Entrop y information-based heterogeneous deep selective fused features using deep convolutional neural network for sketch recognition

Shaukat Hayat\(^1\) | She Kun\(^1\) | Sara Shahzad\(^2\) | Parinya Suwansrikham\(^1\) | Muhammad Mateen\(^3\) | Yao Yu\(^1\)

\(^1\)School of Information and Software Engineering, University of Electronic Science and Technology of China, Chengdu, China
\(^2\)Department of Computer Science, University of Peshawar, Pakistan
\(^3\)Department of Computer Science, Air University Multan Campus, Pakistan

Correspondence
She Kun, School of Information and Software Engineering, University of Electronic Science and Technology of China, Chengdu, China. Email: kun@uestc.edu.cn

Funding information
The Science and Technology Project of Sichuan under Grant, 2020YFG0300

Abstract
An effective feature representation can boost recognition tasks in the sketch domain. Due to an abstract and diverse structure of the sketch relatively with a natural image, it is complex to generate a discriminative features representation for sketch recognition. Accordingly, this article presents a novel scheme for sketch recognition. It generates a discriminative features representation as a result of integrating asymmetry essential information from deep features. This information is kept as an original feature-vector space for making a final decision. Specifically, five different well-known pre-trained deep convolutional neural networks (DCNNs), namely, AlexNet, VGGNet-19, Inception V3, Xception, and InceptionResNetV2 are fine-tuned and utilised for feature extraction. First, the high-level deep layers of the networks were used to get multi-features hierarchy from sketch images. Second, an entropy-based neighbourhood component analysis was employed to optimise the fusion of features in order of rank from multiple different layers of various deep networks. Finally, the ranked features vector space was fed into the support vector machine (SVM) classifier for sketch classification outcomes. The performance of the proposed scheme is evaluated on two different sketch datasets such as TU-Berlin and Sketchy for classification and retrieval tasks. Experimental outcomes demonstrate that the proposed scheme brings substantial improvement over human recognition accuracy and other state-of-the-art algorithms.

1 | INTRODUCTION

Sketches are extensively used in daily life and have been utilised as a powerful communicating tool for a long time to express views, thoughts, and ideas. Due to the widespread use of touch-screen portable devices, a sketch becomes more popular and gets attention among the computer vision community to recognise it more efficiently. The use of these devices enables the users to properly search the required sketch object and retrieve similar objects from the desired database. The sketch is used very commonly in various daily living activities such as art and design industry, entertainment, drawings and making cartoons in education, criminal sketch recognising system, and so on.

In the recent past, research on sketches has been flourished, and many researchers explored its various characteristics in different applications, such as sketch recognition [1,2], sketch-based image retrieval [3], and sketch-based three-dimensional shape retrieval [4,5], etc. However, sketch recognition drawn by a non-artist is not only difficult for a computer, but even it is hard to recognise through human eyes. Although hand-drawn sketches represent scenes and objects in very simple coarse information, it is not essential to be artistic. Sketches are not like natural images. They are composed of lines and strokes without filled any colour or details. The contour details of sketches are considered highly informative and serve effectively for recognition purposes by humans. Thus, sketch recognition is still a challenging task to get robust
representation for recognising and retrieving sketches through a computer.

The main idea behind the sketch recognition is to correctly label a particular sketch in a class among the predefined sketch-object categories. In order to recognise sketch recognition, efficient feature extraction methods and retrieval techniques are needed to understand the semantics of input sketch images. Previously, sketch recognition follows traditional methods which have been generally used to extract handcrafted features for natural image classification. These methods are SIFT [6], HOG [7], GIST [8], Fisher vector [9], and often tied with Bag of Visual Words (BoW). However, in some cases, it corresponds to a final feature representations and then feeds them into the classifier for a final decision. In this vein, utilisation of deep learning approaches to extract deep features from input images [10,11], and/or deep architecture [12], particularly designed for sketch recognition also achieved state-of-the-art performance.

Recently, a deep neural network (DNN) provides a promising performance in the field of computer vision, including image recognition [13–15]. Deep learning techniques are employed for sketch recognition [2,16], and analysing a large scale sketch dataset [3]. Deep learning has the ability to learn more discriminative features from sketch input images and performs better in sketch recognition and retrieval than traditional approaches. For sketch recognition, a first deep neural network which has been particularly designed for sketch by [16], and the extracted deep features were implemented with improved performance in terms of sketch recognition. Similarly, at the beginning of deep features generation in sketch recognition domain, two most popular deep convolutional neural network (DCNN) architectures such as AlexNet [14] and LeNet [17] were utilised by [1], and very little improvement has been made. Recently, sketch classification and retrieval have been carried out utilising deep features from different layers based on various CNN architectures [10,18]. Visual recognition and retrieval task mostly depends on extracted features using deep models. Neural network architectures consist of different levels of layers, and each level has unique learning representations about input samples. The classification performance depends on the depth of model architecture, utilisation of various levels of layers, whether it may single layer or may a combination of multiple layers and number of training samples in the domain area. Generally, deep features are broadly divided into three basic levels, that is, high, middle, and low-level. Each level of feature has its individual strength and potential. The most recent studies [10,19], utilised a combination of multiple layers to get more robust feature representation as compared to single-layer performance.

The effectiveness of fused features from various architectures and features selection has a significant role in enhancing classification performance. Nevertheless, due to feature selection techniques, it helps to identify the most discriminative features from the input samples generated through multiple layers of various deep CNN architectures. An understanding of distinctive and powerful feature is the key issue in sketch recognition. So far, most studies consider only a fusion approach, while some consider the limited number of model layers. Although, in order to get better recognition or classification performance, to reduce computational efforts and eliminate redundancy for system efficiency, both feature selection, and feature fusion techniques should go together to obtain more robust features. By doing so, high-level deep layers of pre-trained DCNN architectures would be exploited to capture various levels of sketch semantics. We believe that sketch classification performance can be enhanced by considering the feature selection parallel with the feature fusion approach. This makes the requirement to get efficient selective features from the deeper layers of pre-trained DCNN architectures. Based on this, we propose a novel pipeline of Shannon entropy plus neighbourhood component analysis (NCA) for selecting the most appropriate and discriminative features for sketch recognition in this manuscript. The success of the proposed pipeline depends on the ability of selective fused discriminative features for each category of sketch images. The proposed approach makes our idea different from existing methods regarding sketch recognition and provides an edge in terms of accuracy from the state-of-the-art methods. Finally, the following can be the major contributions of this manuscript.

