修　士　論　文

Recognizing Multi-scale Material Traits

(スケールの異なる画像からのマテリアル属性の認識)

東京大学大学院
情報理工学系研究科
電子情報学専攻

48-136450　張　璐霖

指導教員　佐藤 洋一　教授

平成27年2月

Abstract

Material traits (e.g. smooth, transparent, shiny) provide abundant perceptual information of materials. Comparing with some low-level features, recognizing materials with its perceptual traits is much more similar to the function of human beings' brains. Learning material traits, which have been proved discriminative and recognizable, provides a new perspective for materials recognition in computer vision.

In latest research of Schwartz *et al.*, a per-pixel algorithm was proposed to recognize material traits, with Flickr Material Database built by Sharan *et al.*. Experiment results showed that their approach was effective and object-independent for material traits recognition in arbitrary images. However, scale change of both training and testing data had not been taken into consideration in their work.

In this thesis, we propose an approach to recognize material traits from images with different scales. We train hierarchical convolutional features of material traits at different scales, with an unsupervised auto-encoder algorithm. Then multi-scale randomized decision forest classifiers are trained on these features to directly recognize material traits. We also proposed two methods to select appropriate scale effectively in a recognition task. Our results present an obvious improvement on recognition accuracy in a multi-scale recognition task comparing with the single-scale approach.

Acknowledgements

First and foremost, I have to express my sincere gratitude to my advisor, Prof. Yoichi Sato, for offering me the opportunity to study in the University of Tokyo, for his continuous and patient support of my research and my life in Japan. Prof. Sato is not only an excellent advisor, but also an outstanding researcher. He always shows a strong desire for new knowledge, which impresses us deeply when we see our professor asking numerous questions to learn from his students with interest in our lab meetings.

I also would like to express thankfulness to every member in Sato Laboratory for their cooperation and assistance in my research, especially to Prof. Imari Sato and Prof. Takahiro Okabe and Dr. Feng Lu who are always there to give many constructive suggestions when I face problems in my research. Without their assistance and encouragement in every step throughout the process, this thesis would have never been accomplished. I also would like to thank the secretaries, Sakie Suzuki, Yoko Imagawa, Chio Usui for their continuous help.

I would like to give credits to the Bank of Communications, Tokyo branch, where I did a long-term internship for more than one year. It gives me not only a financial support, but also an opportunity to practise Japanese, to improve IT skills, and to expand my knowledge of economy and finance. Especially, I would like to express gratefulness to my department head, Mr. Jianbin Situ, from whom I learn a lot.

I also would like to thank all my dear friends, especially Guan Yang, Yaxi Lu, Kumiko Oyama, Peng Wang and Fei Feng. I can not imagine my life in Japan without them.

Most importantly, I would like to express my gratitude to my parents, Yongping Zhang and Yuexing Chen. Without their support, I might not be able to do anything. This month, they will spend their third Chinese New Year without me at home . Although they do not complain one word about this, as the only child in my family, I can still feel their loneliness. I am sorry about this, and I would like to say to my dear parents that I love you.

February 2015

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Object recognition is a topic which is never out of season in computer vision. Since the first day we human beings invented a robot, we had never stopped to teach these machines how to recognize an object. Unfortunately, until now, our robots can not be as smart as real human being to recognize any objects around us. Figure 1.1 shows that even if these three objects show similar texture patterns, they are made of different materials, and obviously they are completely different objects. So in order to recognize an object automatically and precisely, it is necessary to discuss the problem that how to recognize different materials effectively.

Actually, recognizing materials (e.g. metal, ceramic, fabric, plastic, wood) is an meaningful task in both scientific research and industrial application. This technology is not only widely utilized in industrial product inspection, mineralogy, remote sensing, but also in our everyday life such as food inspection and resource recycling. Many features contained in materials, such as color, Bidirectional Reflectance Distribution Function (BRDF), texture, translucency, and polarization, provide us various solutions to solve material recognition problems.

Although some conventional approaches can obtain a high accuracy when implemented in a laboratory environment or for a well-prepared raw material, recognition for an everyday object in an arbitrary photo is still a challenging but meaningful task for us.

Figure 1.1: Object with similar texture patterns can be made of different materials (from left to right): fabric, plastic, and paper [1].

In conventional approaches, material categories are often recognized directly from low-level features, which depends too much on raw material properties. Such *In-Lab* approach may cause a consequence that well-prepared raw materials can be recognized at a very high accuracy rate in a lab environment, while it may not work when recognizing a painted, dyed material in an everyday life photo. In addition, complicated and expensive experimental devices are not acceptable in a everyday application. Obviously, people should like to recognize or classify something with a portable device like our smart phone, instead of doing this in a laboratory or a factory.

Recently, Fleming *et al.* [6] proposed a method to predict the material's perceptual properties(e.g. hardness, rigidity, colorfulness), and proved that these qualities are systematically related to their corresponding material classes. After them, Schwartz *et al.*[7] proposed a more systematically generalized approach to recognize per-pixel material traits(e.g. smooth, shiny, liquid, organic).

