Deep Learning for Surface Material Classification Using Haptic and Visual Information

Haitian Zheng, Lu Fang, Mengqi Ji, Matti Strese, Yigitcan Özer, and Eckehard Steinbach, *Fellow, IEEE*

Abstract—When a user scratches a hand-held rigid tool across an object surface, an acceleration signal can be captured, which carries relevant information about the surface material properties. More importantly, such haptic acceleration signals can be used together with surface images to jointly recognize the surface material. In this paper, we present a novel deep learning method dealing with the surface material classification problem based on a fully convolutional network, which takes the aforementioned acceleration signal and a corresponding image of the surface texture as inputs. Compared to the existing surface material classification solutions which rely on a careful design of hand-crafted features, our method automatically extracts discriminative features utilizing advanced deep learning methodologies. Experiments performed on the TUM surface material database demonstrate that our method achieves state-of-the-art classification accuracy robustly and efficiently.

Index Terms—Convolutional neural network, haptic signal, hybrid inputs, surface material classification.

I. INTRODUCTION

WITH today's sensor technology, a wide variety of data types can be captured. Understanding the sensory data and recognizing objects is becoming an important research topic. Recent work in object recognition [1] and indoor scene recognition [2], for instance, demonstrate that combining visual information with depth sensory input improves the classification performance. Taking material surface classification as another example, besides camera sensors, the easily-accessible acceleration sensor is able to record vibration signals when it slides over a surface. Such vibration signals capture information about the material properties of the surface, and also reveal rich haptic attributes. These haptic signals are complementary to visual input, providing an opportunity towards a better material classification scheme. In this paper, we investigate such a multimodel material surface classification problem which involves both visual and haptic inputs.
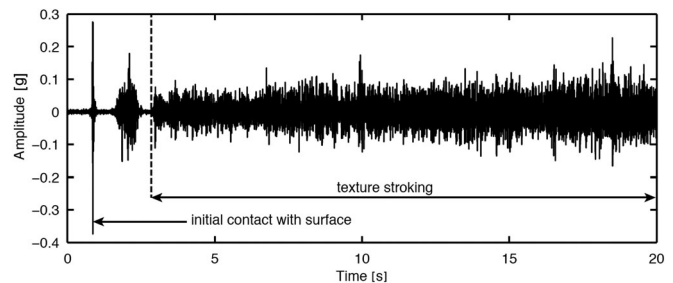
Fig. 1. Surface texture data trace, reproduced from [7]. It can be observed that higher velocities (left to right) increase the signal power and variance, while the exerted normal force is held constantly.

Surface material classification has recently gained increasing interest. When a rigid tool slides on a surface, the resulting vibrations of the tool contains information about characteristic properties of the surface [7], such as hardness and roughness of the material. There has been increasing interest to recognize surface materials using robots [7], [11], [13], [14] and to recreate the haptic feel of real surfaces. However, surface material classification from tool / surface interaction data becomes particularly challenging if free-hand movements are considered.

When a human strokes a rigid tool over an object surface, the exerted normal force, the tangential scan velocity, and the angle between the tool and the surface might vary during the surface exploration and between subsequent exploration sessions. These scan-time parameters strongly influence the nature of the recorded acceleration signals [9]. Fig. 1 shows an example of an acceleration data trace, where the scan velocity is linearly increasing and reveals, how this change influences the data trace with regard to its signal power and variance. The variability of the acceleration signals thus complicates the texture classification process.

Before surface classification using acceleration data (captured while interacting with the surface) emerged [7], [11]–[14], a significant number of previous works have focused on using photos of material surfaces to classify the material types. These approaches mainly rely on hand-crafted image features including locally binary pattern features [4], filter bank features [3], [6], co-occurrence matrix based features [5] etc., in combination with appropriate machine learning tools to distinguish the different texture types.

Designing specific features requires specific domain knowledge. Recently, convolutional neural networks (CNNs) have become a popular tool for pattern recognition, allowing better features being extracted automatically. In the context of surface classification, [22], [25] aim to classify texture image patches by training CNNs, [26] designs fisher-vector descriptors of texture

images using the ImageNet pretrained CNN, [27] uses CNN to learn texture image filter banks and [29] designs dynamic texture descriptors utilizing the ImageNet pretrained CNN. Meanwhile in the context of haptic classification, our previous work [11] proposes an auto-encoder pre-trained CNN for classifying texture haptic segments.[1] However:

1) Compare to the texture signal (image or haptic) which could be of arbitrary size, CNN has a relatively small receptive field. In order to reconcile the disagreement of the two mentioned sizes, an inefficient sliding window based approach needs to be adapted, which might jeopardize the efficiency of a real-time application.

2) Regardless of the significant progress in haptic (acceleration)-only or image-only classification, there are rarely approaches dealing with hybrid data input. For instance, two types of surface materials with similar image appearance can lead to completely different acceleration data, and vice versa. In such cases, better classification performance can be achieved by utilizing both haptic (acceleration) data and image data.

To overcome the inefficiency of *CNN + sliding window* scheme, a fully-convolutional network (FCN) approach is adapted in our work. FCN [19] is a special type of convolutional neural network which replaces fully connected layers with convolutional layers with a 1×1 convolution kernel. Without fully connected layers, the FCN is able to take input of arbitrary size, and outputs label predictions at every receptive field. More importantly, compared to the approach of *CNN + sliding window*, the FCN can be trained and tested more efficiently.

Different from previous approaches in adapting FCN for vision tasks, we propose a systematic FCN scheme for recognizing surface materials from hybrid data (haptic signals and images). The FCN for haptic data recognition is trained using concepts developed for speech recognition, as haptic data share similar characteristics with speech data [7], [12]; the FCN for image-based texture recognition is trained by fine-tuning the network weights from [16], inspired by transfer learning [40]. Afterwards, additional hybrid networks further integrate the haptic/visual features for better classification performances. Experiments conducted on publicly available surface material datasets [7], [23], [24] demonstrate the superior performance of our scheme in terms of both efficiency and accuracy for surface material classification. While the concurrent work [15] also applies deep learning for haptic/visual hybrid input-based surface classification, we would like to point out three important differences. First, we are eventually handling very different dataset from [15]. In [15], the material object is supposed to be at the center of a fixed-size image, and the haptic trace records the fixed-length signal when robot gripper explores specific procedures such as squeeze, hold, etc. While the TUM image/haptic-texture dataset contain repetitive patterns, and can be of arbitrary size. Second, given the fix-size dataset, [15] developed a one-shot CNN classification framework. To deal with the dataset

with repetitive patterns and arbitrary size, we develop a *FCN + max-voting* framework, where FCN improves the naive *CNN + sliding window* approaches by speed, and the max-voting further boosts the FCN prediction accuracy. Third, although [15] tested image/haptic fusion, they did not perform the joint training of fusion network. In this paper, with the help of the larger dataset, the joint training is possible and performed.

The remaining paper is structured as follows. In Section II, we discuss the related work and present CNN and FCN. In Section III, the details of our method are elaborated. Section IV describes the surface classification experiment on the TUM dataset and discusses the additional results. Finally, Section V concludes the paper.