- Propose a novel and efficient DCNN-based scheme to undertake the problem of sketch recognition, which exploits the strength of deep selective fused features with low-dimensions.
- Investigate the impact of different deep layers for various DCNN architectures, that is, AlexNet, VGGNet-19, Inception-v3, Xception, and InceptionResNetV2 on sketch classification performance.

The rest of this manuscript is organised as follows: existing literature is reviewed in Section 2. In Section 3, the authors threw light on pre-trained DCNN architectures utilised in this study. Section 4 provides the details of the proposed scheme, and sketch datasets used in this work are explained in Section 5. Section 6 presents an experimental setup. Experimental results, discussion, and evaluation are provided in Section 7. Section 8 describes retrieval performance. Section 9 explains the limitations of the study and finally, the manuscript is concluded in Section 10.
2 | RELATED WORK

In the recent decade, computer vision researchers proposed a large number of research methods for sketch classification. The previous methods to sketch recognition include the handcrafted features approach and deep learning methods. These methods usually consist of two basic steps, feature extraction from input samples and classification using a classifier. We provide and briefly discuss sketch related reviews in this section.

2.1 | Handcrafted features approach

Previously sketch-related classification mainly focussed on global and local handcrafted features representation [3,20]. The most similar work to hand-drawn sketch, that is, CAD and artistic drawings as input samples were first handled by [21]. After the formation of the first large-scale hand-drawn sketch dataset TU-Berlin by [3], sketch classification becomes a hot topic among the computer vision community.

Eitz et al. [3] analyse input sketch images through handcrafted feature representation and use a SVM as a classifier. They achieve 56% classification accuracy on the TU-Berlin sketch dataset. Human recognition performance was also measured as 73.1% on this dataset. Additionally, considering TU-Berlin, they also define similarities and variations for inter-class and intra-class. Schneider and Tuytelaars [9] adapted the SIFT and GMM-based Fisher-vector encoding with SVM for sketch recognition and improved the results on a sketch benchmark dataset. An ensemble matching method was proposed by Li et al. [22] for sketch recognition. This method covers and matches them both local and global structures for sketch input samples. The matched structure was encapsulated in a bag-of-features as a single framework. A symmetric-aware flip invariant sketch histogram (SYM-FISH) descriptor has been proposed for sketch classification and retrieval tasks. The FISH descriptor of the input image was formed, and then, the kurtosis coefficient was calculated through the symmetry character of the input image. Finally, the symmetry table was generated. This descriptor encodes the symmetry information of the original shape, which represents natural objects and scenes. Eitz et al. [20] threw light and implemented local feature vector descriptors such as SIFT, HOG, SHOG, shape context, and spark features for sketch classification and sketch-based image retrieval system. A multi-kernel feature approach was demonstrated by Li et al. [24]. In this approach, various local features were combined for sketch recognition. Moreover, the performance of every individual feature was also analysed, and HOG performed better, among others.

In the recent past, a DCNN has revolutionised several research areas, including sketch recognition. Handcrafted features are replaced with deep features, and it shows the state-of-the-art performance. Research studies [1,12,25,26] employed DCNN to learn more discriminative features of sketch input samples with improved outcomes.

2.2 | Deep features approach

Recently, due to the state-of-the-art performance of DCNN in the field of computer vision, many researchers showed efforts for sketch classification and retrieval using a deep learning approach. Yang et al. [16] proposed the first CNN architecture for sketch recognition. This algorithm achieved superior accuracy (72.2%) on the TU-Berlin sketch dataset surpassing handcrafted features performance (68.9%). Later on, another research [1] extracts features from sketch input samples by implementing two CNN architectures, namely SketchNet [14] and modified LeNet [17] with SVM classifier. A specifically designed CNN architecture, namely Sketch-a-Net, has been proposed by Yu et al. [2] for sketch recognition. It achieved a 74.9% sketch classification accuracy and beat human sketch recognition performance. The same architecture was then modified in Reference [12], and the classification performance reached 77.95% on the TU-Berlin benchmark.

However, Seddati et al. [27] proposed residual block-based CNN model and improve the performance of query sketch and sketch retrieval accuracy. A five layers CNN model [28] is proposed and trained on augmented images (mixing natural images with sketch images). Moreover, various image transformations were generated to get more discriminative ability of the training dataset. The final outcomes were presented with the label. Jamil et al. [11] demonstrated a fine-tune deep CNN architecture to recognise and retrieved partially colour sketch images. Moreover, in Reference [26], the authors develop SketchNet to learn the common structure between natural images and sketch images automatically. This model is composed of three subnet models: R-net was utilised for extracting features from the conventional images. Second, S-net was used for sketch image analysis, while the last subnet model was exercised to identify the common structure between sketches and real images. The final result shows 80.42% classification accuracy on the TU-Berlin sketch dataset. The authors demonstrate the feature fusion approach [19] for sketch recognition based on Smartphones considering various layers of the CNNs model for feature extraction. More recently, for sketch classification and retrieval, Zhang et al. [29] propose a hybrid CNN model. This model is composed of a-Net (modified AlexNet) and S-Net to indicate appearance and shape feature representations. This model was evaluated on multiple sketch datasets and achieved a competitive performance. Another study [10] develops a DCNN framework using a transfer learning approach for sketch classification. This framework uses augmented variants with sketch images and extracts feature maps to construct a feature vector based on the global average pooling (GAP) layer. The proposed framework was evaluated on TU-Berlin augmented variants sketch dataset and achieved notable performance. Sert et al. [18] analyses different CNN architectures adapting the transfer learning approach for sketch recognition. In this framework, an early fusion of different layers using PCA for feature reduction and RBF-SVM classifier was observed. The performance of the defined framework was evaluated on two different publically available sketch datasets and achieved...
97.91% and 72.5% classification accuracy on Sketchy and TU-Berlin benchmark, respectively.