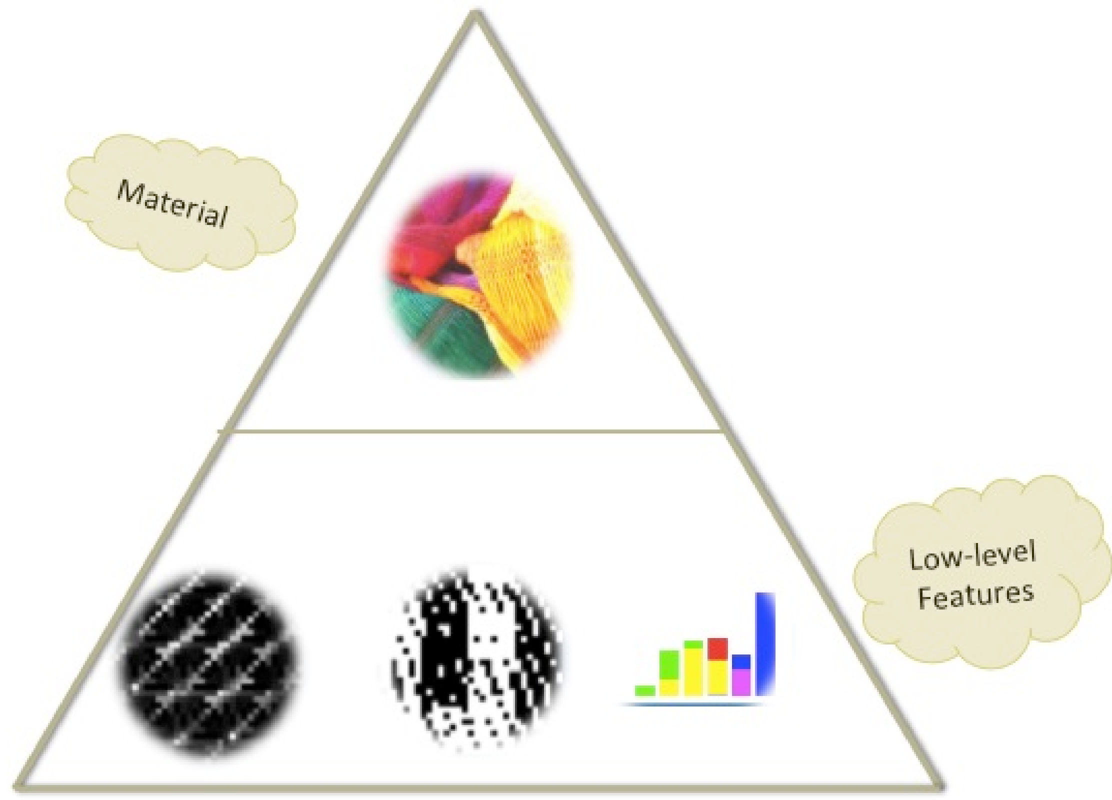
No matter what accuracy they will obtain, we can find that this approach based on materials' perceptual properties is not only a great improvement, but also a trend in material recognition. By employing perceptual properties, we will force our computer vision system to think more likely to human's brain and neural network. In an other word, we are trying to teach our computers how to distinguish a material in a perceptual layer(Fig. 1.2(b)), instead of recognizing materials with some basis visual information such as color, texture(Fig. 1.2(a)). When we take the first glance at Fig.1.1, maybe some of us will also be confused for a few seconds, as same as our robots. However, after these few seconds, all the knowledge and memories stored in our brain will be activated in a short time, and then we can soon make a precise judge of their corresponding materials. In this process,usually we make our final decision by analysing material traits, such as smooth, shiny, soft, instead of some fundamental features.

Therefore, recognizing material traits, which encode discriminative material information is a good solution to do material classification in photos of everyday life objects. In addition, a per-pixel recognizing approach may avoid dependence on information of material's object-specific features, making the system still effective for other test samples.However, in the paper of Schwartz *et al.*, they do not consider the case of handling material recognition with different image scales.