II. RELATED WORK

CNNs serve as powerful tools for both image and speech recognition, which are a type of trainable multistage feed-forward artificial neural networks that extract a hierarchical feature representation. A typical CNN consists of convolution layers, pooling layers, and fully connected layers. The main ingredients of a CNN can briefly be summarized as follows:

1) Convolution layers extract feature maps from the input by applying consecutive convolution operations between the input and trainable kernels, followed by a non-linear activation function.

2) Pooling layers usually follow convolution layers, aiming to reduce both the dimensionality and translation sensitivity of the input feature maps. For image recognition, pooling layers significantly boost spatial translation invariance, while for spectrogram recognition pooling layers lead to temporal-frequency invariance.

3) Fully-connected layers are usually used at the ending stage of a CNN, providing more flexible feature mapping. In a fully-connected layer, the input feature vector is linearly converted into a new feature vector before being fed into a non-linear activation function.

4) Softmax layer is usually used for handling the multiple label regression problem. At the end of the neural network, the softmax layer outputs the normalized exponential of the input vector, which indicates the probability of each label.

The activation function of the CNN can be chosen among sigmoid function, tanh function, and rectified linear (ReLu) function [20], among which, ReLu becomes more and more popular due to its efficiency for training and effectiveness for improving the classification performance. Additionally, dropout regularization [21] is commonly used after fully-connected layers, which significantly reduces co-adaptation between features, and hence prevents over-fitting and boosts the classification performance significantly.

CNNs have experienced great success in a wide range of vision tasks, including image recognition [16], [17], detection [45], [46], segmentation [19] and many specific applications such as image aesthetics assessment [30] and eye fixations prediction [31]. In image recognition specifically, with the help of large datasets (i.e., ImageNet), CNN methods [16]–[18] have

---

[1][11], a preliminary version of this work, has appeared at MLSP 2015. It mainly focuses on directly dealing with the one-dimensional haptic signal without any preprocessing procedures. As an extension, this work dives into a more efficient solution with hybrid-input (haptic as well as image signals).

taken over the lead in large scale visual recognition challenges (ILSVRC) since 2012. For a small dataset, learning the millions of parameters of a CNN is usually impractical and may lead to over-fitting. CNNs, however, still successfully show their power – studies on transfer learning [40]–[42] show that the trained CNN model for one specific vision task usually learns a good representation of natural images, which works for other visual recognition tasks as well. Inspired by the success of transfer learning, we investigate in this paper how to handle the surface texture recognition task using the relatively small TUM image dataset [7].

CNN has also been shown to be a powerful tool for speech recognition. Recent works [32]–[36] show that CNNs notably outperform fully-connected deep neural network (DNN). The superior performance is attributed to the property of temporal-frequency translation invariance inherent with CNN [34]. In addition to speech recognition, CNNs are applied to acoustic recognition tasks such as music genre classification [38], music onset detection [39] and music adversaries [37]. Motivated by the previous work on one-dimensional (1-D) speech signal recognition, and evidence in [7] and [12] which show that the acceleration signals captured during the interaction with an object surface share certain characteristics with speech signals, we investigate a CNN method to deal with our haptic acceleration signal based recognition task.

Though CNNs provide powerful recognition abilities, additional designing is required to handle the special properties of the acceleration signals which describe the surface material. Specifically, the input can be of arbitrary size and contains frequently repeated local patterns. Thus it is more desirable to take local signal segments as the input, rather than the entire signal. Intuitively, the most simple texture prediction pipeline could be summarized as follows. First, convert the acceleration input into segments using sliding windows. Second, predict every segment with a trained CNN. Third, perform max-voting among multiple CNN predictions. Our experiments show that with a carefully trained CNN, the aforementioned work-flow achieves high prediction accuracy. However, a sliding window based approach is inappropriate for real-world applications, as dense prediction of the CNN is computationally expensive.

In our work, to alleviate this inefficiency, the naive sliding windows approach (step 1 and 2) is replaced by the FCN. FCN is a special type of convolutional neural network which replaces fully connected layers which output n features with convolutional layers with n 1×1 convolution kernels. The key observation of FCN is that fully-connected layers in a CNN are special convolutional layers: convolutional layers which take a 1×1 feature map and output another 1×1 feature map. In contrast to a fully-connected layer which takes fixed-length input merely, the corresponding convolutional layer is generalized to take input of arbitrary size, which is extremely beneficial for extracting dense features at every spatial location.

Taking this key observation into account, the FCN can replace the fully-connected layers of a CNN by convolutional layers with the same weights. Given the input data of arbitrary size, FCN first performs convolution and max-pooling operation alternatively as a usual CNN, then performs multiple

Fig. 2. Haptic stylus used for texture analysis.

1×1 convolution operations (and drop-out), finally provides label predictions at every receptive field. Compared to the sliding windows approach which heavily re-computes feature maps within overlapping "windows", FCN is significantly less computationally expensive. As further noted in [19], both training and inference of FCN can be performed by standard neural network approaches, leading to an efficient and systematic scheme.

III. PROPOSED CNN SCHEME WITH HYBRID INPUT

In this section, the proposed CNN scheme with hybrid input will be elaborated by starting with the explanation of hybrid data recording (Section III-A), followed by the surface texture classification schemes (Section III-B) via *HapticNet*, *VisualNet* and *FusionNet*.

A. Hybrid Data Recording

Haptic acceleration data: In our work, we use the haptic stylus from [12], which is a free-to-wield object with a stainless steel tool-tip, shown in Fig. 2. In [12], a three-axes LIS344ALH accelerometer (ST Electronics) with a range of $\pm 6\,g$ was applied to collect the raw acceleration data traces. All three axes were combined to one using DFT321 (see [10]). This approach, which preserves the spectral characteristics of the three axes, was adapted in order to have less computational effort in terms of feature calculation.

Image data: Different from acceleration modality, images are taken in a non-dynamic way of recording. However, differences in distance, rotation, light condition as well as focus also complicate the collection of uniform images for the surface classification task.