3 | PRE-TRAINED DEEP CONVOLUTIONAL NEURAL NETWORK ARCHITECTURES

3.1 | AlexNet

AlexNet is the first successful CNN architecture proposed by Krizhevsky et al. [14] and trained on 1.2 million images of 1000 object classes, a large-scale ImageNet dataset [30]. This architecture consists of five convolutional layers of $11 \times 11$, $5 \times 5$, $3 \times 3$ filter sizes, followed by three sub-sampling layers appended with three fully connected layers, and final layer is the softmax classifier (see Figure 1). It comprises of approximately 60 million trainable parameters. AlexNet is the winner of ILSVRC in 2012 by reducing the top five errors from 26% to 15.3%.

3.2 | Visual geometric group net

The visual geometric group (VGG) CNN architecture was trained by Zisserman et al. [13]. This network is the runner-up in localisation and classification tasks in the ILSVRC-2014 competition. It has a smaller convolutional filter of size $3 \times 3$ compared to AlexNet [14], five max-pooling layers and $2 \times 2$ three fully connected layers and finally having a linear classifier. This network increases the depth of CNN architecture to learn more complex features. In this study, we utilised the layers of VGGNet-19 for features extraction as illustrated in the Figure 2.

3.3 | Inception-v3

Inception-v3 [31] is a deep convolutional neural network and the advanced version of GoogleNet [32]. This module is the winner of the ILSVRC-2014 competition for classification and detection tracks. This network has 44 layers and 21 million learnable parameters. It contains many auxiliary classifiers within the middle layers to enhance the discriminative ability in the lower layers of the model. This architecture used varying sizes of filters at the same layers and can extract different sizes of patterns thus providing in-depth information. Although, this model can learn deeper features representation as AlexNet and with fewer parameters compared to VGGNet architecture. The Figure 3 illustrates the schematic structure of inception-v3, which is adopted in the current study.

3.4 | Xception

The Xception network is identical, and an extreme version of inception-v3 and proposed by Chollet [33]. It contains about 20 million learnable parameters, which is almost equal to the inception module. This architecture has 36 convolutional layers with linear residual connections. According to Reference [33], the inception module has been replaced with depth-wise separable convolutional layers. In this architecture, the inception module is considered as an intermediate between the lower (regular convolutional operations) and higher extreme (depth-wise separable convolutional). As mentioned earlier, a setup contains two significant convolutional layers: a depth-wise separable convolutional layer where unique convolution takes place separately in each input data channel. The second convolution layer is a point-wise convolutional layer, where $1 \times 1$ convolutional layer links the output streams through depth-wise convolution to new stream space. Figure 4 illustrates the compressed view of the Xception model.

3.5 | InceptionResNetV2

This architecture is the combination of two successful deep CNN networks, that is, Inception [31,32] and ResNet [34]. The detailed schematic diagram of the network is shown in Figure 5. Although, in this network, batch normalisation only used over traditional layers instead of using over the summations. The addition of residual modules in the architecture is to extend the blocks of the inception and the depth of the network accordingly. The problem associated with very deep networks is to affect the training phase, and this problem can be addressed by adding residual connections [34]. The training problem and the network stability are handled by scaling down the residuals connections [35]. Table 1 shows the details of higher layers that will be used in the proposed scheme and their dimensions, that is, it includes top layers or fully connected layers and average pooling layers and its respective dimensions of deep convolutional neural networks.

4 | RESEARCH DESIGN

The proposed methodology for sketch classification is illustrated in Figure 6. It is mentioned earlier in Section 3, that different deep CNN models would be used to extract features from its different layers. To precede this concept, feature selection procedure and dimensionality reduction technique would be applied on extracted features to hold the most
discriminative features for the classification task. In the following, all the steps are described in detail.

4.1 | Network training

4.1.1 | Fine-tuning of pre-trained architectures

Fine tuning strategy for pre-trained CNN architectures is the optimal solution when training samples for any target dataset is limited. In this case, the previous layers, which consist of general features are kept frozen, while the fine adjustment is made in the top-layers of the pre-trained architectures. Freezing is the method that takes place in the training phase, where the weights for any individual layer or more than one layer are not updated. Fine-tuning strategy can utilise the adjusted parameters of the reused network model and make it reasonable for the target dataset.

Fine-tuning can be distinctly performed; the first way is to consider all layers of the pre-trained architecture, and while in the second way, only top-layers are considered. The most preferred way is the fine-tuning of top-layers [36]. The reason...
Table 1: All the utilised layers and dimension details of DCNN architectures.

| Network architecture | Utilised layers | Dimensions | Output |
|-----------------------|-----------------|------------|--------|
| AlexNet               | FC-6            | 4096       | 1 × 1 × 4096 |
| AlexNet               | FC-7            | 4096       | 1 × 1 × 4096 |
| AlexNet               | FC-8            | 1000       | 1 × 1 × 1000 |
| VGG-19                | FC-6            | 4096       | 1 × 1 × 4096 |
| VGG-19                | FC-7            | 4096       | 1 × 1 × 4096 |
| VGG-19                | FC-8            | 1000       | 1 × 1 × 1000 |
| Inception-v3          | AvgPool         | 2048       | 1 × 1 × 2048 |
| Inception-v3          | FC              | 1000       | 1 × 1 × 1000 |
| Xception              | AvgPool         | 2048       | 1 × 1 × 2048 |
| InceptionResNetV2     | AvgPool         | 1536       | 1 × 1 × 1536 |
| InceptionResNetV2     | FC              | 1000       | 1 × 1 × 1000 |

Abbreviation: DCNN, deep convolutional neural network.

behind is that top-layers encode more specific features than the earlier layers of the network. Nevertheless, the earlier layers of the networks generate very generic and reusable features that may not be effective for extracting more specific information from the input image. Fine-tuning of all layers of the network can cause overfitting problem due to an extensive number of parameters involvement [36].

In this study, fine-tuning of top-layers was carried out from the pre-existing deep CNN architectures. These top-layers be only fully connected layers (e.g., AlexNet and VGGNet) or some of the additional layers, that is, convolutional layers, average pool (e.g., Inception, Xception and InceptionResNetV2 networks). Therefore, fine-tuning of top-layers enabled us to match the effectiveness of both fine-tuned layers whether it may be FC and convolution layer or average pool layer. Notably, in this study, two large-scale publically available sketch image datasets are considered to train fine-tune deep CNN architectures.