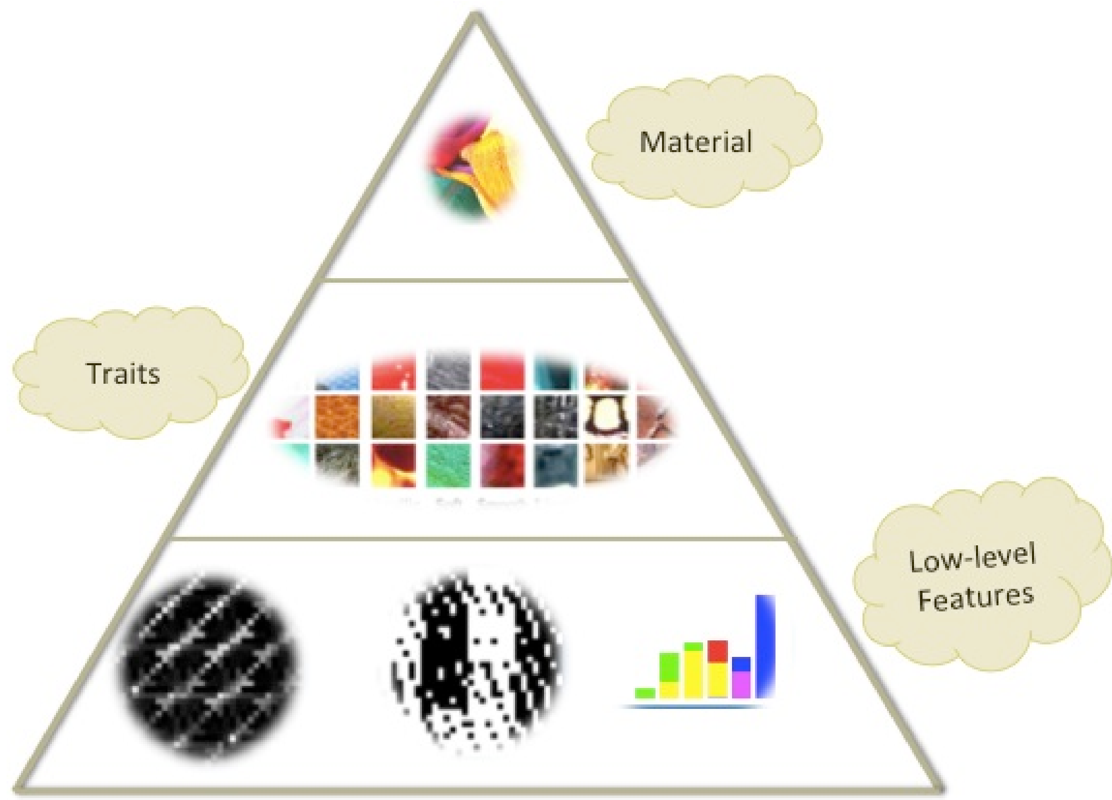
As Fig. 1.3 shows, even though patches are extracted with the same size from a photo of a certain material, they may contain completely different appearance of the same material, because the object may show different scales in one photo.

We consider that recognition results predicted by a *wrong scale* classifier conceivably might be erroneous. Therefore, it must be an interesting and meaningful topic to research recognition of material traits with different scales.

In this thesis, we propose an approach to recognize multi-scale material traits. We firstly train Convolutional Auto-encoder (CAE) Filters, which can represent the features well under an unsupervised learning, for material image patches with S scales. These filters will later been proven to be very important in our classifiers.



(a) Recognize materials by low-level features



(b) Recognize materials by perceptual traits

Figure 1.2: A perceptual layer in material recognition

Figure 1.3: Image patches of the same size may contain material information with different scale

We also prepare annotated material trait patches, including patches with a certain material trait(e.g. smooth, metallic, organic) present or not present. For each trait, we take balanced sets of positive and negative examples for our classifier training. Then, we take pooled convolutional responses of prepared positive and negative trait patches by trained CAE filters of each scale, also together with some other non-linear features (e.g. HOG, LBP, Color Histogram), as a huge Feature Matrix. We utilize feature matrices obtained from CAE filters of N scale, to train S Random Decision Forest Classifiers, which can directly recognize material traits with different scales. After that we also propose two effective approaches of scale selection for these classifiers to make a final decision, by searching maximum confidence among the predictions given by multi-scale classifiers, or computing the average prediction probability.

The remainder of this thesis is organized as follows:

In Chapter 2, we introduce some related works in material recognition firstly. Then we will introduce the material database employed in our experiment.

In Chapter 3, we will firstly bring the concept of material traits and talk about our goals in this research. Then our algorithms of multi-scale material trait recognition will be introduced. Finally, we will talk about some strategies for selecting the best scale of input samples.

In Chapter 4, procedures and settings of our experiments will be introduced. Particularly, we would like to introduce how we train convolutional auto-encoder filters and randomized decision forest classifiers with different scales. Our testing results will be given to establish that our proposed approach does contribute to an improvement of accuracy in a multi-scale recognition task.

In Chapter 5, we make a summary of our work and talk briefly about the future direction of this research.