Fig. 3 shows all the used surfaces and their abbreviated names, where the images of the surfaces are taken by the rear camera of a common smartphone (Samsung S4 Mini) and have a resolution of 8 Mega-Pixels. The illumination and viewing conditions differ within the same class of surfaces. For each surface, there are 5 images under daylight and 5 images under ambient light conditions. Viewing direction and camera distance are chosen arbitrarily for each picture, resulting in variations within each
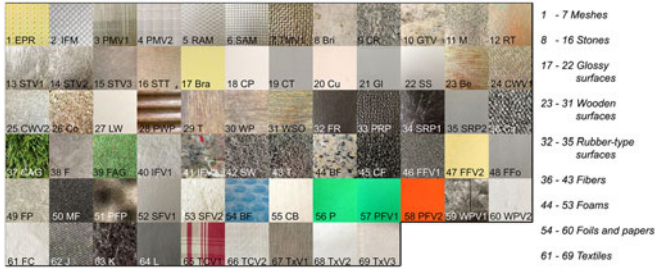
Fig. 3. Materials included in the TUM haptic texture database, freely accessible at http://www.lmt.ei.tum.de/texture/.
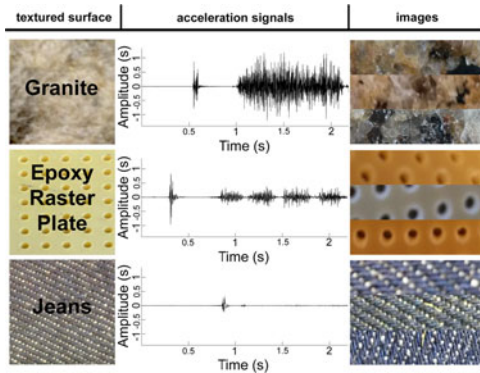

Fig. 4. Example signal traces of the used image and acceleration data recordings. Arbitrary variations during the recording of the acceleration data traces have been applied. Also, the 10 images per textured surface were captured under varying distances, light conditions, and focus conditions, as well as different camera inclinations towards the surface.

individual class. As an example, we choose three textured surfaces and plot the raw acceleration and image data in Fig. 4.

B. Surface Material Classification

Haptic trace/image preprocessing: As shown in Fig. 1, the haptic acceleration signal usually starts with a short initial impulse signal, when the rigid device initially touches the surface. The following data is a much longer movement signal when the device moves over the surface. As demonstrated by many speech recognition works [34]–[36], [38], converting a 1-D signal into the spectral domain is helpful for the CNN to achieve translation invariance in both temporal domain and frequency domain. Therefore, the 1-D raw acceleration data of the steady-state movement signal is transferred to its spectrogram.

A Hamming window in the time domain is used for enframing haptic signals, where the Hamming window length is set to 500, and the window shift is set to 100. At a sampling rate of 10 kHz, this is equivalent to 50 ms for the window size. Following [8], acceleration segments recorded during unconstrained exploration procedures generally stay stationary in such frame sizes. We select the first 50 low-frequency channels from the spectrogram, which preserved the most of energy from the haptic signal. Finally, the spectrogram is normalized such that the response in each channel has a minimum and maximum value of 0 and 1, respectively.

TABLE I
DETAILED STRUCTURE AND LAYER CONFIGURATIONS
OF THE PROPOSED *HapticNet*

layer type	patch size / stride	outputchannel size
convolution	3*3 / 1	50
pooling	2*2 / 2	50
normalization	–	50
convolution	3*3 / 1	100
pooling	2*2 / 2	100
convolution	3*3 / 1	150
pooling	2*2 / 2	150
convolution	3*3 / 1	200
pooling	2*2 / 2	200
convolution	4*12 / 1	400
dropout	–	400
convolution	1*1 / 1	250
dropout	–	250
convolution	1*1 / 1	69
softmax	–	69

Image input is resized to half-size for preprocessing. In this way, most of the texture patterns in an image can be preserved in the 224×224 receptive field window of AlexNet [16] (please refer to *VisualNet* for more detail on our visual classification approach).

HapticNet: Unlike previous work which applies CNN for haptic surface material recognition, we use a trained FCN to achieve dense prediction for the spectrogram input. FCN replaces the CNN's fully-connected layers by 1×1 convolutional layers. Without fully-connected layers, FCN produces dense prediction for the input with arbitrary length. Usually adjacent predictions share overlapping receptive fields and intermediate features. Since FCN gets rid of the computational redundancy for the overlapping intermediate features, compared to the approach of *CNN + sliding window*, FCN is much more efficient.

Our proposed FCN network for the haptic (acceleration) data, denoted as *HapticNet*, takes input of arbitrary length, and the output is a sequence of softmax-vectors of length L, $\{\mathbf{y_1}, \dots, \mathbf{y_L}\}$ representing the categorical probability distributions at different temporal locations [illustrated in Fig. 5(a)]. The detailed structure of *HapticNet* is shown in Table I. The *HapticNet* consists of three normal convolution max-pooling layers, and two 1×1 convolutional layers which replace the CNN's fully connected layers. The following layer of the final convolutional layer is the softmax layer, which gives categorical probability distributions at every temporal location. The detailed structure and layer configurations of the *HapticNet* are shown in Table I.

With the predicted output distribution vectors $\mathbf{y}_i$ at each temporal location i by the FCN, the corresponding class labels c_i are obtained by finding the label with maximum probability, i.e.,

$$c_i = \arg\max_c (\mathbf{y_i}[c]) \qquad (1)$$

where $\mathbf{y}_i[c]$ represents the c-th element of vector $\mathbf{y}_i$, and c enumerates all texture categories. In order to obtain the class label of the entire haptic signal C from multiple predictions $\{c_1, \dots, c_n\}$, a max-voting procedure that selects the class label
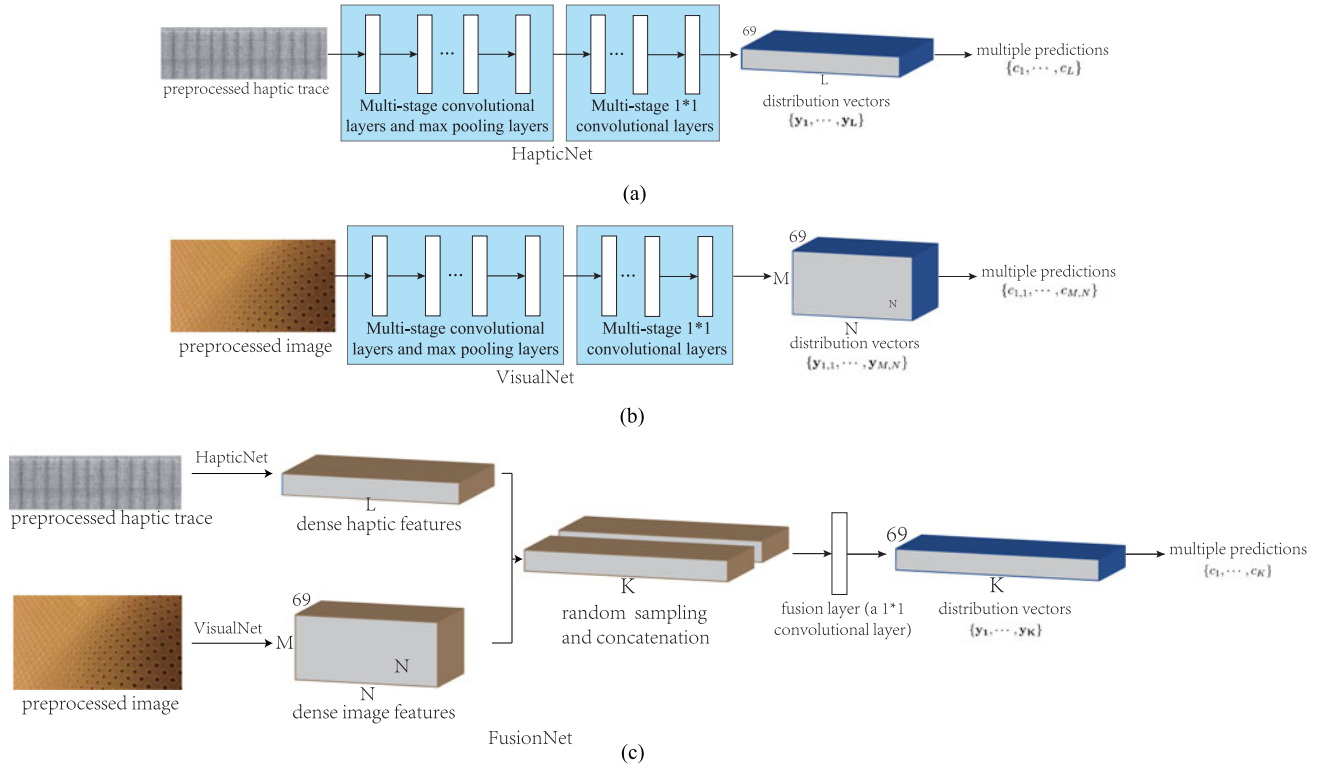
Fig. 5. Pipeline of the classification scheme. (a) Classification of haptic input (*HapticNet*); (b) classification of image input (*VisualNet*); (c) classification of hybrid input (*FusionNet*).

