animation and Rendering*. ACM, 2018, p. 3.

[16] G. Saponaro, L. Jamone, A. Bernardino, and G. Salvi, "Beyond the self: Using grounded affordances to interpret and describe others' actions," *IEEE Transactions on Cognitive and Developmental Systems*, 2019.

[17] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[18] R. G. Schneider and T. Tuytelaars, “Sketch classification and classification-driven analysis using fisher vectors,” *ACM Transactions on Graphics (TOG)*, vol. 33, no. 6, p. 174, 2014.

[19] E. Yanık and T. M. Sezgin, “Active learning for sketch recognition,” *Computers & Graphics*, vol. 52, pp. 93–105, 2015.

[20] R. K. Sarvadevabhatla, S. Surya, T. Mittal, and R. V. Babu, “Game of sketches: Deep recurrent models of pictionary-style word guessing,” in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[21] I. E. Sutherland, “Sketchpad a man-machine graphical communication system,” *Simulation*, vol. 2, no. 5, pp. R–3, 1964.

[22] P. Sangkloy, N. Burnell, C. Ham, and J. Hays, “The sketchy database: learning to retrieve badly drawn bunnies,” *ACM Transactions on Graphics (TOG)*, vol. 35, no. 4, p. 119, 2016.

[23] D. Ha and D. Eck, “A neural representation of sketch drawings,” *CoRR*, vol. abs/1704.03477, 2017. [Online]. Available: http://arxiv.org/abs/1704.03477

[24] 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.

[25] R. K. Sarvadevabhatla and R. V. Babu, “Freehand sketch recognition using deep features,” *arXiv preprint arXiv:1502.00254*, 2015.

[26] O. Seddati, S. Dupont, and S. Mahmoudi, “Deepsketch 3,” *Multimedia Tools and Applications*, vol. 76, no. 21, pp. 22 333–22 359, 2017.

[27] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[28] M. Sert and E. Boyacı, “Sketch recognition using transfer learning,” *Multimedia Tools and Applications*, vol. 78, no. 12, pp. 17 095–17 112, 2019.

[29] X. Zhang, Y. Huang, Q. Zou, Y. Pei, R. Zhang, and S. Wang, “A hybrid convolutional neural network for sketch recognition,” *Pattern Recognition Letters*, 2019.

[30] J. Zheng, C. Lu, C. Hao, D. Chen, and D. Guo, “Improving the generalization ability of deep neural networks for cross-domain visual recognition,” *IEEE Transactions on Cognitive and Developmental Systems*, 2020.

[31] Y. Wang, X. Nie, Y. Shi, X. Zhou, and Y. Yin, “Attention-based video hashing for large-scale video retrieval,” *IEEE Transactions on Cognitive and Developmental Systems*, 2019.

[32] S. Yan, J. S. Smith, W. Lu, and B. Zhang, “Multibranch attention networks for action recognition in still images,” *IEEE Transactions on Cognitive and Developmental Systems*, vol. 10, no. 4, pp. 1116–1125, 2017.

[33] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, “Attention is all you need,” in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[34] S. Bai, J. Z. Kolter, and V. Koltun, “An empirical evaluation of generic convolutional and recurrent networks for sequence modeling,” *arXiv preprint arXiv:1803.01271*, 2018.

[35] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[36] A. Bensalah, P. Riba, A. Fornés, and J. Lladós, “Shoot less and sketch more: An efficient sketch classification via joining graph neural networks and few-shot learning,” in *2019 International Conference on Document Analysis and Recognition Workshops (ICDARW)*, vol. 1. IEEE, 2019, pp. 80–85.

[37] M. Hanmandlu, J. Grover, A. Gureja, and H. Gupta, “Score level fusion of multimodal biometrics using triangular norms,” *Pattern Recognition Letters*, vol. 32, pp. 1843–1850, 10 2011.

[38] M. Dehghani, S. Gouws, O. Vinyals, J. Uszkoreit, and Ł. Kaiser, “Universal transformers,” *arXiv preprint arXiv:1807.03819*, 2018.

Gaurav Jain (Student Member, IEEE) received B.Tech. degree in computer science from Delhi Technological University, New Delhi, India in 2020.

He is currently pursuing a Ph.D. in computer science at Columbia University, New York, NY, USA. His research interests include human-computer interaction, computer vision, medical imaging, and deep learning applications.

Shivang Chopra received his B.Tech. degree in computer science from Delhi Technological University, New Delhi, India in 2020.

His current research interests include natural language processing, computer vision, and deep learning applications.

Suransh Chopra received his B.Tech. degree in computer science from Delhi Technological University, New Delhi, India in 2020.

His current research interests include natural language processing, computer vision, medical imaging, and deep learning applications.

Anil Singh Parihar received his B. Tech. and Master in Engineering degree in Electronics and Communication Engineering. He is PhD in the area of Computer Vision & Evolutionary Computing. He joined the Department of Information Technology at Delhi Technological University,Delhi, India as Assistant Professor in 2010. He is currently an Associate Professor in the Department of Computer Science & Engineering at Delhi Technological University, Delhi, India. His current research interest includes Computer Vision, Machine Learning, Deep learning, Pattern Recognition, Soft computing and evolutionary algorithms.