Chapter 2

Related Works

2.1 Material Recognition

Material recognition is a challenging but meaningful topic in computer vision, in [8], Adelson *et al.*firstly separate material recognition from object recognition and illustrate importance of material recognition in computer vision. Until now, many algorithm have been proposed for material recognition, including per-pixel and object-dependent methods. Although per-pixel recognition without object's non-local features such as edge contours would be more error-prone, which is proved in [1], it is still the ideal approach for material recognition. In our everyday life environment, many objects are not made of only one material (Fig. 2.1), which makes per-pixel recognition a meaningful work to do. Wang *et al.* [2] proposed an approach to utilise Bidirectional Reflectance Distribution Function(BRDF) feature of materials' surface for per-pixel material recognition. Based on this work, Gu *et al.* [**?**] proposed a algorithm to obtain an optimal coded illumination, after projecting to which, the spectral BRDFs of different materials can be maximally separated. And then they utilized this coded illumination to directly classify raw materials directly. In [9], they employed the same model in [**?**], but to utilize coded illumination to measure projections of Bidirectional Texture Functions (BTFs) directly for material classification. Both of these works can obtain a high accuracy, however, these two methods require a very complicated and expensive device (a dome with cameras and light sources fixed at different direction) to collect BRDF features on material surface. In [10], Shiradkar *et al.*propose a method to use handheld flashlight camera to capture a 1D BRDF splice for material classification and ink identification. Lombardi *et al.*[11] proposed an algorithm to estimate BRDF directly from single images, however in this algorithm, geometry information of the object is required.

Other than BRDF features which is acceptable and wildly utilized in material recognition, we still have some other features which are reliable to employ. In computer vision field, convolution owns a impregnable position for many situations. Filters usually have a much smaller size than the images, but after a convolution operation with them, many features of the original image will strongly appear,

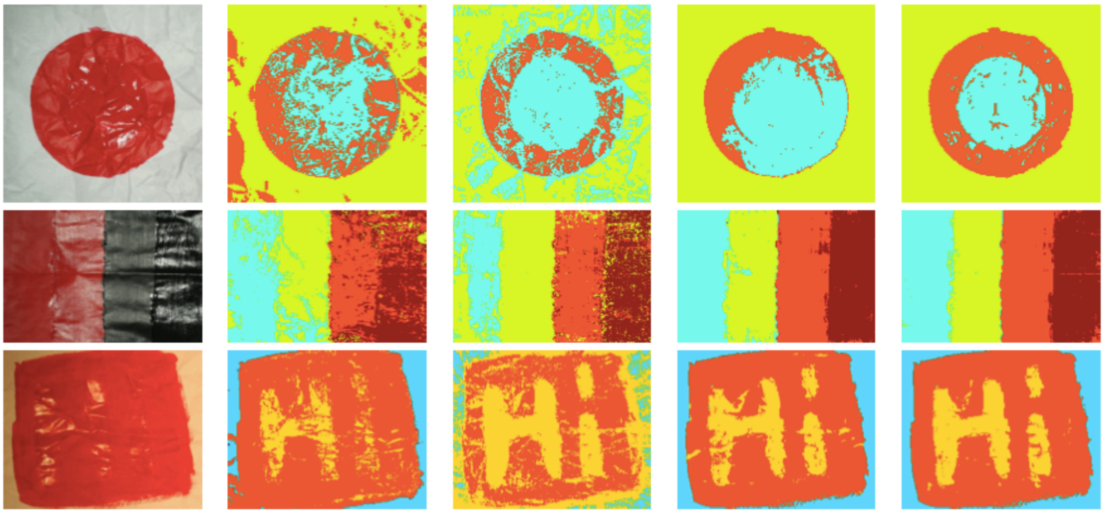


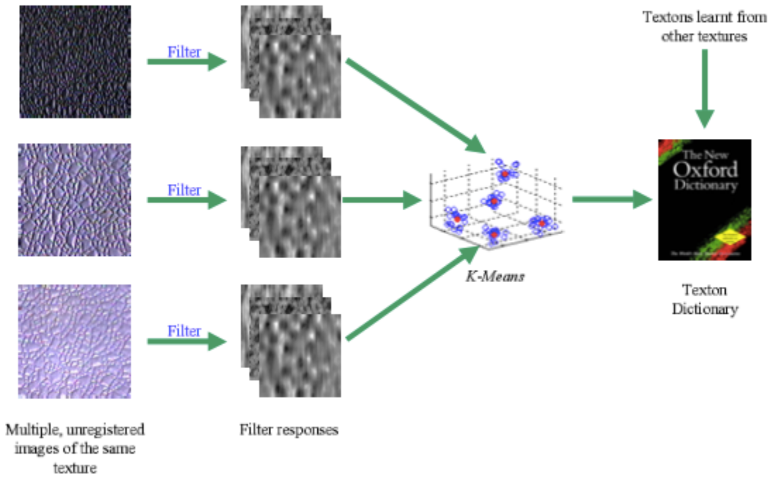
Figure 2.1: Objects made of different materials [2]

especially for materials' different textures.