with the maximum vote is adapted

$$C = \arg\max_c \sum_i (\mathbb{1}(c_i = c)). \quad (2)$$

VisualNet: The classification pipeline for image data is similar to the one used for haptic data, i.e., an input image of arbitrary size is fed into a FCN network (denoted as *VisualNet*) for outputting categorical probability distributions of size $M \times N$, $\{\mathbf{y_{1,1}}, \ldots, \mathbf{y_{M,N}}\}$ at every spatial position. As illustrated in Fig. 5(b), the class labels $\{c_{1,1}, \ldots, c_{M,N}\}$ are obtained by applying (1) for every $\{\mathbf{y_{i,j}}\}$. Then, the class label of the entire image is obtained by applying (2) accordingly.

The structure of *VisualNet* is motivated by AlexNet which was proposed in [16]. In order to provide dense prediction, the last three fully connected layers are replaced by 1×1 convolutional layers (to make the representation simple, we denote the three 1×1 convolutional layers as FC_1, FC_2 and FC_3 respectively). In addition, the number of output features of FC_1, FC_2 and FC_3 are changed accordingly from $\{4096, 4096, 1000\}$ to $\{300, 250, 69\}$. The numbers of output features of FC_1 and FC_2 are reduced to prevent overfitting. We use the learned convolutional layers from AlexNet to initialize our corresponding top convolutional layers, and random weights are used to initialize the remaining 1×1 convolutional layers. The overall pipeline for image classification (*VisualNet*) is shown in Fig. 5(b), which predicts a distribution vector at every spatial location given the image input.

FusionNet: Suppose that two material surfaces A and B are similar in their visual appearance, but completely different with regard to their haptic perception. In such case, the haptic features will compensate the visual features for better indicating the surface type, and vice versa. Generally, in the scenario where both haptic and image data can be easily captured, a fusion framework can be proposed, where the hybrid input serves for complementing each of the individual information sources, providing richer features for the material surface classification task.

To fuse predictions using both haptic and image data, the *FusionNet* structure is proposed where image and haptic signals are fed to *VisualNet* and *HapticNet* respectively; then haptic/visual features are randomly sampled from *HapticNet/VisualNet*'s convolutional feature maps for K times; finally, the sampled haptic features and visual features are concatenated and fed into a 1×1 convolutional layer with 69 outputs for final predictions. Given multiple predictions like this, the max-voting is applied to obtain the final output class label. The structure of the proposed *FusionNet* is depicted in Fig. 5(c).

Training the *FusionNet* end-to-end is non-trivial, due to the described random sampling procedure which is not a common routine for standard DNNs. However, note that when input size of *FusionNet* is equal to the respective receptive field size of *HapticNet/VisualNet*, only one haptic/visual feature can be sampled from *FusionNet*. In such case, the random sampling procedure can be ignored, while *FusionNet* becomes a conventional fusion network. Exploiting such trait, *FusionNet* training can be much simplified. During testing, however, the described *FusionNet* structure still remains. In all, training of *FusionNet* can be simple, while testing of *FusionNet* can be as efficient as our proposed *HapticNet* and *VisualNet*.

TABLE II
HYPER-PARAMETERS FOR TRAINING *HapticNet*, *VisualNet*, AND *FusionNet*

	λ_0	γ	k	N	inputsize	batch-size
HapticNet	10^{-4}	0.3	4×10^4	10^5	$50 * 300$	10
VisualNet	3×10^{-5}	0.75	4×10^4	10^5	$384 * 384$	2
FusionNet	10^{-6}	0.1	4×10^6	10^5	$50 * 192$ & $224 * 224$	1

When training *FusionNet*, the *FusionNet* weights are initialized from the pretrained weights of *VisualNet* and *HapticNet*. Also during training, a much larger learning rate (10 times to 50 times) can be employed for the final 1×1 convolutional layer to speed up the training. More details about the training of *FusionNet* can be found in the experimental results section of the paper.

IV. EXPERIMENTAL RESULTS AND DISCUSSIONS

Dataset: The proposed *HapticNet*, *VisualNet* and *FusionNet* are evaluated using the TUM Haptic Texture dataset [7], which contains 69 texture classes and each class consists of 10 sampled haptic traces + 10 images, respectively. For the haptic data, these free-hand sampled traces vary in force and velocity. For the image data, it contains various samples under different lighting conditions.

The dataset is separated into training set and testing set using a ten-fold cross validation. In each fold, the training set contains 9 haptic traces and 9 images for each texture class, and the testing set contains the one remaining haptic data trace and image in each class. The training set is used to train the proposed network, and the testing set is used to evaluate the performance of the network and the following max-voting scheme. Before the ten-fold cross validation, a train/validation/test split is adapted for networks hyper-parameters tuning.

Training details: Our implementation is based on the Caffe deep learning framework [43], using a computer with 8GB RAM and Nvidia GPU GTX-860M. The training of *HapticNet*, *VisualNet* and *FusionNet* is conducted by applying Adam [44], an gradient-based stochastic optimization algorithm. For training each network, the learning rates are initialized to the base learning rate λ_0, then dropping by a constant factor γ every k iterations. Different choices of λ_0, γ and k, as well as the total training iterations N for each network is depicted in Table II. The $L2$ weight decay is set to 5×10^{-4}. Parameters of Adam are set to the default value proposed by [44], specifically $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-8}$.

During the testing stage of *HapticNet*, the preprocessing procedures described in Section III-B are first applied to the haptic tracks. The preprocessed data are then input to *HapticNet*. In the training stage, however, the haptic spectrograms are subsampled using a fixed length, then being fed into the network for training. By enforcing subsamples to be consistent in length when training *HapticNet*, we are able to increase the mini-batches size and thus speed up training. The size of the training subsample and mini-batch is depicted in Table II.