4.2 Selection of feature layers

Fine-tune pre-trained deep convolutional neural networks can be employed for feature extraction from the input images. Since, various layers of CNN architecture provide different visual effects and thus offer distinct features. The transferred weights would be kept frozen on their initial values to extract off-the-shelf features from CNN layers during the training phase. The choice of the features layers has been utilised in different computer vision applications and considered feature maps and fully connected layers as feature extractors. In this study, various feature layers from different fine-tuned pre-trained DCNN architectures are selected. Top-layers, that is, FC6, FC7, and FC8 are considered from AlexNet and VGGNet architectures. The last average pool layer and fully connected layer are selected from Inception and InceptionResNetV2 architectures. Furthermore, the last average pool layer is adopted for feature extraction from the Xception network. The adopted layers and their notations are illustrated in Table 2. The rationale behind this concept is to consider the strength of multiple layers of various deep CNN architectures to improve the sketch classification accuracy.

CNNs are capable for learning various levels of features from the input data. Lower layers of the network capture relatively less informative features, while the higher layers encode semantically rich features. Each layer of the network consists of neuronal activation, which matches the intermediate representations of the input image. These representations of the image can be utilised for either classification or retrieval tasks. Nevertheless, higher layers of the network can

Figure 6: Diagram of the proposed scheme.
TABLE 2 Details of selected layers of different architectures and feature vector notations

| Index | Model           | Feature Layers | Notations Used for feature Vectors |
|-------|----------------|----------------|-----------------------------------|
| 1     | AlexNet         | FC6, FC7, FC8  | \(F_{AX1}, F_{AX2}, F_{AX3}\)    |
| 2     | VGG-19          | FC6, FC7, FC8  | \(F_{GV4}, F_{GV5}, F_{GV6}\)    |
| 3     | Inception-v3    | Avgpool, FC    | \(F_{PV7}, F_{PV8}\)             |
| 4     | Xception        | Avgpool        | \(F_{XV9}\)                      |
| 5     | InceptionResNetV2 | Avgpool, FC  | \(F_{R_{V10}}, F_{R_{V11}}\)      |

![Figure 7](image-url)  

**FIGURE 7** (a) and (b) Sketch features representation (TU-Berlin). (c) and (d) Sketch features representation (Sketchy)

learn more discriminative and domain-relevant features [37], so their performance is better than lower layer feature representations.

Therefore, in this study, the higher layers of the neural networks are adopted for features extraction. The adopted different deep layers from multiple architectures are more suitable and have enough discriminative features ability for sketch classification and retrieval. We have shown some sample sketch images and extracted features from the same images in Figure 7a–d. These sketch images belong to the same classes, but their appearances are visually different. However, the extracted features from such sketch images are similar, which
clearly indicate the discriminative ability of the proposed research pipeline for sketch recognition.

4.3 Feature fusion

Different layers from different deep architectures offer dynamic visual characteristics. Extracting features from multiple layers of different architectures can strengthen classification outcomes as compared to the utilisation of an individual layer. In this research, the idea behind the heterogeneous feature fusion of different deep CNN architectures is to analyse their impact on sketch recognition. In this study, a high-level feature from the average pooling layers and top-layers of various neural networks are considered and directly concatenated to construct a joint feature space. Let different layers of different deep CNN architecture be represented as follows:

Let

\[ A = a \in \{1, 2, 3, \ldots, 11\} \]
\[ B = b \in \{1, 2, 3, 4, 5\} \]

Equation (1), represents a number of selected layers, where the alphabet ‘a’ contains a range of values from 1 to 11 represents each layer exist in different DCNN architectures, for example, the architecture AlexNet having three layers FC6, FC7, FC8. Similarly, three layers of VGGNet that is, FC6, FC7, FC8 and so on. Therefore, we have total different 11 layers from different deep CNN architectures in this study. The details of selected layers of the deep CNN architectures are provided in Table 2. Similarly, Equation (2), showing a number of different CNN architectures, where the alphabet ‘b’ consists the values from 1 to 5, and each value represents different individual deep CNN architecture for example, in this study we utilised total five DCNN architectures. The details of DCNN architectures are mentioned in Table 2. Therefore, the fused feature vector \( f_{vB} \) having dimensions \( D \) can be expressed as:

\[ f_{vB} = f_{vA} \text{ or } f_{vB} = f_{vB} \]

The impact of the fused features on the classification results is always effective as compared to those features which are extracted from the single layer. However, to ensure a better outcome from fused features, a feature dimensionality reduction technique is necessary to reduce the multi-feature redundancy, to decrease the computational cost, and improve the classification performance. In this vein, a Shannon entropy as a feature selection method with NCA for reducing the curse-of dimensionality is employed.

4.4 Feature selection and reduction

In order to get improved sketch recognition, the impact of fused features approach has certain advantages as compared to a single one; by considering these features and to overcome the computational cost, a series of asymmetry features are exploited for selecting the most efficient features. Nevertheless, feature fusion can cause the curse of dimensionality problem. To select the most appropriate features and to reduce their dimensions, we propose entropy-based NCA. For the current research problem, Shannon entropy is utilised for feature selection, while NCA is used to reduce the feature dimensions. Entropy has been used to extract brain wave features [38] from various brain rhythms of electroencephalography for person identification. In this vein, feature selection and efficient sketch recognition, nonlinear analysis of Shannon entropy [39] is more appropriate to provide more effective information in terms of feature representation.

Accordingly, mathematically assume that \( \Phi = \{ (sl_1, sm_1), \ldots, (sl_i, sm_i), \ldots, (sl_N, sm_N) \} \) is training set having \( N \) labels, where \( SL = \{ sl_i \}_{i=1}^{N} \in \mathbb{R}^d \) is \( d \)-dimension feature vector and \( SM = \{ sm_i \}_{i=1}^{N} \) corresponds to class labels with \( sm_i \in \{1, \ldots, K\} \). We have this space, and it has \( \varphi \) measure with \( \varphi(SL) = 1 \); then Shannon entropy to be expressed as:

\[ E(SL) = -\sum_{i=1}^{N} (sl_i) \log \varphi(sl_i) \]

where \( \varphi(sl_i) \) shows the probability to specifically perceive \( sl_i \) of \( SL \). This helped to obtain most discriminative feature vectors, which are the best representative of sketch training samples. The high rank features are selected in terms of their contribution and discard the low rank features. All the high ranked features are organised in the global entropy. Entropy covers the first level reduction features by eliminating low-rank features and preserves the real information of high-ranked features for the next level of reduction.