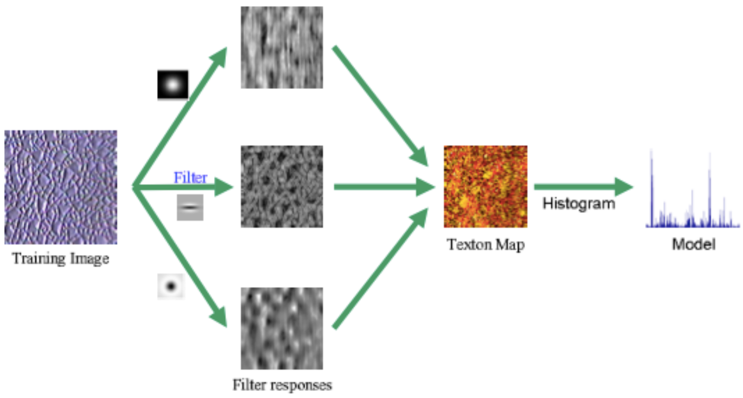
Varma *et al.*[3] proposed an algorithm to present materials' textures as textons, which is features obtained by convoluting images with a filter bank and clustering the responses via K-Means. At the first stage(Fig. 2.2(a)), multiple, unregistered images of the same texture are convolved with a filter bank. Then obtained filter responses will be clustered into N textons via K-means algorithm with N clustering centres. This procedure is repeated to train textons from different texture classes and finally all the trained textons will be combined to form a texton dictionary. In the material training stage (Fig. 2.2(b)), training image of a certain material category is convolved with the same filter bank as stage I, then each filter response is labelled with the closest texton in the texton dictionary. Finally the histogram of textons, showing frequency of each texton's presence in texton map with labelling, will form a material-texton model of the training data. Thus, after training a set of material-texton models of a material class, a testing image can classified easily by comparing its material-texton model with the trained model dictionary and picking the closet one as its material category.(Fig. 2.2(c)) This work is widely employed and has been developed to many good algorithms by other researchers. Although our algorithm is different from this work, the basic procedures and methodologies are quite similar. In [12], the same group of Varma *et al.*proposed a new approach to represent textures as a set of exemplar patches instead of filter bank responses, and this method has been shown better than filter bank based representation. However, patch representation has a disadvantage of quadratic increase in the dimension of feature space with the neighbourhood patch size increasing.

In [13], Yaccob *et al.*investigated the relationship between the appearance of a same material when it was dry and wet. Material classification and recognition was also researched with a virtual synthesised dataset. [14][15]

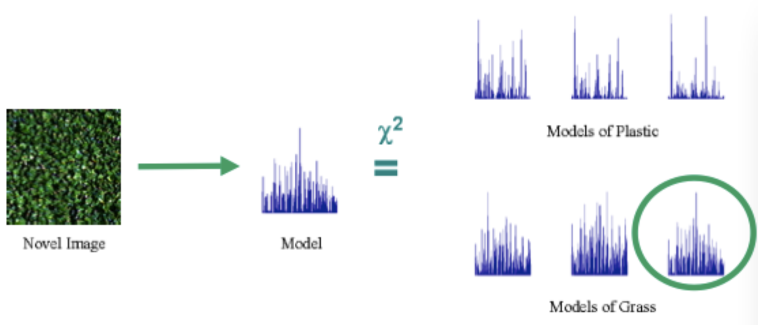
To deal with multi-scale task in material recognition, Kang *et al.* [5] proposed algorithm to compute texton clustering in random forest based on the work of [4]. Kadir *et al.* [16] proposed a concept of scale saliency to extract areas of



(a) Learning stage I: Generating the texton dictionary.



(b) Learning stage II: Model generation.



(c) Classification stage.

Figure 2.2: Procedures of texton-based texture recognition. [3]

different scales in an image. Based on this work, in [26], Blunsden *et al.*proposed a method to extract textons of VZ's MRF model at different scales. In [17], Li *et al.*introduced an unsupervised model of Multi-Scale Spike-and-Slab Sparse Coding to learn multi-scale material features,

In [18], Sharan *et al.*build a new material database with single images of our every life materials. Thanks to this good database, many new research has been conducted successfully. In [19], Liu *et al.*utilised a rich set of low and mid-level features from materials. Then they combine an proposed model of augmented Latent Dirichlet Allocation (aLDA) with these features under a Bayesian generative framework to learn an optimal combination of different features. In [6], Fleming *et al.* proposed a method to predict the material's perceptual properties(e.g. hardness, rigidity, colorfulness), and proved that these qualities were systematically related to their corresponding material classes. In this work, they conducted two experiments to research the interactions between materials and material properties in both visual and semantic domains. Based on this work, Schwartz *et al.*[7] proposed a more systematically generalized approach to recognize per-pixel material' traits(e.g. smooth, shiny, liquid, organic), which had been proved to be discriminative and object-independent. Schwartz *et al.*untilzed these per-pixel material traits in material category recognition and segmentation successfully.

In this thesis, our research is most closely related to Schwartz's work of recognizing per-pixel material traits.

2.2 Database

In this section, we briefly introduce our selection of an ideal image database for this work.

