Multimedia Tools and Applications (2019) 78:17095–17112
https://doi.org/10.1007/s11042-018-7067-1

Sketch recognition using transfer learning

Mustafa Sert[1] · **Emel Boyacı**[1]

Received: 6 April 2018 / Revised: 31 October 2018 / Accepted: 11 December 2018 /
Published online: 3 January 2019

Abstract
Humans have an excellent ability to recognize freehand sketch drawings despite their abstract and sparse structures. Understanding freehand sketches with automated methods is a challenging task due to the diversity and abstract structures of these sketches. In this paper, we propose an efficient freehand sketch recognition scheme, which is based on the feature-level fusion of Convolutional Neural Networks (CNNs) in the transfer learning context. Specifically, we analyse different layer performances of distinct ImageNet pretrained CNNs and combine best performing layer features within the CNN-SVM pipeline for recognition. We also employ Principal Component Analysis (PCA) to reduce the fused deep feature dimensions to ensure the efficiency of the recognition application on the limited-capacity devices. We perform evaluations on two real sketch benchmark datasets, namely the Sketchy and the TU-Berlin to show the effectiveness of the proposed scheme. Our experimental results show that, the feature-level fusion scheme with the PCA achieves a recognition accuracy of 97.91% and 72.5% on the Sketchy and TU-Berlin datasets, respectively. This result is promising when compared with the human recognition accuracy of 73.1% on the TU-Berlin dataset. We also develop a sketch recognition application for smart devices to demonstrate the proposed scheme.

Keywords Sketch recognition · Transfer learning · Convolutional neural networks (CNNs) · Feature fusion

1 Introduction

Nowadays sketches appear in many aspects of daily life and play an important role in education, history, human-computer interaction (HCI), and human relations by expressing strong emotions. For instance, horse, weapon, and tool drawings on the cave walls from ancient times are translated into sketches in our tablets, computers, and phones. On the other hand, most interactions with smartphones are realized by physical movements of the hands on

✉ Mustafa Sert
msert@baskent.edu.tr

Emel Boyacı
21310038@mail.baskent.edu.tr

[1] Department of Computer Engineering, Başkent University, 06790 Ankara, Turkey

device screens. Through the advances in technology, the widespread usage of graphic tablets and other touch-screen display and input devices enabled people to draw sketches and handwrite in digital form. To be able to make it more accessible for searching and retrieving from such generated content, one needs to understand and recognize the content, properly. In order to efficiently organize, and search these sketches in user devices, both computer vision and information retrieval techniques are needed. The very important step to enable such functionalities is to understand and recognize the semantics of these sketches.

The main objective in sketch recognition is to properly label a given sketch into a class among a pre-defined set of categories. In order to perform this classification, sketch recognition studies need to perform useful and robust features from given sketches. In regard to features, the most commonly used low-level representations in sketch recognition are HOG [10], GIST [32], SIFT [31], and Bag of Words (BoW) based local features for image classification. Towards this direction, instead of using hand-crafted features, utilization of deep features and/or sketch-specific deep neural network models also achieves state-of-the-art results for sketch recognition tasks [4, 34, 39].

Specifically, deep learning architectures such as Convolutional Neural Networks (CNNs) have an important use in the field of computer vision in recent years. These architectures, which in essence are multi-layered neural network models, are used not only as classifiers but also to learn effective feature representations of data sets. The CNN features used for this purpose provide great performance returns. These general features were extracted from CNN model layers. As a multi-layer network, CNN architectures are composed of several layers depending on the target domain. These layers are generally referred to as convolution, sampling (pooling), non-linear layer, and fully-connected layers. In these architectures, different layers convey distinct representations of the input data and learning can be performed by selecting proper layer information as the features. On the other hand, some of the recent studies combine different layer features to obtain robust features in computer vision tasks and obtain improvements compared with the individual layer features [16, 18]. Therefore, we combine deep features for efficient representation of the content. This procedure is also referred to as data fusion in the literature, which is the combination of data from multiple sources, processed or correlated [18]. Main types of data fusion can be at the feature level (early fusion) or model level (late fusion). In computer vision tasks, feature level fusion often performed better results compared to the model level fusion, since the features fused at feature level contain richer information than the output decision or matching value of a classifier [43].

One of the main challenges in sketch recognition is that, sketches are not constrained by space and time. It is commonly used by people in different areas such as education, making cartoons, entertainment (playing game), criminal face recognition systems, and so forth. Thus, developing automated and computer based methods for sketch recognition is an important step for enabling various applications in these fields. Today's handheld smartphones have a wide usage in our everyday life due to its increasing computing power and durability. Therefore, we consider smartphones to be an ideal platform for supporting people with an informal sketching tool. Moreover, due to the widespread use of smartphones, sketch recognition application is needed where people can reach to play game, write or draw something in classroom activities everywhere like university, school, or beach.

The CNNs convey different levels of information among their layers. In this study, our main motivation is to exploit the recognition ability of distinct layers of ImageNet [11] pretrained CNN architectures in sketch recognition task. To this end, we employ three robust CNN architectures, namely AlexNet [25], VGG19 [42], and GN-Triplet [38] in analyses and propose an efficient sketch recognition scheme based on the layer-level feature

fusion and collaboration of CNN architectures to capture the different levels of semantics of sketches. Finally, we develop a sketch recognition application for smartphones using the best performing fusion scheme based on the client-server application architecture. The main contributions of this article can be listed as follows:

– We propose an efficient scheme for sketch recognition, which is based on the feature-level fusion of ImageNet pretrained CNN models within the CNN-PCA-SVM pipeline. To the best of our knowledge, the proposed scheme achieves state-of-the art recognition performance on the TU-Berlin [14] and the Sketchy [38] datasets except the CNNs [34, 41] specifically designed and trained for sketches.
– We present extensive analyses to evaluate the performances of both the individual- and joint-CNN models in the feature-fusion context.
– We develop a sketch recognition application for smart devices utilizing the proposed method.

The paper is organized as follows: First, we review related studies in Section 2. We introduce the proposed sketch recognition scheme in Section 3 and describe our feature extraction and fusion schemes. In Section 4, we present our experimental results and evaluations. Finally, we conclude in Section 5.

2 Related work

Sketches have broad range of application areas, such as a way of user interaction with smart devices, sketch-based retrieval of visual content, and generating trajectories resulting from hand/body gestures in Virtual Reality (VR). However, the diversity of sketches and their abstract structures make recognition of sketches a challenging task even for humans. Therefore, there have been various studies to overcome these problems. We review the literature in two parts, which are sketch recognition/classification and sketch recognition applications.