TABLE III
SURFACE CLASSIFICATION RESULTS USING
HAPTIC DATA AND/OR VISUAL DATA

Haptic Classification	Fragment	Max voting
MFCC + GMM [7]	–	80.23%
Modified MFCC Decreasing+ Naive Bayes [12]	–	89%
ACNN [11]	81.8%	–
HapticNet	**85.3%**	**91.0%**
Visual Classification	*Fragment*	*Max voting*
TCNN [27]	87.1%	–
VGG-M-FV [26]	73.7%	–
AlexNet-FV [26]	78.7%	–
VisualNet	85.6%	93.3%
VisualNet-TCNN	**87.1%**	**95.5%**
Hybrid Classification	*Fragment*	*Max voting*
FusionNet-FC$_3$	95.0%	98.1%
FusionNet-FC$_2$	96.2%	98.4%
FusionNet-FC$_2$-TCNN	96.6%	98.4%
FusionNet-FC$_3$-TCNN	**96.6%**	**98.8%**

TABLE IV
VisualNet CLASSIFICATION RESULTS KYLBERG TEXTURE
DATASET AND KTH-TIPS-2B DATASET

Kylberg Texture Dataset	Fragment	Max voting
VisualNet	**96.9%**	**97.8%**
VisualNet-TCNN	96.3%	96.9%
TCNN [27]	96.00%	–
D$_8$ [28]	82.0%	–
KTH-TIPS-2b Dataset	*Fragment*	*Max voting*
VisualNet	72.0%	72.1%
VisualNet-TCNN	**72.4%**	**72.4%**
TCNN [27]	72.36%	–
VGG-M-FV [26]	73.3%	–

Similarly, in the testing stage of *VisualNet*, preprocessed image is fed into the network for prediction. In the training stage, the preprocessed images are first subsampled into fixed-size patches, then fed into the network for training. Data augmentation using random rotation is also applied during *VisualNet* training. The size of the training subsample and minibatch is also depicted in Table II.

To train *FusionNet*, we restrict the size of the haptic and image input to be small enough, such that *HapticNet* will generate a single prediction for haptic input and *VisualNet* will only generate a single prediction for image input (as also described in *FusionNet* of Section III-B). The input size and mini-batch settings for *FusionNet* are depicted in Table II. During testing, we set the randomly sampled number K described in *FusionNet* of Section III-B to 1000. However, we do not observe other numbers to induce a significant influence on the classification performance.

HapticNet: The classification accuracy is shown in the *Haptic Classification* part of Table III. Specifically, we compare our results with several existing methods [7], [11], [12]. [7] proposes to combine MFCC features with a Gaussian mixture model (GMM) for movement phase recognition, which is denoted as 'MFCC + GMM' here. The following work [12] carefully discusses variant features for representing the movement signal, and variant discriminative models that are proposed for giving
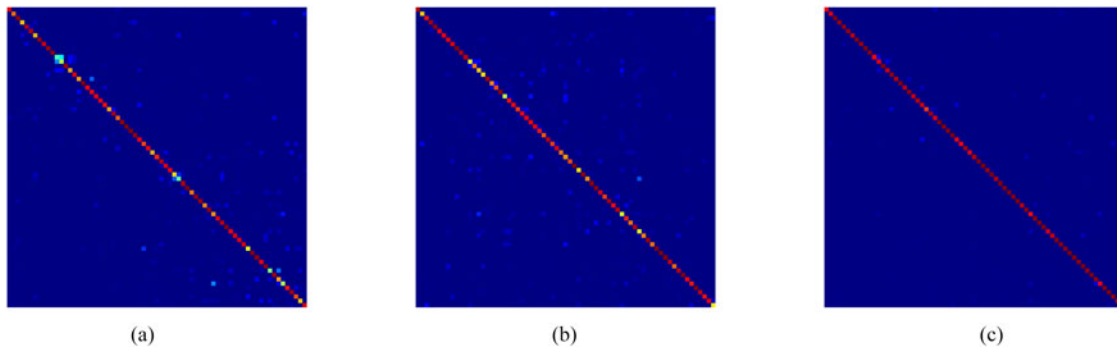
Fig. 6.   Fragment classification confusion matrix (10-fold averages) of (a) *HapticNet*, (b) *VisualNet*, and (c) *FusionNet-FC$_2$*.
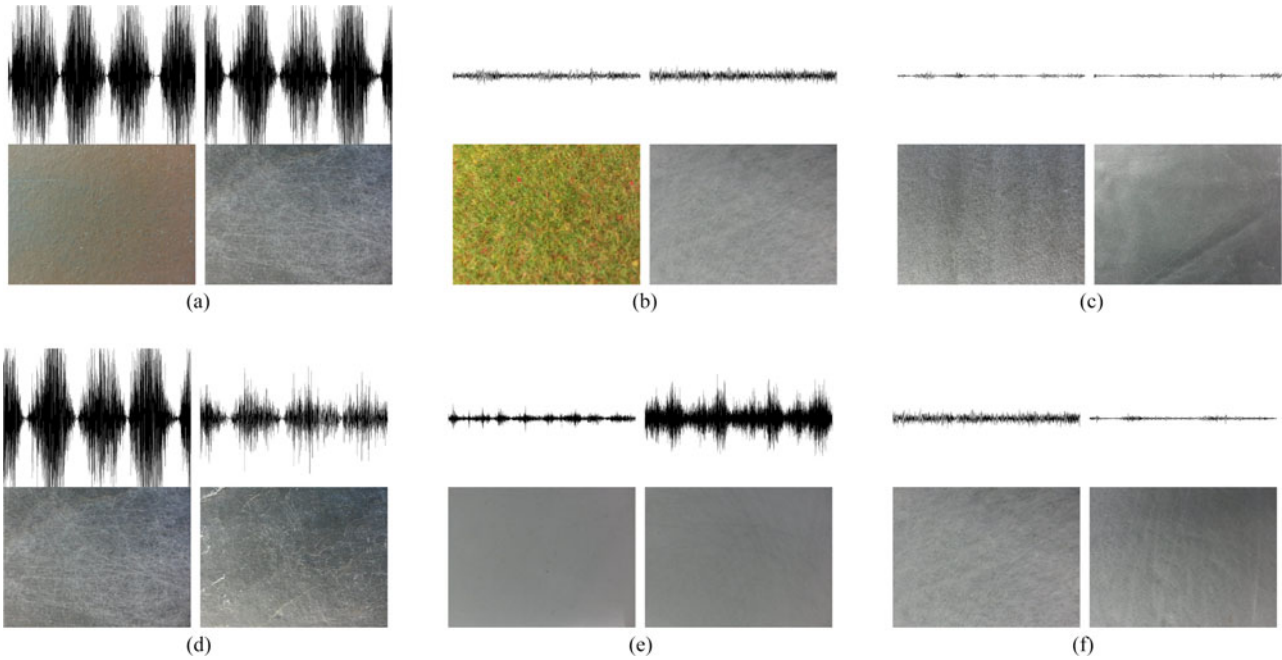