Recently, there exist many excellent database of computer vision provided by researchers. *Columbia-Utrecht Reflectance and Texture Database* (CuRRET)[20] is a very famous material database, containing BRDF database, BRDF parameter database and a BTF database with it. However, in Figure 2.3 we can find that although this database provides abundant information of materials' texture, its images are too different from our everyday life photo. So it may not be a good choice in this research.

Microsoft Research Computer Vision Team at MSR Cambridge also provide a popular object database for free, containing abundant object categories in everyday life photos. They also provide ground truth masks for users to annotate a certain object category they want to utilize. However, this is not a database designed for material recognition, so it is really difficult to prepare and pre-process patches for material research effectively.

Fortunately, Sharan *et al.*provide us an excellent database, *Flickr Material Database*(FMD), for material researches[18]. In this database, there are images of 10 material categories(*fabric, foliage, glass, leather, metal, paper, plastic, stone,*

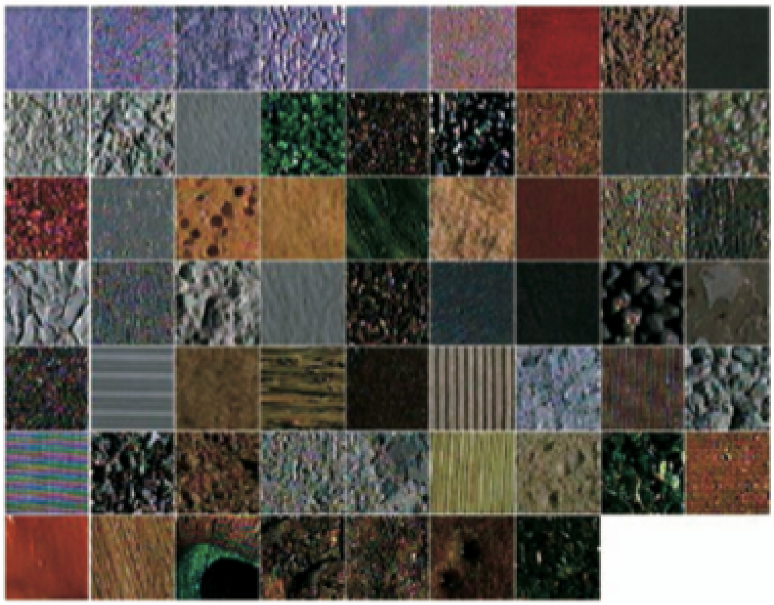
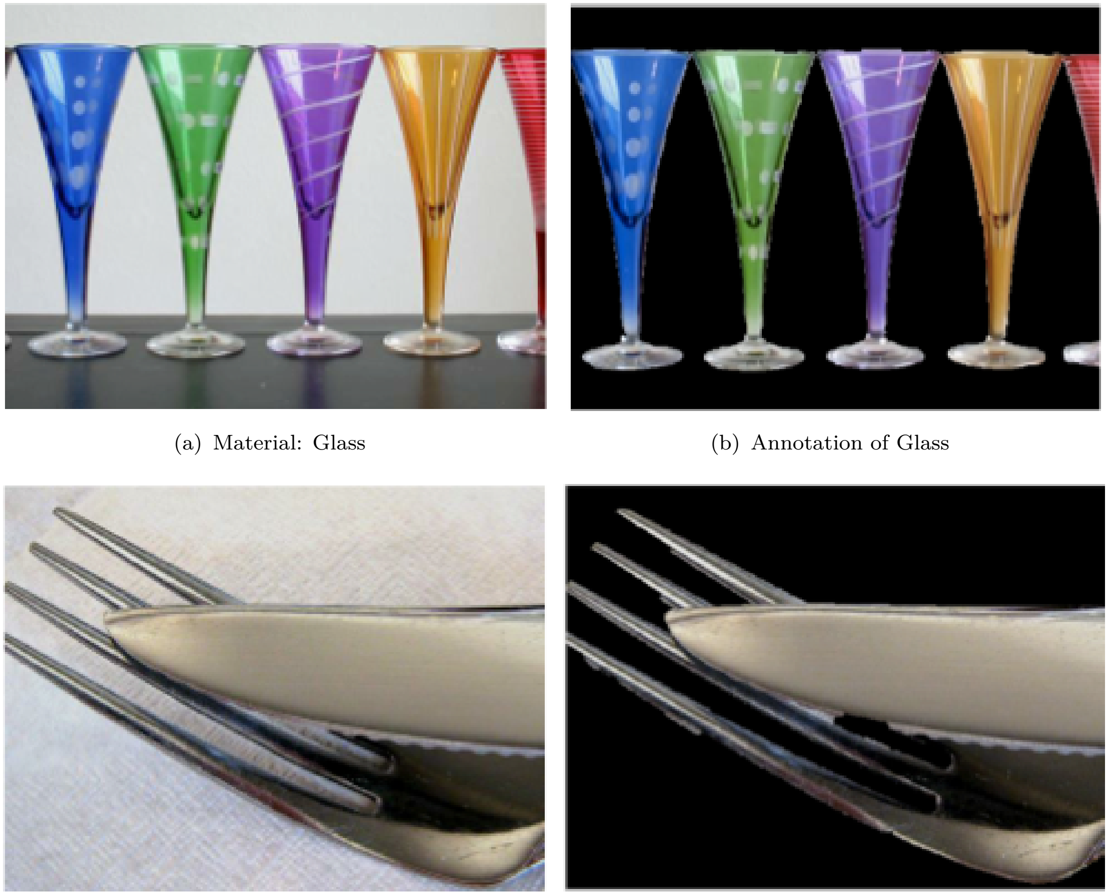