2.1 Sketch recognition

In early phases of the studies, many researchers have focused on improving classification performance by using different classifiers with the help of hand-crafted local and global feature representations [13, 14]. Eitz et al. [13] collect the first dataset of sketches and use it to evaluate human vs. computational recognition rates. They develop a bag-of-features based sketch representation and classify sketches using the Support Vector Machine (SVM) algorithm. They compare their recognition rate of 56% with the human performance, which is also measured as 73% on the dataset. A sketch-based image retrieval (SBIR) system is designed in [14]. They evaluate the bag-of-feature representation of six descriptors in SBIR system and conclude that although the performance of the study is promising, the results are far from the human performance since it requires extensive knowledge on drawings and there is also no guideline about how to construct optimal features. A more recent attempt on sketch-based image retrieval is introduced by Seddati et al. [41]. They design a sketch-based CNN architecture for query-by-example and sketch-based image retrieval tasks. In contrast to standard CNNs, they use residual blocks in their architecture and show that, addition of residual blocks improves the retrieval accuracy. Jahani-Fariman et al. [21], employ VR visualization to enhance the free-hand sketch recognition. They improve the recognition accuracy by utilizing compressed sensing based block-sparse bayesian learning (MATRACK) aproach. Although they achieve high classification accuracy, the method

needs to convert each sketch from the dataset to a multi-dimensonal signal beforehand using a Wii-mote input device. An interesting application of sketches appears in video retrieval domain. Wu et al. [49], propose a method to search video content by using hand-drawn motion sketches to define typical motion patterns of desired objects. They conclude that, using hand-drawn motion sketches enhances the expression ability of user queries.

In latest studies [27, 34, 40], several attempts have been made to predict sketch labels like *airplane*, *seagull*, etc. in human sketch datasets. In [27, 34], multiple kernel learning (MKL) SVMs and Fisher Vectors (FV) are utilized instead of local features (e.g., HOG and SIFT) along with Bag-of-Word (BoW) representation, and significantly better performances are achieved. Li et al. [27] propose a novel structured representation of a sketch to capture the holistic structure and achieve state-of-the-art recognition performance. They use MKL SVM for sketch recognition by fusing several features common to sketches and report an accuracy of 65.81%. The Fisher Vector method [40] achieves an accuracy of 68.9% and outperforms the MKL method. However, the Fisher Vector method has the drawback of memory complexity due to its higher dimensionality. On the other hand, gathering adequate number of pre-labelled data is also crucial for training supervised machine learning algorithms, especially when considered the diversity of free-hand sketches. Liu et al. propose an iterative sketch collection annotation method [29] to cope with this problem. They discover the categories of the collections iteratively in three-stages. Their method is based on online learning including the user at each iteration and perform metric learning by semi-supervised clustering to annotate sketch collections.

In recent years, the CNNs have demonstrated excellent performance in learning semantics of audio-visual data [16, 24, 25, 42]. This success can be described by the generalization ability of the CNN architectures. In practice, training an entire CNN from scratch with random initialization is a costly task due to the lack of sufficient datasets and/or processing power. Therefore, a typical recent usage of deep CNN models is to reuse a model, which is developed for a specific task, as a starting point for a model on related problems. This approach is also referred to as transfer learning and can be a practical way when lack of sufficient pre-labelled data. In visual concept recognition tasks, the ImageNet pretrained CNN models, such as Alexnet [25], VGG [42], and GN-Triplet [38] have been widely used for this purpose. In particular, internal layers of CNNs convey different levels of semantics and these layer features can also be used as descriptors instead of hand-crafted features by machine learning algorithms. This raises the question whether the CNN models that are pretrained on photographic images can be applied to freehand sketch-recognition problem. One attempt on using existing ImageNet pretrained CNN architectures on sketch recognition problem has been made in [4]. In their study, different layers of the AlexNet and the VGG19 architectures are fused based on the experiments and they show that the standalone CNN layer feature accuracy can be improved by fusing different network layers. Guo et al. address the problem of oracle character recognition of Chinese characters and evaluate visual representations in shape-related works [19]. They combine low-level (FD-SIFT) and mid-level (BoW) representations to recognize oracle characters and achieve an accuracy of 89.2% on the Oracle-20K dataset by feeding the combined feature representations (FD-SIFT+BoW) into a kernel SVM. Also, they perform experiments on transfer learning in oracle character recognition task and report that, using additional data with ImageNet pretrained CNN models (AlexNet, VGGNet, and GoogLeNet) improves the performance significantly (94.2%) in contrast to the without using external data.

Another practical application of free-hand sketches appears in law enforcement and digital entertainment, where a photo image is transformed into a sketch and it is also referred to as face sketch synthesis. The Generative Adversarial Networks (GANs) [17] have shown

promising results on image-to-image translation problems (photo-to-sketch synthesis in particular) as well as in other tasks, such as image editing/generation and representation learning [12, 35, 37, 54]. The main aim of the GANs is to use two networks, namely a generative network (G) and a discriminative network (D), to learn excellent representations of data. The success of GANs led to its derivative algorithms, such as CGAN [30], DCGAN [35], infoGAN [8], cycle GAN [55], WGAN [3], and Dual GAN [51]. As a derivative of GANs, the cGAN learn a conditional generative model, whereas GANs learn a generative model of data. Mirza et al. [30] extended the GANs to a conditional model (cGAN) by feeding auxiliary information to both the generator and discriminator models. They not only show how both models (G and D) can be conditioned but also illustrate how cGAN can enable to learn multi-modal model. Some studies showed cGANs can give reasonable results on a wide variety of graphics and vision tasks, including photo generation, semantic segmentation, where image-to-image translation is needed [20]. Creswell and Bharath [9] applied the representations learned by GANs to sketch retrieval problem. They introduced a novel GAN architecture that is specifically designed for sketch retrieval and report an increase of stability to rotation and scale compared to the standard GANs.

In the image-to-image translation task, the main goal is to learn a mapping between input and output images using image pairs (e.g., photo-sketch). Zhu et al. [55] addresses this issue and propose a learning approach, which is also referred to as CycleGAN, to translate an image when paired training examples do not exist. Since the main idea behind the success of GANs is to use adversarial loss and this loss is specifically robust in image generation tasks, the CycleGAN adopts an adversarial loss in a way such that translated images cannot be distinguished from images in the target domain. In particular, the CycleGAN utilize the cycle consistent property of translations, so that for a given two translators G and F on domains X and Y, $G : X \rightarrow Y$ and $F : Y \rightarrow X$, the mappings G and F should be inverses of each other. To this end, the property is applied by training both the mappings simultaneously and adding a cycle consistency loss [53]. In the end, the overall objective of the CycleGAN is given by combining this loss with adversarial losses on both domains. Chen et al. [7] proposed SketchyGAN, a specifically designed deep learning architecture for the image synthesis from sketches. The architecture favors from GANs and make use of data augmentation technique to address the lack of sufficient human annotated training data. Although GAN based methods achieve compelling results, they also have some limitations such as the tasks on handling some transformations that require geometric changes or generating high-resolution realistic images (e.g., photo-to-sketch synthesis). An attempt on this direction is motivated by using both multi-adversarial networks and the CycleGAN framework and addressed high-quality facial photo-sketch synthesis problem [48]. An attempt on this direction is motivated by using both multi-adversarial networks and the CycleGAN framework and proposed Photo Sketh MAN (PS2-MAN) framework [48] for high-quality facial photo-sketch synthesis problem. The PS2-MAN framework of Wang et al. demonstrates superior results in comparison to existing state-of-the-art generative solutions (CycleGAN [55], pix2pix [20], DualGAN [51]) on both image- and sketch-matching rates.