NCA [40] is the supervised distance metric learning approach, which learns from both samples and label of samples. The aim of this approach is to learn the transformation matrix and to boost the performance of the nearest neighbour classification. The expected leave-one-out (LOO) error of stochastic nearest neighbour classifier in the projection space requires to optimise by the NCA through this transformation. The LOO error is an important statistical estimator of the performance of learning algorithm [41]. By doing this, NCA needs to find weighting vector \( w \) for feature selection to choose feature subset by itself for classification optimisation. This technique take up projection matrix \( Q \in \mathbb{R}^{p \times d} \) shows the transformation which projects the training vector \( sl_i \) into a \( p \)-dimensional representation \( \mathbf{r}_i = Q sl_i \), where \( \mathbf{r}_i \in \mathbb{R}^p \) and \( p \leq d \).

NCA utilises quadratic distance for the vector \( w \) and computes the Euclidean distance between the samples \( sl_i \) and \( sl_j \) by the following:

\[ d_w(sl_i, sl_j) = \sum_{k=1}^{d} \left( Q sl_{ik} - Q sl_{jk} \right)^2 \left( Q sl_{ik} - Q sl_{jk} \right) \]
\[ = \sum_{k=1}^{d} w_{ik}^2 (sl_{ik} - sl_{jk}) \]
where $w_k$ shows the weight linked with $k$ feature vectors. During the feature selection, NCA uncovers low dimensional feature representation, which maximises labelled sample separation and takes up differentiable cost function in the transformed space through the stochastic neighbour assignment as follows; with reference of some probability, each input sample $sl_i$ selects another input sample $sl_j$ as follows

$$P_{ij} = \frac{\lambda^- (d_w(sl_i, sl_j))}{\sum_{i \neq j} \lambda^- (d_w(sl_i, sl_j))}, \text{ if } i \neq j$$  \hspace{1cm} (6)

where $\lambda^- (\gamma) = \exp(-\gamma/\beta)$ is kernel function and $\beta$ is the width of kernel and the input parameter to be selected as reference, which influences the probability for each input sample. This whole operation makes the model more flexible. Due to the stochastic rule, the probability $P_i$ that the input sample $sl_i$ will be accurately classified is shown as:

$$P_i = \sum_{j \in Y_i} Y_{ij} P_{ij}$$  \hspace{1cm} (7)

where $Y_{ij} = 1$ if $Y_i = Y_j$, otherwise $Y_{ij} = 0$; NCA looks for the maximisation of expected number of error-free sample, and LOO can be presented as

$$\eta(Q) = \sum_i \sum_j Y_{ij} P_{ij} = \sum_i P_i$$ \hspace{1cm} (8)

Nevertheless, to carry out feature selection, and to overcome the overfitting, [42] initiate regularisation $\lambda$ into cost function. The objective function is written as;

$$\eta(Q) = \sum_i \sum_j Y_{ij} P_{ij} = \lambda \sum_{k=1}^l w_k^2$$ \hspace{1cm} (9)

Here $\lambda$ is the regularisation parameter and $\lambda > 0$, which can be tuned through cross-validation [42].

$$\frac{\partial \eta(Q)}{\partial w_k} = 2w_k \left( \frac{1}{\beta} \sum_{j} \left[ P_i \sum_{j \neq k} Y_{ij} P_{ij} \right] \left| sl_{ik} - sl_{jk} \right| - \sum_j Y_{ij} P_{ij} \right) \left| sl_{ik} - sl_{jk} \right| $$ \hspace{1cm} (10)

A gradient-based optimiser can be used to maximise the objective function, such as conjugate gradients shown in Equation (10).

### 5 | DATASETS

In this manuscript, we used two standard benchmark sketch datasets namely TU-Berlin and Sketch dataset for the evaluation of proposed research design.

#### 5.1 | TU-berlin

This dataset was first generated by Simonyan and Zisserman [3]. It consists totally of 20,000 hand-drawn sketches. These sketches are distributed over 250 object categories. Every category has 80 sketches. Figure 8 illustrates a sample sketch images from this dataset. Sketch dataset was collected on Amazon Mechanical Turk (AMT) by 1350 non-expert participants. Every sketch image is provided in 1111 x 1111 sizes.
To make it more manageable, the size of sketch images fixed to be 256 × 256 pixels in this experiment. Human recognition accuracy on this dataset is 73.1%. This dataset is available on the following website (http://cybertron.cg.tu-berlin.de/eitz/projects/classifysketch/).

5.2 | Sketchy

It is fine-grained large collection of sketch-photo pairs dataset [25]. This dataset consists 12,500 unique images of objects. It also contains 75,481 sketch images from 125 object categories. Figure 9 shows the sample image examples from sketchy dataset. This dataset covers the ImageNet object categories, which mostly contain all the TU-Berlin sketch instance categories. This dataset is publically available to make improvements in the sketch-photo recognition. This dataset is available on the following link (http://sketchy.eyetube.gatech.edu/).

6 | EXPERIMENTAL SETUP

This section describes the experimental details to carry out the proposed pipeline for sketch classification. Both sketch datasets are split into training and testing parts. The deep CNN architectures that are considered for this study are AlexNet, VGGNet, Inception-v3, Xception, and InceptionResNetV2. During the fine-tuning, all the parameters of the original networks were retained. Nevertheless, the learning rate values were set to be 0.01, the value for decay was selected as 10^{-4}, and 30,000 number of iterations was set for experiments. An optimisation algorithm, Stochastic Gradient Descent (SGD) and the loss function, cross entropy were selected for training the models. In this study, to carry out all these experiments, NVidia GTX 1070 Ti GPU using keras framework with CUDA version 9.0 was used.

7 | RESULTS AND DISCUSSION

All the classification results are performed on two different sketch benchmark datasets with all the selected training and testing measurements using three-fold cross validation to make comparisons with alternatives. Prior to implementing a proposed fusion pipeline, we investigate the performance of individually selected layers of different DCNN architectures. Later on, the classification performance of the proposed fusion pipeline is compared with baselines, along with the sketch-based image retrieval performance. Compared with the state-of-the-art results, this section concludes the effectiveness of the proposed sketch recognition pipeline.

7.1 | Individual layer performance

This subsection presented the individual layer performance about sketch classification using different fine-tuned, pre-trained CNN architectures. Two different sketch datasets are considered to evaluate networks’ individual layer performance.