Figure 2.3: Material images from the Columbia-Utrecht database [3]

water and wood), and 100 images per one category. Consider that images in this database all come from Photography Website:*Flickr.com*, all of them are vivid, colorful, and have a very high quality, as some images shown in Figure 2.4. Besides, they have made a mask image for each material image, so that the users can annotate the region of interest (Metarials) with the provided masks, which is quite convenient. Some annotation examples are showed in Figure 2.5.

Figure 2.4: Images of different materials in FMD [1]



(a) Material: Glass (b) Annotation of Glass

(c) Material: Metal (d) Annotation of Metal

Figure 2.5: Annotation of material region with the mask

Chapter 3

Recognize Multi-Scale Material Traits

3.1 Background

Recently, Fleming *et al.* [6] proposed a method to predict the material's perceptual properties(e.g. hardness, rigidity, colorfulness), and proved that these perceptual qualities are systematically related to their corresponding material classes. Based on this, Schwartz *et al.*[7] proposed a more systematically generalized approach to recognize per-pixel materials' traits(e.g. smooth, shiny, liquid, organic). They have proved that these material traits have ability to encode per-pixel information of materials. What's more, these properties which are independent from object-specific features, are proved to be discriminative. In Schwartz's experiment, they have successfully utilized material traits in material category recognition and image segmentation.

Schwartz's proposed method is the first one to extract and utilize material traits. However, in their research, scale chage in images, which are very important in material recognition, have not been taken into consideration.

In this thesis we propose an approach to recognize material traits with different scales, by training a training hierarchical convolutional features of material traits at different scales, and then utilizing the trained features to train multi-scale randomized decision forest classifiers to recognize multi-scale material traits.

3.2 Representing Multi-scale Material Traits

In this section, we propose an approach to recognize material traits with different scales.

In [7], Schwartz *et al.*proposed an unsupervised setting to train Convolutional

Material Features, based on a Convolution Auto-encoder model[21]. With this model, a sparse set of filters, which encode different material traits, will be trained by solving an minimum problem.

With a set of filters W, input images I_i can be transformed to a feature space E_i with (3.1):

$$\mathbf{E}_i = h(\mathbf{W} * \mathbf{I}_i + b_e) \tag{3.1}$$

where h() is an activation function (3.2), and b_e is encoder bias,

$$h(x_i) = \begin{cases} 0, & \text{if } x_i < 0 \\ x_i, & \text{if } 0 \leq x_i \leq 1 \\ 1, & \text{if } x_i > 1 \end{cases} \tag{3.2}$$

Then, with another set of filters W', images in feature space E_i can be reconstructed to the original images R_i:

$$\mathbf{R}_i = \mathbf{W'} * \mathbf{E}_i + b_r \tag{3.3}$$

Here our target is to make the original image I_i and reconstructed image R_i as similar as possible, described as a reconstruction error term T_r:

$$T_r = \frac{1}{N}\sum_{i=1}^{N} \| \mathbf{I}_i - \mathbf{R}_i \|_2^2 \tag{3.4}$$

Then by adding a sparsity term T_s:

$$T_s = \| p - \frac{1}{N}\sum_{i=1}^{N} \mathbf{E}_i \|_2^2 \tag{3.5}$$

where p is a very small constant.

and a weight-decay term T_w:

$$T_w = \| \mathbf{W} \|_2^2 + \| \mathbf{W}' \|_2^2 \tag{3.6}$$

Finally we can obtain filters W and W' by optimizing such a minimum problem:

$$\min_{W,W'} C = T_r + \alpha T_w + \beta T_s \tag{3.7}$$

where α and β are corresponding weights of weight-decay term and sparsity term.

In our proposed method, in order to train multi-scale classifiers for multi-scale material traits recognition, we should obtain features which can separately represent materials images at different scales. Therefore, we train multi-scale CAE filters $W_k,\ k = \{1, 2, 3, ..., S\}$ for S different scales. In order to train multi-scale filters, we prepare S groups input images I with different scales, and utilize images in each group with one particular scale to train filters W individually. These trained filters $W_k,\ k = \{1, 2, 3, ..., S\}$ will be employed in next steps to train multil-scale randomized decision forest classifiers to recognize material traits with different scales.