2.2 Sketch recognition applications

Nowadays, the technology becomes an indispensible part of our daily lives and humans interact with smart devices more and more. For this reason, there is a huge demand on all kind of applications and novel interfaces in the last decade. As a way of Human Computer Interaction (HCI), there is also a need efficient algorithms for sketch recognition on smart devices. One attempt towards this direction is the A.I. Experiment application named

Quick Draw! – which is a Web-based drawing game built with machine learning. Specifically, the application tries to guess what users draw by using neural nets. A sketch-based image retrieval tool that can be used on mobile devices due to its low memory consumption is detailed in [46]. They propose to utilize visual hashing bits to compact raw visual descriptors. They use high-dimensional distance transform (DT) features and further project those features to more compact binary hash bits. Xiao et al., [50] propose a PPTLens system to convert sketch images taken by smart phones to digital flowcharts in PowerPoint. Their stroke extraction method identifies the borders of the whiteboard or paper for cropping and rectification. Finally, the image is represented in binary form by leveraging the Stroke Width Transform and Markov Random Field (MRF) optimization.

In this study, our aim is to experiment popular CNN architectures in both feature-level fusion and transfer learning context by providing a set of descriptor analyses on CNN layers concerning sketch recognition accuracy and demonstrate our results on the TU-Berlin [14] and the Sketchy [38] datasets.

3 Methodology

In this section, we introduce our proposed scheme for sketch recognition and demonstrate it with a smart device application. The general structure of our sketch recognition scheme is illustrated in Fig. 1. In the followings, we first describe the modules of the proposed scheme and then introduce our sketch recognition application.

3.1 Proposed sketch recognition scheme

Freehand sketches often contain different levels of visual detail. In contrast to photographic images, included data in sketch-images are often sparse. In recent years, CNNs have emerged as powerful architectures for feature representation and recognition especially in computer vision tasks [45]. Since these "deep" networks have different representations of the input data in their hidden layers, we aim to benefit from best performing layers of CNN architectures by employing data fusion technique. In our study, we consider three robust CNN architectures, namely the AlexNet, VGG19, and GN-Triplet due to their success in

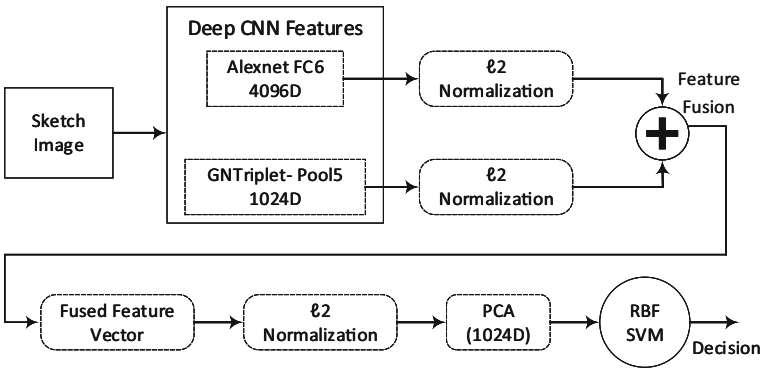


Fig. 1 Proposed sketch recognition scheme

computer vision tasks [44]. The details of these architectures, our feature extraction and data fusion scheme are described in the following sections.

3.1.1 Deep feature extraction

As in many computer vision problems, feature extraction plays a key role in sketch recognition and significantly effects the recognition accuracy. Recent studies show that, CNN architectures can be used as a simple image descriptor and have demonstrated superior or competitive results in computer vision applications [36]. The most common way of using CNN architecture as a feature descriptor is to supply an image to the architecture and using one of the fully-connected layers as an image (global) feature (descriptor). Therefore, we employ deep CNN features from fully-connected layers based on this motivation and also consider to use encoded or aggregated local features from pooling layers to enable data fusion for sketch recognition. Since training CNN models is a complicated and relatively a costly task, we select three well-known ImageNet pretrained networks, namely AlexNet, VGG19, and GN-Triplet. We utilize the Caffe framework [22] while generating deep learning models and extracting CNN features. We present properties of the utilized CNN architectures in Table 1.

The AlexNet architecture consists of 5 convolutional, 3 fully connected, and one softmax layers. This CNN model is the first successful demonstration of CNNs at such large scale [25]. In addition, Angelova et al. [1] state that, the AlexNet architecture is successful in image classification tasks. On the other hand, the VGG19 model contains 19 layers, 3 of which are fully connected. This architecture is based on smaller convolutional filters in all layers of networks. Thus, it increases network depth compared to the big convolutional filters, such as Conv4, without increasing the layer complexity. We also employ the GN-Triplet CNN architecture, consisting of 22 layers. The GN-Triplet architecture is a relatively new architecture compared with the AlexNet and VGG19. The GN-Triplet has been designed with GoogLeNet trained with Triplet and classification loss.

Typical CNN layers such as convolution, pooling, and fully-connected (FC) carry different levels of information regarding the learned concepts. It is shown that, using FC layers as features achieves better accuracy in visual concept recognition applications compared to convolution and pooling layers [15, 45]. In addition, the last layers have fewer dimensions compared to convolution layers, which is an important issue in terms of memory and time complexity of smart devices. Therefore, we extract the features from last layer (Pool5) of the GN-Triplet model. It is also demonstrated that, the layer FC6 of the AlexNet and the layer Pool5 of the VGG19 have relatively better recognition accuracy [4] and hence we also use the features generated from these layers in our scheme.

Table 1 Layer dimensions of the utilized layers of the employed CNN architectures

Layer name	Network	Dimension	Output geometry
FC6	AlexNet	4096	1x1x4096
FC7	AlexNet	4096	1x1x4096
FC8	AlexNet	1000	1x1x1000
FC6	VGG19	4096	1x1x4096
POOL5	VGG19	25088	7x7x512
POOL5	GN-Triplet	1024	1x1x1024

To sum up, we extract the features from the layers Pool5, FC6 and Pool5 of the CNN architectures VGG19, AlexNet and GN-Triplet, respectively. The dimensions of these layers are given in Table 1. Prior to the feature fusion procedure in our scheme, we also apply the ℓ_2 normalization to the feature vectors generated from various layers of different CNN models (Fig. 1).

3.1.2 Feature level fusion

Data fusion is the operation that aggregates the processing and association of data from different sources. There are two main data fusion approaches, which are also referred to as early (feature-level) and late (model-level) fusion in the literature. Our feature-level fusion strategy is mainly based on the concatenation of the pooling and the FC layers and thus falls into the early fusion category. In addition, we also consider to fuse the layers of different CNN models to benefit from their distinctive abilities.

In our fusion scheme, let matrices $A \in \mathbb{R}^{d \times N}$ and $B \in \mathbb{R}^{d \times M}$ are two arbitrary CNN features to be fused, where N and M are the number of feature maps (e.g., 4096, 1024) and d denotes the size of feature maps (e.g., 1×1, 7×7). Both the number and size of feature maps can vary based on the utilized CNN architecture (Table 1). Note that, $d = 1$ for the fully connected layers and hence the dimensions of A and B is defined as $1 \times N$ and $1 \times M$, respectively. Let $a_i \in A$ and $b_i \in B$ denote the ith column of the feature matrices and correspond to separate feature maps. We define feature fusion on A and B using the *concatenation* operator, $\|$, where values of these feature matrices are concatenated into one feature matrix denoted by $F_{\|}$:

$$F_{\|} = [a_i \| b_i]^T \tag{1}$$

Feature matrices that are obtained from distinct layers of the architectures are different in their sizes and may have different scales. In order to unify these scales, we apply ℓ_2 norm before and after the concatenation operator (Fig. 1).