Based on the TU-Berlin sketch dataset, the sketch classification outcomes by evaluating the different individual layers of various fine-tuned pre-trained deep CNN architectures are illustrated in Figure 10. Based on the results, it is evident that the average pool layer with the Xception model achieves the highest performance for sketch classification, that is, feature vector $F_{X,V9} = 69.21\%$. The average pool layers of Inception-v3 and InceptionResNetV2 also produce sketch classification performance of $F_{P,V7} = 69.09\%$ and $F_{R,V10} = 61.34\%$, respectively. Nevertheless, other layers of the networks achieve sketch classification accuracies of feature vectors $F_{A,V1} = 67.81\%$, $F_{P,V8} = 66.93\%$, $F_{G,V4} = 61.1\%$, $F_{R,V11} = 59.49\%$, $F_{G,V5} = 59.33\%$, $F_{A,V2} = 58.49\%$, $F_{G,V6} = 57.54\%$ and $F_{A,V3} = 55.28\%$. Our results also reveal that individual layers of Xception, that is $F_{X,V9}$ and
Inception-v3 architectures perform better than top-layers of VGNNet, AlexNet and InceptionResNetV2.

Considering the individual layer performance of different CNN architectures based on the Sketchy dataset is presented in Figure 11. The sketch classification accuracies from various layers of different CNN architectures showed the strength of features representation. It is observed from the individual layers' outcomes that the Xception network performs better than other layers of deep CNN models, including AlexNet, VGNNet, Inception v3, and InceptionResNetV2. It is also observed that some of the individual layers of both AlexNet and VGNNet architectures perform weaker for a Sketchy dataset and are not the optimal choice to consider it for a classification task. However, some of the layers' performance of these networks are almost similar to others, such as the feature vectors $F_{AV1} = 81.18\%$ and $F_{GV6} = 81.83\%$; similarly, $F_{AV3} = 72.79\%$ and $F_{AV2} = 73.57\%$, $F_{GV4} = 76.51\%$ and $F_{GV5} = 77.64\%$. Moreover, the feature vectors $F_{AV11} = 82.49\%$, $F_{AV10} = 86.34\%$ beat the AlexNet and VGNNet in sketch classification performance. Additionally, the individual layers of Inception-v3 and Xception architectures demonstrate almost alike performance for a Sketchy dataset, for example, the classification performance of these feature vectors are $F_{PV8} = 90.19\%$, $F_{PV7} = 92.09\%$ and $F_{PV9} = 93.47\%$. Similarly, the classification performance of InceptionResNetV2 is better than that of both AlexNet and VGNNet but shows the performance declination for the sketchy dataset in comparison with Inception-v3 and Xception architectures.

### 7.2 Fused features performance

After features extraction from each individual layer by the various fine-tuned deep CNN architectures, for most discriminative features, the extracted features are concatenated before employing features selection and dimensionality reduction techniques. By doing so, feature vectors’ reductions are obtained with an average rate of 94% before the sketch classification and are shown in Figure 12. In this vein, 14 feature vectors are constructed, having a combination of different layers of deep CNN architectures. The entropy-based NCA is employed to reduce the size of feature vectors. After that, the maximum 94.59% reduction of feature vector among others is obtained with kernel width $\beta = 1$ in the step of NCA. Several experiments are conducted with $\lambda = 0.003$ regularisation parameter, which makes available enough discriminative feature representation.

In order to evaluate the proposed scheme, sketch classification accuracies based on the TU-Berlin dataset are shown in Table 3. In the tabulated results, different feature vectors'
The performance of the proposed scheme against only NCA, which produces an accuracy of 96.04%, respectively. The sketch dataset with entropy-based NCA beats all the existing methods. It is also observed from the experimental outcomes that the fusion of average pooling layers of deep CNNs is efficient in sketch recognition as compared to FC layers.

Interestingly, the better performance for sketch classification on both TU-Berlin and Sketchy datasets is achieved using fusion vectors of average pooling layers from different deep CNN architectures. We improve sketch recognition with a rate of 72.93% and 97.96% by using Entropy-based NCA.

To get efficient information for discriminative features representation, we measured the average feature extraction time for every individual input sample using utilised feature vectors for TU-Berlin and Sketchy datasets. The measured average time for a single input sample is shown in Figure 13a,b. It is quite clear that utilised vectors efficiently extract information from sketchy data samples than TU-Berlin sketch samples, and the reason might depend on the strength of training samples.

### 8 Sketch-Based Retrieval Analysis

The proposed scheme extensively evaluated on the sketch retrieval Sketch-Based Image Recognition (SBIR) task. For this experiment, the proposed scheme is trained and validated on a Sketchy [25] dataset. All the natural images from Sketchy are considered as candidate images, while the sketch images are used as query samples. Features are extracted from both sketch images and natural images using the proposed scheme. The

### TABLE 3 Sketch classification accuracies of the proposed scheme are compared with simple NCA using TU-Berlin dataset

| Feature vectors | Reduction rate (%) | Accuracy with NCA | Accuracy with entropy-based NCA |
|-----------------|--------------------|-------------------|---------------------------------|
| F_{XV7}–F_{XV9} | 94.08              | 71.09%            | 72.11%                          |
| F_{XV9}–F_{XV9} | 94.13              | 71.41%            | 72.53%                          |
| F_{XV7}–F_{XV9}–F_{RV10} | 94.59 | 71.88%            | 72.93%                          |

Abbreviation: NCA, neighbourhood component analysis.

### TABLE 4 Sketch classification accuracy of the proposed scheme compared with simple NCA using Sketchy dataset

| Feature vectors | Reduction rate (%) | Accuracy with NCA | Accuracy with entropy-based NCA |
|-----------------|--------------------|-------------------|---------------------------------|
| F_{GV9}–F_{XV9} | 94.07              | 95.17%            | 96.24%                          |
| F_{PV7}–F_{XV9} | 94.47              | 95.74%            | 97.22%                          |
| F_{PV7}–F_{XV9}–F_{RV10} | 94.59 | 96.04%            | 97.96%                          |

Abbreviation: NCA, neighbourhood component analysis.

fusions are used, and sketch classification outcomes are recorded in two different ways, that is, accuracy with NCA and accuracy with a proposed scheme (entropy-based NCA).