Fig. 7.   (a)–(c) show the material types that *HapticNet* cannot distinguish well, but *VisualNet* can distinguish, where (a) is Type 12 (RoofTile) VS 13 (StoneTileVersion1), (b) is Type 39 (FineArtificialGrass) VS 40 (IsolatingFoilVersion1), and (c) is Type 48 (FoamFoilVersion1) VS 64 (Leather). (d)–(f) depict the material types that *VisualNet* cannot distinguish well, but *HapticNet* can distinguish, where (d) is Type 13 (StoneTileVersion1) VS 14 (StoneTileVersion2), (e) is Type 18 (CeramicPlate) VS 19 (CeramicTile), and (f) is Type 40 (IsolatingFoilVersion1) VS 52 (StyroporVersion1).

predictions. Considering [7] uses "averaging feature" to test the prediction accuracy, which performs an underlying model averaging, the obtained results are listed in the max-voting column. [11] is the first work which uses CNN to classify the raw haptic data. By comparison, *HapticNet* achieves superior performance with the fragment accuracy be 85.3% and max-voting accuracy be 91.0%, using the ten-fold cross validation measurement.

VisualNet: The classification accuracy of the concerned scheme is shown in the *Visual Classification* part of Table III. Specifically, we compare our approach *VisualNet* with the T-CNN-3 (denoted as TCNN) in [27] and the descriptor (denoted as *VGG-M-FV*) in [26], as well as the descriptor by replacing VGG-M network with AlexNet (denoted as *AlexNet-FV*). We observe that *VisualNet* achieves comparable fragment accuracy as *TCNN* [27], and higher accuracy than *VGG-M-FV* [26] and *AlexNet-FV* [26]. To demonstrate that the *VisualNet* framework is easily adaptable for other neural network design, we adapt the *TCNN* into a fully convolutional version, which is denoted

as *VisualNet-TCNN*. *VisualNet-TCNN* is achieved by reserving the first three convolutional layers of *VisualNet*, followed by an average pooling layer (which is similar to *TCNN*) and three 1×1 convolutional layers. *VisualNet-TCNN* achieves highest accuracy for classifying images of surface materials.

Additionally, we test *VisualNet* and *VisualNet-TCNN* on other texture image datasets, including Kylberg Texture Dataset [23] and KTH-TIPS-2b [24]. Our experimental results are reported in Table IV. Note that the train/test split of Kylberg Texture Dataset and KTH-TIPS-2b follows the paper [27]. On Kylberg Texture Dataset, the T-CNN-3 (denoted as TCNN) in [27] and the D_8 descriptor [28] are compared with our approach. From comparison, *VisualNet* achieves the best performance, with very similar yet slightly higher accuracy than *VisualNet-TCNN*. It is consistent with the conclusion in [27], which suggests AlexNet slightly outperforms TCNN on Kylberg dataset. On KTH-TIPS-2b dataset, TCNN and the VGG-M-FV descriptor [26] are compared with our approach. From the comparison we can conclude
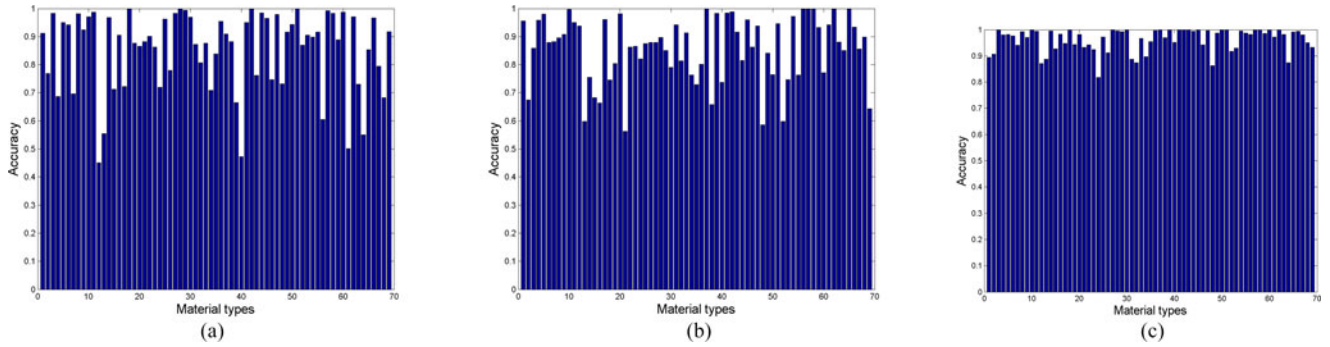
Fig. 8. Fragment classification histogram (10-fold averages) of (a) *HapticNet*, (b) *VisualNet*, and (c) *FusionNet-FC$_2$*.

that *VisualNet* achieves competitive performance. We notice that max-voting obtains only a slight performance gain for the fragment accuracy. This is due to testing image in KTH-TIPS-2b dataset is resized to 256×256 – very small size for the receptive field sizes of *VisualNet* and *VisualNet-TCNN*.

FusionNet: For the experiments of classifying hybrid (haptic+visual) input, we first denote the last three 1×1 convolutional layers of *HapticNet/VisualNet* as FC$_1$, FC$_2$ and FC$_3$ respectively. The *FusionNet* is tested with two different settings: haptic features and visual features from either FC$_3$ layers or FC$_2$ layers are fused (the resulting network are denoted as *FusionNet-FC$_3$* and *FusionNet-FC$_2$* respectively in the *Fusion Classification* part of Table III). As we expected, the accuracy of haptic predictions and image predictions is significantly boosted with the fusion framework. By replacing the visual submodule in *FusionNet* with *VisualNet-TCNN*, we have *FusionNet-FC$_3$-TCNN* and *FusionNet-FC$_2$-TCNN*, which achieve comparable and highest accuracy for classifying surface material.

Experiment Analysis: To better understand the performance of our proposed scheme, we show the fragment classification confusion matrices of *HapticNet*, *VisualNet* and *FusionNet-FC$_2$* in Fig. 6, and the fragment classification accuracy is depicted in Fig. 8 as well. In the confusion matrices in Fig. 6, values between 0 to 1 are represented by colors varying from blue to red. Each row of the confusion matrices represents the probability of a material type being classified into the different 69 classes. Examining Fig. 6(a) and 6(b), we have following observations:

1) The off-diagonal misclassification patterns follow different distributions, implying that *HapticNet* and *VisualNet* have quite different behaviors when classifying surface materials.

2) There are some material types which *HapticNet* usually cannot distinguish well. For example: type 12 (RoofTile) and type 13 (StoneTileVersion1), type 39 (FineArtificialGrass) and type 40 (IsolatingFoilVersion1), type 48 (FoamFoilVersion1) and type 64 (Leather). However, they are more distinguishable using *VisualNet*. Haptic/image samples of these materials are shown in Fig. 7(a)-(c).