Table 2 Sketch recognition performances of the proposed system on the TU-Berlin dataset

Model (architecture)	Layer	Method	Accuracy (%)	Avg. feature extraction time (sec.)
AlexNet	FC8	–	56.20	0.015848887
AlexNet	FC7	–	59.23	0.015748459
AlexNet	FC6	–	67.26	0.010501853
VGG19	FC6	–	60.30	0.01927468
VGG19	Pool5	–	64.14	0.011998142
GN-Triplet	Pool5	–	68.16	0.017179906
AlexNet	FC6-FC7	Early Fusion	66.50	–
Alexnet-VGG19	FC6-FC8	Early Fusion	67.15	–
Alexnet-VGG19	FC6-FC6	Early Fusion	68.25	–
Alexnet-VGG19	FC6-Pool5	Early Fusion	69.175	–
VGG19-GN Triplet	Pool5-Pool5	Early Fusion	69.71	–
AlexNet-GN Triplet	FC6-Pool5	Early Fusion	70.785	–
AlexNet-GN Triplet	FC6-Pool5	Early Fusion + PCA	72.5	–

Based on our analyses as depicted in Tables 2 and 3, we obtained superior performance on the utilized datasets when using the Alexnet-FC6 and the GN Triplet-Pool5 model layers and hence used these layers in the final recognition system.

3.1.3 Dimension reduction and classifier design

Feature level fusion strategies have certain advantages compared to the late (model) fusion strategies, since different feature vectors exhibit distinct characteristics of some patterns and using these features in a combined form keeps effective discriminative information about the data in hand. However, one drawback of such strategies, like the *concatenation* operator we utilized in our scheme is that, the *concatenation* operator clearly exploits the final feature dimension and as a result may cause to the curse-of-dimensionality problem for the learning algorithm. To cope with this problem, and also to preserve the distinctive properties of features, we employ *Principal Component Analysis* (PCA) to reduce the feature dimension, which is also a crucial procedure for devices that have limited computational power [23]. Simply, the PCA method seeks to map data points from a high dimensional space to a low dimensional space while preserving the linear structure intact. We give the concatenated and ℓ_2 normalized feature vector of the best performing CNN architectures to the PCA method by providing the number of dimension that will be retained in the mapping. We select the final (reduced) feature dimension as 1024**D** based on the trade-off between performance and accuracy. Mathematically, assume that $X = (x_1, x_2, ..., x_m)$ is a set of M training instances, where each instance $x_i \in R^{dxN}$. The d and N represent the number of feature maps and their sizes, respectively. The mean of all feature descriptors as a single vector in R^{dxN}:

$$\mu = \frac{1}{M}\sum_{i=1}^{M} x_i \tag{2}$$

In order to re-center the data points in R^{dxN}, each feature descriptor differs from the mean by vector:

$$\phi_i = x_i - \mu \tag{3}$$

The covariance matrix S is defined as:

$$S = \frac{1}{M-1}\phi\phi^T \tag{4}$$

Since S is a symmetric matrix, it can be orthogonally diagonalized by the theorem:

$$S\upsilon_i = \lambda_i \upsilon_i \tag{5}$$

Table 3 Sketch recognition performances of the proposed system on the *Sketchy* dataset

Model (architecture)	Layer	Method	Accuracy (%)	Avg. feature extraction time (sec.)
AlexNet	FC6	–	80.89	0.008912873
VGG19	Pool5	–	77.76	0.010022747
GN-Triplet	Pool5	–	95.16	0.01760935
AlexNet-VGG19	FC6-Pool5	Early fusion	82.18	–
VGG19-GN Triplet	Pool5-Pool5	Early fusion	95.47	–
AlexNet-GN Triplet	FC6-Pool5	Early fusion	96.84	–
Alexnet-GN Triplet	FC6-Pool5	Early fusion + PCA	97.91	–

where υ_i and λ_i represent orthogonal eigenvectors and eigenvalues, respectively and these eigenvectors are also referred to as the principal components. We select 1024 eigenvectors yielding a feature size of 1024**D** instead of 5120**D** (i.e., the concatenated dimension of Alexnet and GN-Triplet).

Our fusion scheme follows a CNN-SVM pipeline. That is, we feed the features extracted from the selected CNN architectures to an SVM classifier after fusing and applying the PCA. We use LibSVM [5] library for the SVM algorithm and make use of one-versus-all (OVA) strategy to cope with the multiclass classification problem. We use *radial basis function* (RBF) as the kernel of the SVM and apply *grid*-search algorithm to optimize the kernel parameters. Let $X = (x_1, x_2, ..., x_n)$ be a set of n feature vectors (instances), where $x_i \in X$ for all $i = 1, ..., n$. For a given feature vector x_i, our goal is to learn a mapping from x to y, for a given training instance of form (x_i, y_i), where $y_i \in Y$ is the target labels of the instances x_i. We use Platt's approximation [33] for the SVM, which means we produce posterior probabilities for each sketch category in the output vector. Let $F_{\|}$ represents the concatenated supervector of $x_i \in X$, C denotes the number of sketch categories in the dataset, and $O = \langle o_1, o_2, ..., o_C \rangle^T$ is the probabilistic output vector of the SVM classifier built by the OVA strategy. The decision, D of the classifier is formulated as follows:

$$D \in \underset{i=1,2,...,C}{\mathbf{argmax}} o_i \tag{6}$$

3.2 Service-based sketch recognition application for smartphones

We design and implement an application to recognize sketches in smart devices. The architecture of our sketch recognition application is illustrated in Fig. 2. The application consists of two modules: Sketch Recognition Service (SRS) and Mobile Sketch Application (MSA). The SRS is a server-side web service application that performs sketch processing and recognition tasks and it is hosted by an application server.

The MSA allows the user to draw a sketch and to send sketch data to be recognized to the SRS via our web service. The SRS performs the recognition task as follows: The service first extracts features from the layers of the utilized CNN architectures, performs ℓ_2 normalization, applies fusion operator (i.e., concatenation), performs the ℓ_2 normalization on the fused features, applies dimension reduction (PCA), predicts the sketch category by

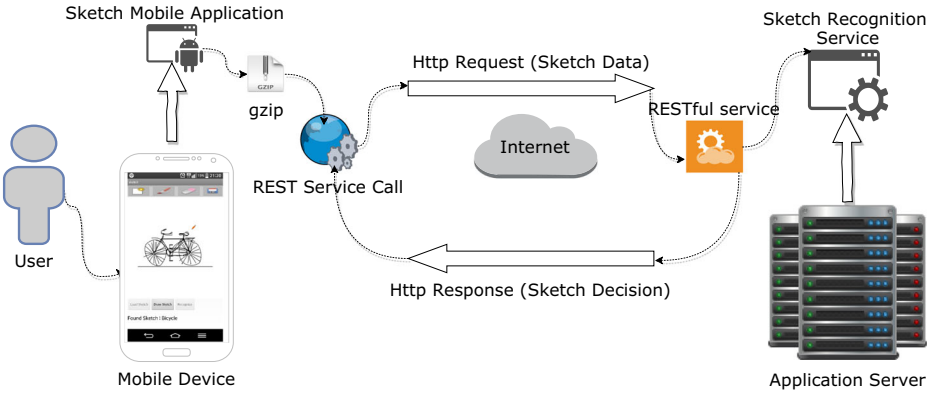


Fig. 2 System architecture of the developed sketch recognition application

using the trained SVM models using the OVA strategy, and finally sends the results back to the MSA. In the end, sketch definition is shown to the user within the MSA interface.