The following observations are made based on the classification results of the TU-Berlin sketch dataset. The performance of the feature vectors fusion F_{PV7}–F_{XV9}–F_{RV10} achieves the highest classification with a rate of 72.93% to follow entropy-based NCA as compared to the fusion of F_{PV7}–F_{XV9} and F_{XV7}–F_{PV7} vectors with a rate of 72.53% and 72.11%, respectively. The dimensionality reduction rate of the fusion vector F_{PV7}–F_{XV9}–F_{RV10} is also bigger than other fusion vectors.

Tabulated results also show that average pooling layers' fusion vectors of deep CNNs architectures are best performing layers than fusion vectors accuracies produced by FC layers. Based on the results and considering the case with NCA, average pooling layers' fusion vector F_{PV7}–F_{XV9}–F_{RV10} also outperforms with the accuracy rate of 71.88% than F_{PV8}–F_{XV9} and F_{XV7}–F_{PV7} fusion vectors with the accuracy rate of 71.41% and 71.09%, respectively.

The performance of the proposed scheme on TU-Berlin dataset is also compared with state-of-the-art methods and sketch-based CNN architectures. The proposed method significantly outperformed the existing methods except for the model [2,12], which has been specifically designed for sketch recognition and stroke-based feature extraction using RNN-RNN dual branch [43]. The proposed scheme classification accuracy (72.93%) also beats the classification accuracy obtained by LeNet (55.2%) [17], HOG-SVM (accuracy rate 56%) [3], sketch classification accuracy (61.5%) by Ensemble matching [22], Multi-Kernel SVM (65.8%) [24], AlexNet-SVM (67.1%) [14], and Fisher vector SP (68.9%) [9], in the literature study. Furthermore, the achieved results are also compared with most recent deep CNNs studies with classification accuracies (72.82%) [10], (72.5%) [18], and (69.17%) [19]. The proposed scheme also achieves competitive results as compared to human accuracy (73.1%) [3] for sketch recognition.

On the other hand, the proposed scheme is evaluated on a sketchy dataset; the experimental outcomes are presented in Table 4. The fusion of F_{PV7}–F_{XV9}–F_{RV10} is the best performing vector with a sketch classification accuracy of 97.96% to follow entropy based NCA. However, a sketchy dataset using F_{GV9}–F_{XV9} and F_{PV7}–F_{XV9} achieves lower accuracy of 96.24% and 97.22%, respectively. The reduction rate of 94.59% of the fusion vector F_{PV7}–F_{XV9}–F_{RV10} is also higher than other fusion vectors. Similarly, the results also show that the proposed scheme outperforms against only NCA, which produces an accuracy of 96.04%, respectively.
extracted features are indexed with concerned images. The anisotropic diffusion approach [44] is used to extract edge-maps from natural images. After that, the proposed scheme is separately used to extract features from edge maps and also from the query sketch images. In the end, SVM classifier with Euclidean distance is used to retrieve the candidate images from the database against randomly selected query sketch images.

The sketch queries and top-11 candidate retrieval images on the sketchy dataset are illustrated in Figure 14. According to retrieval outcomes in the following figure, candidate images are mostly retrieved with high rank similarity (i.e. flower, tiger, mushroom, pineapple) and represent enough discriminative features for retrieval tasks. They had very similar edge-maps with query sketch, which enables the retrieval task in high rank. However, low-rank retrieval results on the sketchy dataset are illustrated in Figure 15. Additionally, the proposed scheme also evaluated and checked its performance with photos and sketch images [10] other than used in training and validation.

In the retrieval results shown in Figure 16, some of the candidate images in top-11 are not relevant to its query sketch image, while the correctly matched candidate images are retrieved with high rank similarity score. These outcomes further validate the effectiveness of the proposed scheme with a huge amount of images other than used during training.

Moreover, the high-rank and low-rank similarity score values of the retrieval performance are shown in Figures 17 and 18, respectively. The score value closer to 1 shows the greater similarity between the query and candidate images.

9  LIMITATIONS

The proposed scheme achieved a competitive recognition performance with the human recognition accuracy on TU-Berlin sketch dataset. This performance can be further
enhanced to generate textures and shape features from the sketch samples. The newly generated features to be further refined to obtain high-ranked features. By doing so, without being a little modification in the architecture of the proposed scheme, generation of texture- and shape-based ranked features are one of the associative limitations of the proposed scheme. We believe that deployment of these practices can overcome the limitations of our proposed scheme, and a better opportunity can be in hands to further improve the sketch recognition performance. This work will be our main focus as a future research direction.

**FIGURE 15** Low rank similarity retrieval performance on sketchy dataset. The sketch image in start of every row is query image and remaining are top-11 retrieved candidate images. Red-boxes show incorrect retrieved candidate objects.

**FIGURE 16** High-rank similarity retrieval performance other than training and validation sketch-photo mixed images. The sketch image in start of every row is query image and remaining are top-11 retrieved candidate images. Red-boxes show incorrect retrieved candidate objects.

**FIGURE 17** High-rank score for top-11 retrieved candidate images on Sketchy dataset.
10 | CONCLUSION

In this manuscript, a fine-tuned, pre-trained, neural network-based scheme was proposed for sketch recognition. This scheme obtains an efficient and discriminative feature representation for sketch recognition and retrieval task. To get enough discriminative features and utilise complementary information for sketch recognition, extracted features of different DCNN architecture were fused prior to implementing selection and dimensionality reduction techniques. The high ranked features were obtained through implementing Shannon entropy, followed by NCA’s assistance in reducing the curse of dimensionality of the fused features. Entropy-based NCA not only reduces features dimensionality but also improves the sketch classification. The recognition performances on the utilised datasets were significantly improved over some existing methods. The proposed scheme is more effective for sketch-photo paired input samples.