3) Similarly, there are samples which *VisualNet* cannot distinguish but *HapticNet* can distinguish well. For example: type 13 (StoneTileVersion1) and type 14 (StoneTileVersion2), type 18 (CeramicPlate) and type 19 (CeramicTile), type 40 (IsolatingFoilVersion1) and type

TABLE V
PROFILING COMPARISON BETWEEN THE PROPOSED *HapticNet/VisualNet* AND *CNN + Sliding Window* APPROACHES

Haptic classification	Average running time
Haptic CNN + sliding window	195.4 ms
HapticNet	**20.5 ms**
Visual classification	*Average running time*
Visual CNN + sliding window	7903.4 ms
VisualNet	**154.3 ms**

52 (StyroporVersion1). Haptic/image samples of these materials are shown in Fig. 7(d)–(f).

Fig. 6(c) shows that the above mentioned samples which were confused when working with a single modality, are mostly correctly classified for hybrid input. The diagonal line is more close to 1 compared to Fig. 6(a) and (b). Different from Fig. 6(a) and (b), fewer off-diagonal values exist. *FusionNet-FC$_2$* hence efficiently removes most misclassifications. The fragment classification accuracies of every material type in Fig. 8 also demonstrate the effectiveness of haptic/visual fusion, as revealed by the fact that *FusionNet-FC$_2$* achieves stably high accuracy (vertical axis) when classifying all the 69 material types (horizontal axis).

Time profiling: To demonstrate the efficiency of our algorithm, we further compare the time profiling of *HapticNet/VisualNet* with two related *CNN + sliding window* approaches. For the sliding window approaches, we transfer the weight of *HapticNet/VisualNet* to two CNNs which share the same layer settings with *HapticNet/VisualNet*, while the 1×1 convolutional layer is replaced by a fully-connected layer. Clearly, with exactly the same weight, the sliding window approaches achieve the same fragment/max-voting accuracy as *HapticNet* and *VisualNet*. However, because FCN is able to avoid the redundant feature computation that exists in *CNN + sliding window* approaches, it should be faster. To validate that FCN based approaches like *HapticNet/VisualNet* are faster, we profile the average running time of *HapticNet/VisualNet* in comparison to *CNN + sliding window* approaches on a Nvidia GTX 860M graphics card. As shown in Table V, for haptic classification, the sliding window-based approach takes 195.4 ms to run, while *HapticNet* only takes 20.5 ms. Moreover, for visual

classification, the sliding window approach takes 7903.4 ms, while *VisualNet* only takes 154.3 ms.

V. CONCLUSION AND FUTURE WORK

We introduce a surface material classification method which uses Fully CNNs. For predicting individual haptic or image input, we apply FCN with max-voting framework. We then design a fusion network dealing with both haptic and image input. Experiments on the TUM Haptic Texture Database demonstrate that our proposed system can achieve competitive classification accuracy compared to the existing schemes at reduced complexity. In our future work, we are aiming at further extending the current *FCN + max-voting* and hybrid classification schemes to more input types (images, haptic acceleration signals, sound signals, further signals from other modalities) for further improving the flexibility and robustness of our system.

REFERENCES

[1] A. Wang, J. Lu, J. Cai, T. Cham, and G. Wang, "Large-margin multi-modal deep learning for RGB-D object recognition," *IEEE Trans. Multimedia*, vol. 17, no. 11, pp. 1887–1898, Nov. 2015.

[2] J. Tang, L. Jin, Z. Li, and S. Gao, "RGB-D object recognition via incorporating latent data structure and prior knowledge," *IEEE Trans. Multimedia*, vol. 17, no. 11, pp. 1899–1908, Nov. 2015.

[3] L. Liu and P. Fieguth, "Texture classification from random features," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 3, pp. 574–586, Mar. 2012.

[4] T. Ojala, P. Matti, and M. Topi, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 7, pp. 971–987, Jul. 2002.

[5] V. Arvis, C. Debain, M. Berducat, and A. Benassi, "Generalization of the cooccurrence matrix for colour images: Application to colour texture classification," *Image Anal. Stereol.*, vol. 23, no. 1, pp. 63–72, 2011.

[6] M. Varma and A. Zisserman, "Texture classification: Are filter banks necessary," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2003, vol. 2 pp. II-691–II-698.

[7] M. Strese *et al.*, "A haptic texture database for tool-mediated texture recognition and classification," in *Proc. IEEE Int. Symp. Haptic Audio Visual Environ. Games*, Oct. 2014, pp. 118–123.

[8] H. Culbertson, J. Unwin, B. E. Goodman, and K. J. Kuchenbecker, "Generating haptic texture models from unconstrained tool-surface interactions," in *Proc. World Haptics Conf.*, 2013, pp. 295–300.

[9] J. M. Romano and K. J. Kuchenbecker, "Creating realistic virtual textures from contact acceleration data," *IEEE Trans. Haptics*, vol. 5, no. 2, pp. 109–119, Apr.-Jun. 2012.

[10] N. Landin, J. M. Romano, W. McMahan, and K. J. Kuchenbecker, "Dimensional reduction of high-frequency accelerations for haptic rendering," in *Haptics: Generating and Perceiving Tangible Sensations*. Berlin, Germany: Springer-Verlag, 2010, pp. 79–86.

[11] M. Ji, L. Fang, H. Zheng, M. Strese, and E. Steinbach, "Preprocessing-free surface material classification using convolutional neural networks pretrained by sparse autoencoder (ACNN)," in *Proc. Mach. Learn. Signal Process.*, Sep. 2015, pp. 1–6.

[12] M. Strese, C. Schuwerk, and E. Steinbach, "Surface classification using acceleration signals recorded during human free hand movement," in *Proc. IEEE World Haptics Conf.*, Jun. 2015, pp. 214–219.

[13] J. M. Romano and K. J. Kuchenbecker, "Methods for robotic tool-mediated haptic surface recognition," in *Proc. IEEE Haptics Symp.*, Feb. 2014, pp. 49–56.

[14] J. A. Fishel and G. E. Loeb, "Bayesian exploration for intelligent identification of textures," *Frontiers Neurorobot.*, vol. 6, no. 4, pp. 1–20, Jun. 2012.

[15] Y. Gao, L. A. Hendricks, K. J. Kuchenbecker, and T. Darrell, "Deep learning for tactile understanding from visual and haptic data," *CoRR*, 2015. [Online]. Available: http://arXiv.org/abs/1511.06065

[16] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[17] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, 2014. [Online]. Available: http://arXiv.org/abs/1409.1556

[18] C. Szegedy *et al.*, "Going deeper with convolutions," *CoRR*, 2014. [Online]. Available: http://arXiv.org/abs1409.4842

[19] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2015, pp. 3431–3340.

[20] V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *Proc. 27th Int. Conf. Mach. Learn.*, 2010, pp. 807–814.

[21] S. Wager, S. Wang, and P. Liang, "Dropout training as adaptive regularization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2013, pp. 351–359.

[22] F. H. C. Tivive and A. Bouzerdoum, "Texture classification using convolutional neural network," in *Proc. IEEE Region 10 Conf.*, Nov. 2006, pp. 1–4.