As for the SRS, there exists two widely used Web service architectures, namely Simple Object Access Protocol (SOAP) and Representational State Transfer (REST). While the SOAP has been the dominant approach for web service interfaces for a long time, the REST architecture is quickly winning out and becoming more and more widespread [2, 47]. The REST architecture has some advantages over SOAP especially in mobile applications, such as changing services that use SOAP often means a complicated code change on the client side. In addition, SOAP client side code generation and implementation from Web Services Description Language (WSDL) and XML Schema Definition (XSD) can be complex.

Therefore, problem of application update dissemination arises on mobile application. On the other hand, the REST has a lightweight and flexible architecture that is relatively easy to implement and maintain. While the SOAP services always return XML, the REST services provide flexibility in regards to the type of data returned. The de facto standard for data payloads from REST services is the Java Script Object Notation (JSON). The JSON payloads are usually smaller than their XML counterparts. Consequently, SOAP spends a lot of bandwidth communication metadata. The use of RESTful services, with data in JSON format is a better choice for mobile applications whether the client device platform is iOS or Android.

As a result, we preferred the RESTful service as the implementation of our sketch recognition service to ensure that applications run smoothly with low latency in order to minimize the use of network bandwidth which is more limited on mobile devices.

4 Experiments and evaluations

In this section, we present our experimental results and evaluations on the utilized datasets.

4.1 Utilized datasets

In this study, we use two different sketch datasets. For the first dataset, we use the largest hand-free sketch benchmark dataset, the TU-Berlin in our experiments [14]. The dataset contains 250 sketch categories having 20000 sketches in total. Some of the categories in the dataset are *seagull*, *sun*, and *snowman*. The dataset includes 80 sketches for each category. The sketches are gathered from 1,350 participants by Amazon Mechanical Turk. Sample sketches from this dataset is illustrated in Fig. 3.

The second dataset, namely the Sketchy is a large-scale collection which consists of sketch-photo pairs [38]. The dataset was created with 12,500 unique photographs of objects and 75,481 human sketches in 125 categories. The Sketchy dataset is open for improving sketch and image understanding. The dataset considers all the ImageNet categories and

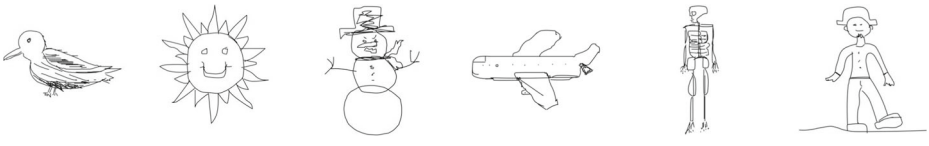


Fig. 3 Sample sketches from *TU*-Berlin dataset

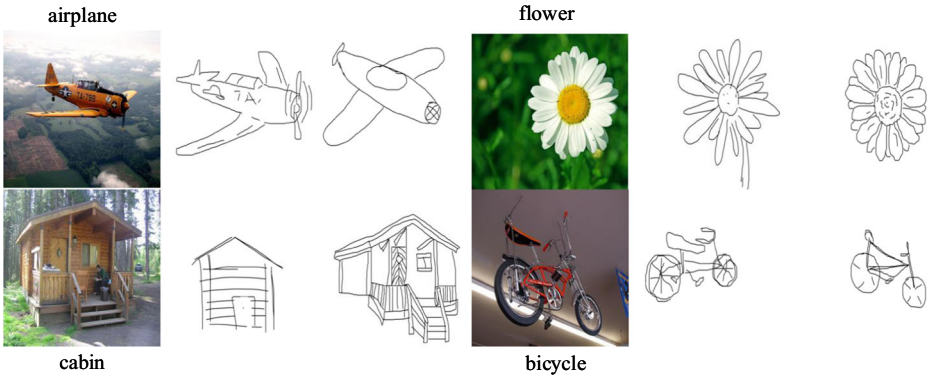


Fig. 4 Sample sketches from the *Sketchy* dataset

cover a large number of common objects of the TU-Berlin sketch dataset. Example instances of this dataset can be seen in Fig. 4.

All the evaluations are performed using 3-fold cross validation to enable comparisons with alternatives.

4.2 Results and evaluations

We conducted two different tests to evaluate our methods. In the analyses, we consider standalone tests on the Pool5, FC6, Pool5 layers of the GN-Triplet, the AlexNet and the VGG19 CNN architectures, respectively. We also exploit two best performing CNN architectures when used in a combined form. All of the tests follow the CNN-SVM pipeline given in Fig. 1. We show our results for the TU-Berlin and Sketchy datasets in Tables 2 and 3, respectively. We also compare our results with a sketch-based CNN architecture [34] to provide a comparison.

Based on the results of the TU-Berlin dataset, following observations can be made: Layer FC6 is the best performing FC layer for the AlexNet with an accuracy of 67.26%. Our results also show that; earlier FC layers perform better than the successor layers of the VGG19 architecture. Layer Pool5 of VGG19 achieves higher accuracy rate with 64.14% compared with the FC6 (60.30%), albeit its dimensionality is bigger. This result significantly outperforms all existing methods, except the one by Qian et al. [34], which is a CNN architecture specifically designed for sketches. Based on our experiments on the TU-Berlin dataset, we achieve best result when we fuse the FC6 and the Pool5 layers of the Alexnet and the GN-Triplet architectures, respectively. Our recognition accuracy of 72.5% is near the human performance (73.1%) on the same dataset. This result also outperforms the HOG-SVM (56%) [14], ensemble matching (61.5%) [28], MKL-SVM (65.8%) [27], AlexNet-SVM (67.1%) [25], and the LeNet (55.2%) [26] methods in the literature (Fig. 5).

For the Sketchy dataset, we achieve the best individual layer performance of 95.16% by the Pool5 layer of the GN-Triplet pretrained model in comparison to the AlexNet and the VGG19 models. In the case of joint layers, the AlexNet–GN-Triplet combination achieves best accuracy with 96.84% among the other combinations. We improve this recognition rate to 97.91% by using the PCA. This result confirms our experiments for the TU-Berlin dataset and we can conclude that using the PCA in the proposed processing pipeline clearly improves the recognition accuracy of the sketches. This is also valid for joint layer features compared with the standalone CNN models.