ORCID
Shaukat Hayat https://orcid.org/0000-0002-4033-5249
Sara Shahzad https://orcid.org/0000-0002-1812-236X

REFERENCES
1. Kiran Sarvadevabhatla, R., Venkatesh Babu, R.: Freehand sketch recognition using deep features. arXiv e-prints (2015)
2. Yu, Q., et al.: Sketch-a-Net that beats humans. arXiv e-prints (2015)
3. Eitz, M., Hays, J., Alexa, M.: How do humans sketch objects? ACM Trans. Graph. 31(4), 1–10 (2012)
4. Mao, D., Hao, Z.: A novel sketch-based three-dimensional shape retrieval method using multi-view convolutional neural network. Symmetry. 11(5), 1–13 (2019)
5. Wang, F., Kang, L., Li, Y.: Sketch-Based 3d shape retrieval using convolutional neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2015)
6. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vis. 60(2), 91–110 (2004)
7. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) (2005)
8. Oliva, A., Torralba, A.: Modelling the shape of the scene: a holistic representation of the spatial envelope. Int. J. Comput Vis. 42(3), 145–175 (2001)
9. Schneider, R.G., Tuytelaars, T.: Sketch classification and classification-driven analysis using Fisher vectors. ACM Trans. Graph. 33(6), 1–9 (2014)
10. Hayat, S., et al.: Deep CNN-based features for hand-drawn sketch recognition via transfer learning approach. Int. J. Adv. Comput. Sci. Appl. 10(9), 438–448 (2019)
11. Ahmad, J., Muhammad, K., Baik, S.W.: Data augmentation-assisted deep learning of hand-drawn partially coloured sketches for visual search. PLoS One. 12(8), 1–19 (2017)
12. Yu, Q., et al.: Sketch-a-Net: a deep neural network that beats humans. Int. J. Comput Vis. 122(3), 411–425 (2017)
13. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv e-prints (2014)
14. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. Adv. Neural Inform. Process. Syst. 25, 1097–1105 (2012)
15. Xu, P.: Deep learning for free-hand sketch: a survey. arXiv preprint arXiv:2001.02600 (2020)
16. Yang, Y., Hospedales, T.: Deep neural networks for sketch recognition. arXiv e-prints (2015)
17. Lecun, Y., et al.: Gradient-based learning applied to document recognition. Proc. IEEE. 86(11), 2278–2324 (1998)
18. Sert, M., Boyaci, E.: Sketch recognition using transfer learning. Multimed. Tool Appl. 78(12), 17095–17112 (2019)
19. Boyaci, E., Sert, M.: Feature-level fusion of deep convolutional neural networks for sketch recognition on smartphones. In: 2017 IEEE International conference on Consumer Electronics (ICCE) (2017)
20. Eitz, M., et al.: Sketch-based image retrieval: benchmark and bag-of-features descriptors. IEEE Trans. Visual Comput. Graph. 17(11), 1624–1636 (2011)
21. Rahim, M., Othman, N., Jupri, Z.: A comparative study on extraction and recognition method of CAD data from CAD drawings. In: IEEE International Conference on Information Management and Engineering (2009)
22. Li, Y., Song, Y.-Z., Gong, S.: Sketch recognition by ensemble matching of structured features. In: British Machine Vision Conference (BMVC) (2013)
23. Cao, X., et al.: Sym-Fish: a symmetry-aware flip invariant sketch histogram shape descriptor. In: 2013 IEEE International Conference on Computer Vision (2013)
24. Li, Y., et al.: Free-hand sketch recognition by multi-kernel feature learning. Comput. Vis. Image Understand. 137, 1–11 (2015)
25. Sangdloy, P., et al.: The sketchy database: learning to retrieve badly drawn bunnies. ACM Trans. Graph. 35(4), 1–12 (2016)
26. Zhang, H., et al.: Sketchnet: sketch classification with Web images. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
27. Sedditi, O., Dupont, S., Mahmoudi, S.: Deepsketch 3 analysing deep neural networks features for better sketch recognition and sketch-based image retrieval. Multimed. Tool Appl. 76(21), 22333–22359 (2017)
28. Wang, X., Duan, X., Bai, X.: Deep sketch feature for cross-domain image retrieval. Neurocomputing. 207, 387–397 (2016)
29. Zhang, X., et al.: A hybrid convolutional neural network for sketch recognition. Pattern Recogn. Lett. 130, 73–82 (2019)
30. Deng, J., et al.: Imagenet: a large-scale Hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition (2009)
31. Szegedy, C., et al.: Rethinking the inception architecture for computer vision. In: IEEE Conference on Computer Vision and Pattern Recognition (2016)

32. Szegedy, C., et al.: Going deeper with convolutions. arXiv e-prints (2014)

33. Chollet, F.: Xception: deep learning with depthwise separable convolutions. arXiv e-prints (2016)

34. He, K., et al.: Deep residual learning for image recognition. arXiv e-prints (2015)

35. Szegedy, C., et al.: Inception-V4, inception-Resnet and the impact of residual connections on learning. arXiv e-prints (2016)

36. Chollet, F.: Deep learning with Python. Manning Publications Co., New York (2017)

37. Babenko, A., et al.: Neural codes for image retrieval. arXiv e-prints (2014)

38. Phung, D.Q., et al.: Using Shannon entropy as EEG signal feature for fast person identification. In: 22nd European Symposium on Artificial neural networks, Computational Intelligence and Machine Learning, ESANN (2014)

39. Kumar, M., Pachori, R.B., Acharya, U.R.: An efficient automated technique for CAD diagnosis using flexible analytic wavelet transform and entropy features extracted from HRV signals. Expert Syst. Appl. 63, 165–172 (2016)

40. Goldberger, J., et al.: Neighbourhood components analysis. Adv. Neural Inform. Process. Syst. 17, 513–520 (2004)

41. Elisseeff, A., Pontil, M.: Leave-one-out error and stability of learning algorithms with applications. NATO Sci. Ser. Sub Ser. III Comput. Syst. Sci. 190, 111–130 (2003)

42. Yang, W., Wang, K., Zuo, W.: Neighbourhood component feature selection for high-dimensional data. JCP. 7(1), 161–168 (2012)

43. Jia, Q., et al.: Sequential dual deep learning with shape and texture features for sketch recognition. arXiv preprint arXiv:1708.02716 (2017)

44. Perona, P., Malik, J.: Scale-space and edge detection using anisotropic diffusion. IEEE Trans. Pattern Anal. Mach. Intell. 12(7), 629–639 (1990)

How to cite this article: Hayat S, Kun S, Shahzad S, Suwansrikham P, Mateen M, Yu Y. Entropy information-based heterogeneous deep selective fused features using deep convolutional neural network for sketch recognition. IET Comput. Vis. 2021;15:165–180. https://doi.org/10.1049/cvi2.12019