[23] G. Kylberg, "Kylberg Texture Dataset v. 1.0," *Centre Image Anal.*, Swedish Univ. Agricultural Sci. and Uppsala Univ., External Rep. (Blue Ser.) 35, 2011.

[24] E. Hayman, B. Caputo, M. Fritz, and J. O. Eklundh, "On the significance of real-world conditions for material classification," in *Proc. 8th Eur. Conf. Comput. Vis.*, 2004, pp. 253–266.

[25] L. G. Hafemann, "An analysis of deep neural networks for texture classification," M.Sc. thesis, Federal Univ. Paraná, Curitiba, Brazil, 2014. [Online]. Available: http://dspace.c3sl.ufpr.br:8080/dspace/handle/1884/36967

[26] M. Cimpoi, S. Maji, and A. Vedaldi, "Deep filter banks for texture recognition and segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2015, pp. 3828–3836.

[27] V. Andrearczyk and P. F. Whelan, "Using filter banks in convolutional neural networks for texture classification," *CoRR*, 2016. [Online]. Available: http://arXiv.org/abs/1601.02919

[28] N. Liu, G. Gimel'farb, and P. Delmas, "High-order MGRF models for contrast/offset invariant texture retrieval," in *Proc. 29th Int. Conf. Image Vis. Comput.*, 2014, pp. 96–101.

[29] X. Qi, C. G. Li, G. Zhao, X. Hong, and M. Pietikinen, "Dynamic texture and scene classification by transferring deep image features," *Neurocomputing*, vol. 171, pp. 1230–1241, 2016.

[30] X. Lu, Z. Lin, H. Jin, J. Yang, and J. Z. Wang, "Rating image aesthetics using deep learning," *IEEE Trans. Multimedia*, vol. 17, no. 11, pp. 2021–2034, Nov. 2015.

[31] C. Shen, X. Huang, and Q. Zhao, "Predicting eye fixations on webpage with an ensemble of early features and high-level representations from deep network," *IEEE Trans. Multimedia*, vol. 17, no. 11, pp. 2084–2093, Nov. 2015.

[32] O. Abdel-Hamid *et al.*, "Convolutional neural networks for speech recognition," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 22, no. 10, pp. 1533–1545, Oct. 2014.

[33] O. Abdel-Hamid, A. Mohamed, J. Hui, and G. Penn, "Applying convolutional neural networks concepts to hybrid NN-HMM model for speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Mar. 2012, pp. 4277–4280.

[34] T. N. Sainath, A. Mohamed, B. Kingsbury, and B. Ramabhadran, "Deep convolutional neural networks for LVCSR," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, May. 2013, pp. 8614–8618.

[35] P. Swietojanski, A. Ghoshal, and S. Renals, "Convolutional neural networks for distant speech recognition," in *Proc. 4th Joint Workshop Hands-Free Speech Commun. Microphone Arrays*, 2014, pp. 172–176.

[36] O. Abdel-Hamid, L. Deng, and D. Yu, "Exploring convolutional neural network structures and optimization techniques for speech recognition," in *Proc. INTERSPEECH*, 2013, pp. 3366–3370.

[37] C. Kereliuk, B. L. Sturm, and J. Larsen, "Deep learning and music adversaries," *IEEE Trans. Multimedia*, vol. 17, no. 11, pp. 2059–2071, Nov. 2015.

[38] Q. Kong, X. Feng, and Y. Li, "Music genre classification using convolutional neural network," in *Proc. Int. Soc. Music Inform. Retrieval (ISMIR)*, 2014.

[39] J. Schluter and S. Bock, "Improved musical onset detection with convolutional neural networks," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, May 2014, pp. 6979–6983.

[40] M. Oquab, B. Leon, L. Ivan, and S. Sivic, "Learning and transferring mid-level image representations using convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2014, pp. 1717–1724.

[41] J. Donahue *et al.*, "DeCAF: A deep convolutional activation feature for generic visual recognition," *CoRR*, 2013. [Online]. Available: http://arxiv.org/abs/1310.1531

[42] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 818–833.

[43] Y. Jia *et al.*, "Caffe: Convolutional architecture for fast feature embedding," in *Proc. ACM Int. Conf. Multimedia*, 2014, pp. 675–678.

[44] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, 2015. [Online]. Available: http://arxiv.org/abs/1412.6980

[45] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2014, pp. 580–587.

[46] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Conf. Adv. Neural Inf. Process. Syst.*, 2015, pp. 91–99.

Matti Strese received the graduate and M.Sc. degrees in electrical engineering from the Technische Universität München, Munich, Germany, in 2012 and 2014, repectively.

In September 2014, he joined the Media Technology Group, Technische Universität München, where he is currently a member of the research staff. His research interests include the field of analysis and synthesis of haptic texture signals and haptic devices.

Haitian Zheng received the B.Sc. and the M.Sc. degrees in electronics engineering and informatics science from the University of Science and Technology of China, Hefei, China, under the supervision of Prof. L. Fang, in 2012 and 2016, respectively.

His research interests include computer vision and machine learning.

Yigitcan Özer received the B.S. degree in electrical and electronics engineering from Bilkent University, Ankara, Turkey, in 2013, and the M.Sc. degree in communications engineering from the Technical University of Munich, Munich, Germany, in 2015.

In February 2016, he joined Cape Analytics, Inc., Munich, Germany. His research interests include deep learning, computer vision, and geographic recognition systems.

Lu Fang received the Ph.D. degree from the ECE Department, Hong Kong University of Science and Technology (HKUST), Hong Kong, China, in 2011.

In 2012, she was an Associate Professor with the University of Science and Technology of China, Hefei, China. She is currently visiting the HKUST Robotics Institute. Her research interests include image/video processing and vision for intelligent robot.

Prof. Fang was the recipient of the Humboldt Research Fellowship for Experienced Researchers in 2014.

Eckehard Steinbach (M'96–SM'08–F'15) studied electrical engineering at the University of Karlsruhe, Karlsruhe, Germany, the University of Essex, Colchester, U.K., and ESIEE, Paris, France.

From 1994 to 2000, he was a Member of the research staff of the Image Communication Group, University of Erlangen-Nuremberg, Erlangen, Germany, where he received the Engineering Doctorate in 1999. From February 2000 to December 2001, he was a Postdoctoral Fellow with the Information Systems Laboratory, Stanford University, Stanford, CA, USA. In February 2002, he joined the Department of Electrical and Computer Engineering, Technical University of Munich, Munich, Germany, where he is currently a Full Professor for media technology. His research interests include area of haptic and visual communication, teleoperation over the Tactile Internet, and indoor mapping and localization.

Mengqi Ji received the B.S. degree from the University of Science and Technology Beijing, Beijing, China, in 2012, and the M.Sc. degree from the Hong Kong University of Science and Technology, Hong Kong, China, in 2013, where he is currently working toward the Ph.D. degree.

He was with the University of Limerick, Limerick, Ireland, in 2012, and with the Technical University of Munich, Munich, Germany, in 2015. His research interests include multimedia signal processing and deep learning.