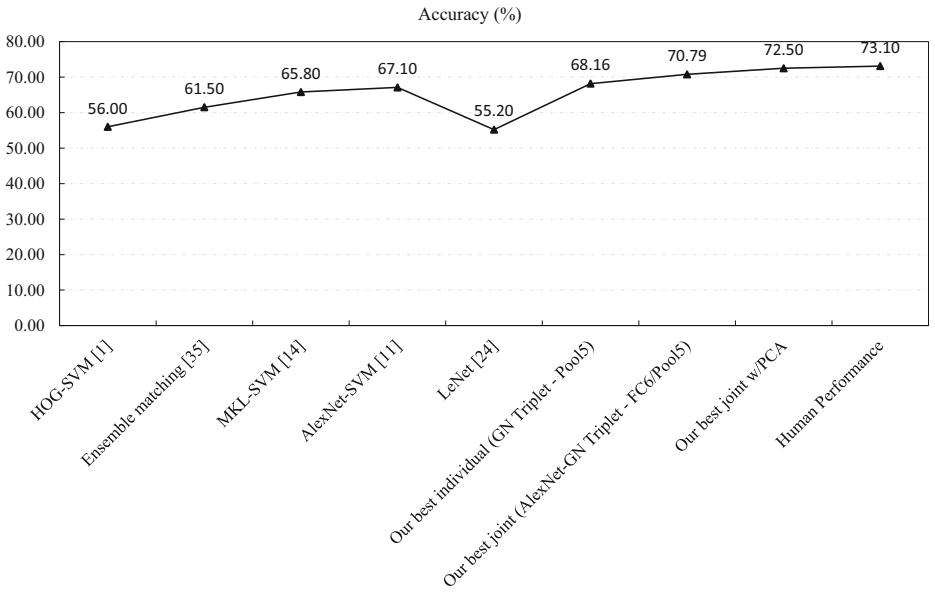


Fig. 5 Comparison with state-of-the art results on sketch recognition

In sketch recognition, one serious problem is that the sketch recognition heavily relies on the manual feature extraction which may be very time consuming. Therefore, we conducted feature extraction experiments to provide efficiency data for deep features. To this end, we measured feature extraction times of the utilized individual layers of the pre-trained CNN architectures and presented the results in Tables 2 and 3. Specifically, for both datasets (TU-Berlin and Sketchy), we extracted deep features from the (utilized) individual layers of the models and measured elapsed feature extraction time for each sample. Then we calculated the average times of all samples and noted them in the tables. All experiments were carried out on computer equipped with Nvidia GTX 1070 Ti GPUs using keras deep learning framework and Cuda 9.0, CuDNN 7.1 library. The average feature extraction times for the TU-Berlin and the Sketchy datasets are shown in Fig. 6a and b, respectively.

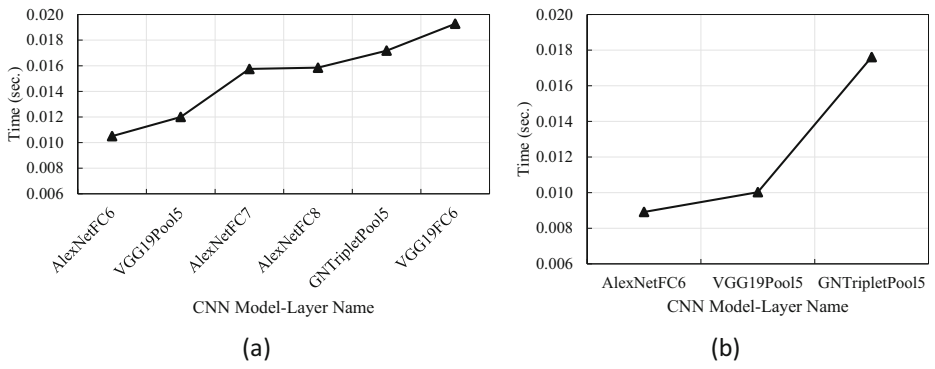


Fig. 6 Average feature extraction time for one sample using the utilized CNN layers: **a** the TU-Berlin dataset, **b** the Sketchy dataset

Based on the experiments, we observe that extracting deep features for a sample from individual layers varies between 8-19 milliseconds for both datasets. For our configuration, considering the 20000 samples in the TU-Berlin dataset, this means approximately 160-380 seconds to extract all features and approximately 603-1434 seconds for the 75481 samples of the Sketchy dataset.

5 Limitations and discussion

One of the important drawback associated with CNN is that their limited ability against to geometric invariance property. Although our method can achieve compelling results in comparison to human performance, it may be vulnerable to geometric variations, such as rotations, scaling, and translations due to its underlying CNN architecture.

To analyze this situation, we have also explored the feature activations of CNN against to those geometric variations. Figure 7 shows our results on the AlexNet for some geometric changes. The first column on the left depicts the input sketch images. The other columns from left to right illustrates the strongest activations obtained from the conv1, conv2, conv3, conv4, and conv5 layers of the net for each input sketch. In the activation figures, white pixels represent strong positive activations and black pixels represent strong negative activations. A channel that is mostly gray does not activate as strongly on the input image and only

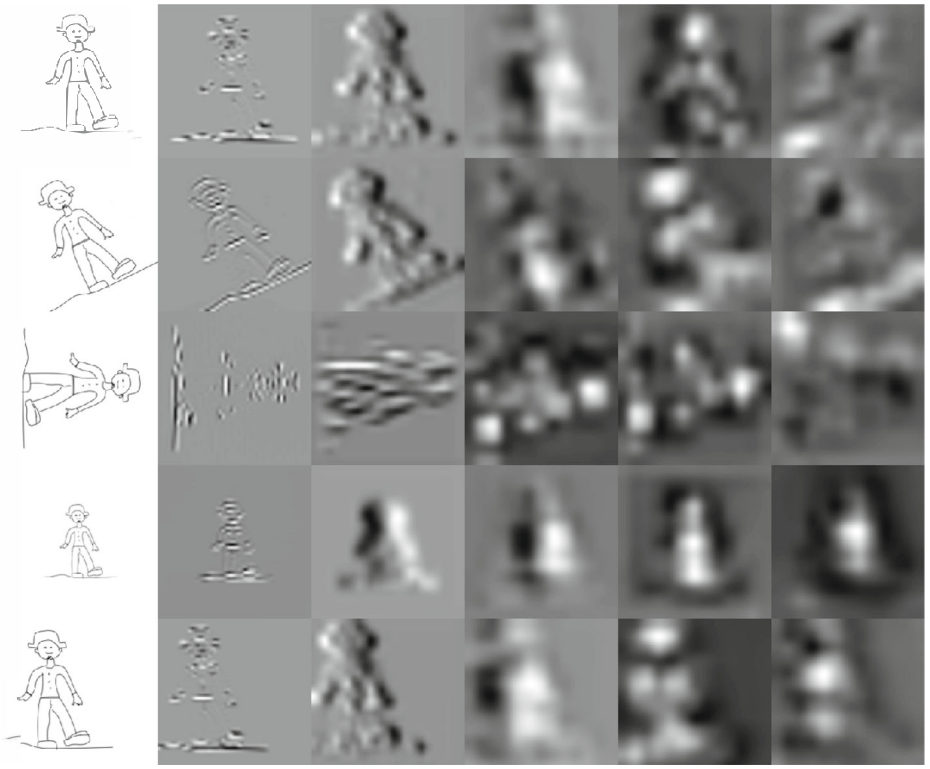


Fig. 7 The AlexNet activations. From left to right, the columns represent the input sketch, conv1, conv2, conv3, conv4, and conv5 outputs, respectively. Original image (1st row), rotation (2nd & 3rd rows), scaling (4th row), translation (5th row)

the positive activations are used in the network because of the rectified linear units (ReLU) following the convolutional layers. As can be seen from the top first row, the conv1 layer activates positively on white-black edges, and negatively on black-white edges. Although not all layer representation are understandable, we can say that the conv3, conv4, and the conv5 layers of the network focusing on the righ side of the body, the face, and the left foot in the input sketch, respectively. This result also confirms that the network learns mid-level representations while going into to the deeper layers. The second and third rows depict random-angle rotations on the original image, while the fourth and the fifth rows show the scaling and translation cases.

Based on our results we observe that the ability of the network is more limited to scaling variations compared to the rotation and translation changes. This might be caused by the size of the filters in the convolutional layers, in particular the size of the first convolutional layer might be the most sensitive layer considering all subsequent operations depends on the first layer output of the network. In our case, we achieve the best performance with the AlexNet and the GN-Triplet combination utilizing $11x11$ and $7x7$ filters in their first convolutional layers, respectively. Based on our results, the VGG19 comes in the third place in individual feature performance and it uses a filter size of 3x3 in its first convolutional layer. Therefore, we can conclude that the larger filter size is more successful for our dataset in use. As the sketches lack texture information, the filter size can be selected based on the sketch context in use; i.e., larger filters may be used to capture more structured context based on the sketch characteristics.

There are also other techniques to address the geometric invariance problem of CNNs. Common practices show that data augmentation techniques (flips, random crops, translations, scaling, etc.), average pooling [6], and multi-scale pyramid pooling [52] also help to achieve better geometric invariance of CNNs. We believe that, utilizing these common practices can be a good extension to further improve the recognition performance of our method.

6 Conclusions

In this paper, we present feature level fusion scheme that uses CNNs to recognize freehand sketches and develop a sketch recognition application for smart devices to demonstrate the proposed scheme. We employ three well-known ImageNet pretrained CNN models, namely AlexNet, GN-Triplet, and VGG19 in both feature-level fusion and transfer learning context by providing a set of descriptor analyses on CNN layers concerning sketch recognition accuracy.

Based on our experiments on the TU-Berlin [14] and the Sketchy [38] datasets, the AlexNet FC6 and the GN-Triplet Pool5 layer combination achieves the best result. The PCA method not only reduces the fused feature dimensions, also improves the recognition accuracy about 2% for the utilized datasets. To the best of our knowledge, our best result on the TU-Berlin dataset significantly outperforms the existing methods, except the CNN architectures [34, 41] that are specifically designed and trained for sketches. Our results is also very competitive with the human recognition accuracy of 73.1% on the same dataset. We achieve an accuracy of 97.91% for the Sketchy dataset, which is quite higher than the TU-Berlin results. This difference is due to the fact that the dataset includes photo-sketch pairs of each sketch category and these pairs are used while retraining the pretrained CNN models. The proposed scheme can be used solely in applications where both image and sketches are involved, such as image/sketch retrieval and matching forensics sketches to images.

Our future works lie on two directions. One of them is to benefit from late fusion schemes. Another direction would be to explore the potential use of the proposed scheme in forensics domain.

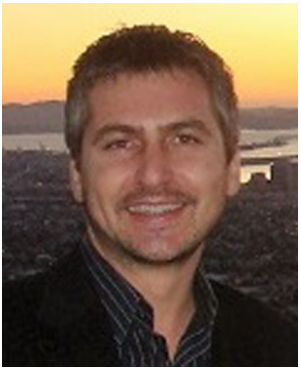
Acknowledgments The authors thank Berkay Selbes for running the feature extraction time experiments.

References

1. Angelova A, Krizhevsky A, Vanhoucke V, Ogale A, Ferguson D (2015) Real-time pedestrian detection with deep network cascades
2. Aihkisalo T, Paaso T (2012) Latencies of service invocation and processing of the REST and SOAP Web service interfaces. In: 2012 IEEE 8th world congress on services. Honolulu, pp 100–107
3. Arjovsky M, Chintala S, Bottou L (2017) Wasserstein GAN. arXiv:1701.07875
4. Boyaci E, Sert M (2017) Feature-level fusion of deep convolutional neural networks for sketch recognition on smartphones. In: Proceedings of IEEE international conference on consumer electronics (ICCE2017), January 8-10, 2017, Las Vegas, Nevada, USA, pp 485–486
5. Chang C, Lin C (2011) LIBSVM: a library for support vector machines. ACM Trans Intell Syst Technol 2(3):27:1–27:27
6. Chatfield K, Simonyan K, Vedaldi A, Zisserman A (2014) Return of the devil in the details: delving deep into convolutional nets. In: Proceedings of British machine vision conference (BMVC)
7. Chen W, Hays J (2018) SketchyGAN: towards diverse and realistic sketch to image synthesis. arXiv:1801.02753
8. Chen X, Duan Y, Houthooft R, Schulman J, Sutskever I, Abbeel P (2016) InfoGAN: interpretable representation learning by information maximizing generative adversarial nets. In: Proceedings of the 30th international conference on neural information processing systems (NIPS'16). Curran Associates Inc., pp 2180–2188
9. Creswell A, Bharath AA (2016) Adversarial training for sketch retrieval. In: Computer vision - ECCV 2016 workshops, lecture notes in computer science, vol 9913. Springer, Cham, pp 798–809
10. Dalal N, Triggs B (2005) Histograms of oriented gradients for human detection. In: Proc. IEEE Comput soc conf comput vis pattern recognit (CVPR), pp 886–893
11. Deng J, Dong W, Socher R, Li L-J, Li K, Fei-Fei L (2009) ImageNet: a large-scale hierarchical image database. IEEE Computer Vision and Pattern Recognition (CVPR)
12. Denton EL, Chintala S, Fergus T et al (2015) Deep generative image models using a Laplacian pyramid of adversarial networks. In: NIPS
13. Eitz M, Hildebrand K, Boubekeur T, Alexa M (2011) Sketch-based image retrieval: benchmark and bag-of-features descriptors. IEEE Trans Visual Comput Graph 17(11):1624–1636
14. Eitz M, Hays J, Alexa M (2012) How do humans sketch objects? ACM Trans Graph 31(4):1–10
15. Ergun H, Akyuz YC, Sert M, Liu J (2016) Early and late level fusion of deep convolutional neural networks for visual concept recognition. Int J Semant Comput 10(03):379–397
16. Ergun H, Sert M (2016) Fusing deep convolutional networks for large scale visual concept classification. In: IEEE international conference on multimedia big data (BigMM2016)
17. Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S (2014) Generative adversarial nets. In: Advances in neural information processing systems 27. Curran Associates, Inc., pp 2672–2680
18. Guo J, Gould S (2015) Deep CNN ensemble with data augmentation for object detection. arXiv:1506.07224
19. Guo J, Wang C, Roman-Rangel E, Chao H, Rui Y (2016) Building hierarchical representations for oracle character and sketch recognition. IEEE Transactions on Image Processing (TIP)
20. Isola P, Zhu J, Zhou T, Efros AA (2017) Image-to-image translation with conditional adversarial networks. In: 2017 IEEE conference on computer vision and pattern recognition (CVPR). Honolulu, pp 5967–5976
21. Jahani-Fariman H, Kavakli M, Boyali A (2018) MATRACK: block sparse Bayesian learning for a sketch recognition approach. Multimed Tools Appl 77(2):1997–2012
22. Jia Y, Shelhamer E, Donahue J, Karayev S, Long J, Girshick R, Guadarrama S, Darrell T (2014) Caffe: convolutional architecture for fast feature embedding. In: Proceedings of the 22nd ACM international conference on multimedia, pp 675–678

23. Jolliffe L (1986) Principal component analysis. Springer, New York
24. Karpathy A, Toderici G, Shetty S, Leung T, Sukthankar R, Fei-Fei L (2014) Large-scale video classification with convolutional neural networks. In: Proceedings of the 2014 IEEE conference on computer vision and pattern recognition, pp 1725–1732
25. Krizhevsky A, Sutskever I, Hinton GE (2012) ImageNet classification with deep convolutional neural networks. Adv Neural Inf Process Syst, 1097–1105
26. LeCun YA, Bottou L, Müller KR, Orr GB (2012) Efficient BackProp. In: Montavon G, Orr GB, Müller KR (eds) Neural networks: tricks of the trade. Lecture notes in computer science, vol 7700, pp 9–48
27. Li Y, Hospedales TM, Song YZ, Gong S (2015) Free-hand sketch recognition by multi-kernel feature learning. Comput Vis Image Underst 137(C):1–11
28. Li Y, Song Y, Gong S (2017) Sketch recognition by ensemble matching of structured features. In: BMVC
29. Liu K, Sun Z, Song M et al (2017) Iterative samples labeling for sketch recognition. Multimed Tools Appl 76(10):12819–12852
30. Mirza M, Osindero S (2014) Conditional generative adversarial nets. arXiv:1411.1784
31. Nowak E, Jurie F, Triggs B (2006) Sampling strategies for bag-of features image classification. In: Computer vision - ECCV. Springer, New York, pp 490–503
32. Oliva A, Torralba A (2001) Modeling the shape of the scene: a holistic representation of the spatial envelope. Int J Comput Vis 42(3):145–175
33. Platt JC (1999) Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In: Adv. Large margin classifiers. MIT Press, pp 61–74
34. Qian Y, Yongxin Y, Yi-Zhe S, Xiang T, Hospedales TM (2015) Sketch-a-net that beats humans. In: Proceedings of the British machine vision conference 2015, (BMVC), pp 1–12
35. Radford A, Metz L, Chintala S (2016) Unsupervised representation learning with deep convolutional generative adversarial networks. In: ICLR
36. Razavian AS, Azizpour H, Sullivan J, Carlsson S (2014) CNN features off-the-shelf: an astounding baseline for recognition. In: Proceedings of the 2014 IEEE conference on computer vision and pattern recognition workshops (CVPRW '14). IEEE Computer Society, Washington, DC, pp 512–519
37. Salimans T, Goodfellow I, Zaremba W, Cheung V, Radford A, Chen X (2016) Improved techniques for training GANs. In: NIPS
38. Sangkloy P, Burnell N, Ham C, Hays J (2016) The sketchy database: learning to retrieve badly drawn bunnies. ACM Trans Graph 35(4):119:1–119:12
39. Sarvadevabhatla RK, Babu RV (2015) Freehand sketch recognition using deep features. arXiv:1502.00254
40. Schneider RG, Tuytelaars T (2014) Sketch classification and classification-driven analysis using fisher vectors. ACM Trans Graph 33(6):1–9
41. Seddati O, Dupont S, Mahmoudi S (2017) DeepSketch 3 analyzing deep neural networks features for better sketch recognition and sketch-based image retrieval. Multimed Tools Appl 76(21):22333–22359
42. Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. arXiv:1409.1556
43. Snoek CGM, Worring M, Smeulders AWM (2005) Early versus late fusion in semantic video analysis. In: Proceedings of the 13th annual ACM international conference on multimedia, pp 399–402
44. Srinivas S, Ravi Sarvadevabhatla K, Mopuri KR, Prabhu N, Kruthiventi S, Babu RV (2016) A taxonomy of deep convolutional neural nets for computer vision. Front Robot AI, 2(36)
45. Szegedy C, Liu W, Yangqing J, Sermanet P, Reed SE, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A (2015) Going deeper with convolutions. In: IEEE conference on computer vision and pattern recognition (CVPR), pp 1–9
46. Tseng KY, Lin YL, Chen YH, Hsu WH (2012) Sketch-based image retrieval on mobile devices using compact hash bits. In: Proceedings of the 20th ACM international conference on multimedia. ACM, pp 913–916
47. Wagh K, Thool R (2012) A comparative study of SOAP vs REST web services provisioning techniques for mobile host. J Inf Eng Appl 2(5):12–16. ISSN 2224-5782 (print), ISSN 2225-0506 (online)
48. Wang L, Sindagi V, Patel V (2018) High-quality facial photo-sketch synthesis using multi-adversarial networks. In: 13th IEEE international conference on automatic face & gesture recognition (FG 2018). Xi'an, pp 83–90
49. Wu S, Yang H, Zheng S et al (2017) Motion sketch based crowd video retrieval. Multimed Tools Appl 76(19):20167–20195
50. Xiao C, Wang C, Zhang L (2015) PPTLens: create digital objects with sketch images. ACM Conference on Multimedia
51. Yi Z, Zhang H, Tan P, Gong M (2017) DualGAN: unsupervised dual learning for image-to-image translation. In: 2017 IEEE international conference on computer vision (ICCV). Venice, pp 2868–2876
52. Yoo D, Park S, Lee J-Y, Kweon IS (2014) Fisher kernel for deep neural activations. arXiv:1412.1628

53. Zhou T, Krähenbühl P, Aubry M, Huang Q, Efros AA (2016) Learning dense correspondence via 3D-guided cycle consistency. In: 2016 IEEE conference on computer vision and pattern recognition (CVPR). Las Vegas, pp 117–126
54. Zhu J-Y, Krähenbühl P, Shechtman E, Efros AA (2016) Generative visual manipulation on the natural image manifold. In: ECCV
55. Zhu J-Y, Park T, Isola P, Efros AA (2017) Unpaired image-to-image translation using cycle-consistent adversarial networks. In: 2017 IEEE international conference on computer vision (ICCV), pp 2242–2251



Mustafa Sert received his MSc and PhD degrees in computer science from Gazi University, Ankara, Turkey, in 2001 and 2006, respectively. He has been an assistant professor of computer engineering at the Baskent University. He has research interests in theory and applications of audio signal processing, machine learning, pattern recognition, and multimedia databases. He mainly focuses on semantic content extraction from audio and video data, audio scene recognition, video concept detection, multimodality, and content modeling for multimedia search and retrieval. He serves in technical reviewing and organization committees of several international conferences including VLDB 2012, FUZZ-IEEE 2015, IEEE ISM (2008-2009, 2017), FQAS 2009, IEEE ICSC 2016-2017, IEEE IRC 2017, and IEEE BigMM 2016. He also serves as a reviewer of the IEEE&ACM TASLP, IEEE SPL, Springer MTAP, Springer SIVP, and IEEE TFS.

Dr. Sert is senior member of the IEEE and the IEEE Computer Society.

Emel Boyacı received the B.Sc. and M.Sc. degrees in computer engineering from Baskent University, Ankara, Turkey, in 2013 and 2017, respectively. She is currently working as a software developer at NOKIA Co., Turkey.