3.3 Multi-Scale Randomized Decision Forest

Randomized Decision Forest [22] is an ensemble learning model developed by Breiman *et al.*from the decision tree algorithm. With a good performance, random forest has become a very popular algorithm widely utilised for classification, regression and other tasks.

Figure 3.1 gives a decision tree ,which contains many split nodes and leaf nodes. Each node represents a prediction on a feature, and it will split into branches connected to its next level with its predicting result. A node, which arrives the maximum level or can not be split any more, will be noted as a leaf node. Each leaf node will make a final decision, classifying the input data to one class label. Therefore, we can obtain a class distribution $P(c|l)$, representing frequency of each class label given by all the leaf nodes, in a decision tree.

Figure 3.2 shows a random forest which is an ensemble of T decision trees with different structures. When predicting the class of an input sample, each decision tree in the forest will obtain a class distribution $P(c|l)$. Therefore, T trees in this forest will produce T histograms of class distribution $P(c|l_t), t = (1, 2, 3, ..., T)$. Finally, class label of input sample will be predicted by the average class distribution:

$$P(c|L) = \frac{1}{T}\sum_{t=1}^{T} P(c|l_t) \tag{3.8}$$

In our experiment, multi-scale random forests $\mathcal{F}_k, k = (1, 2, 3, ..., S)$ will be trained from balanced sets of positive and negative material traits samples at S different scale levels. Since in our setting we have $S = 4$, four random forests with T trees in each will be trained, as shown in Figure 3.3. When predicting a novel sample, forest at scale level k will obtain T class distributions $P_k(c|l_t), t = (1, 2, 3, ..., T)$ and the final prediction of the k-th forest will be made by computing its average class distributions:

$$P_k(c|L_k) = \frac{1}{T}\sum_{t=1}^{T} P_k(c|l_t), \quad k = (1, 2, 3, ..., S) \tag{3.9}$$

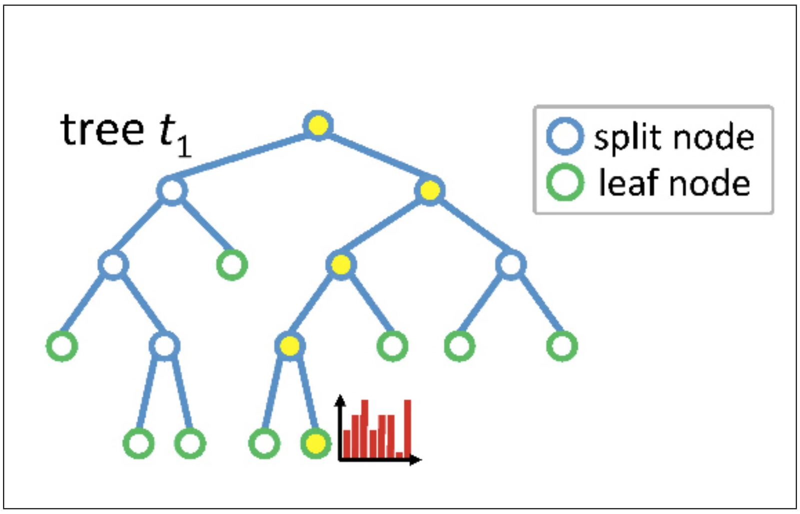


Figure 3.1: A single decision tree[4]

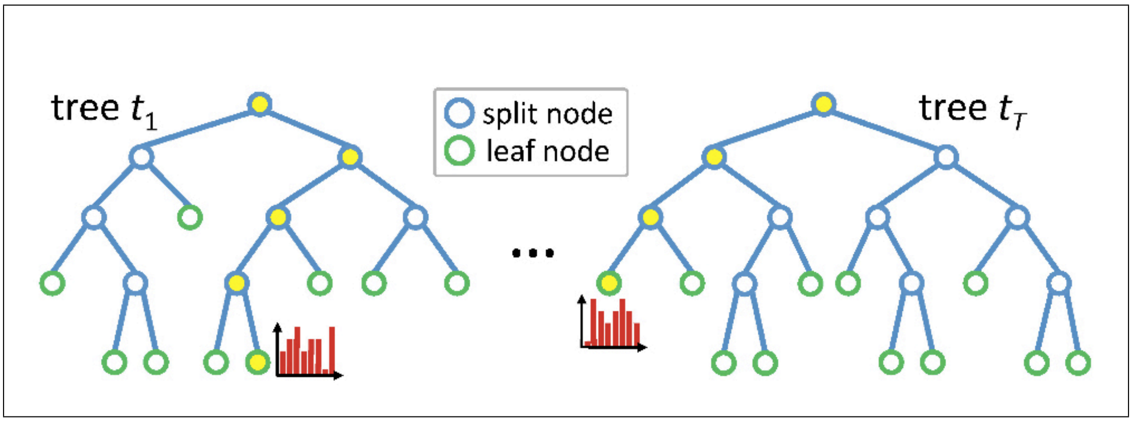


Figure 3.2: A random forest with T trees[4]

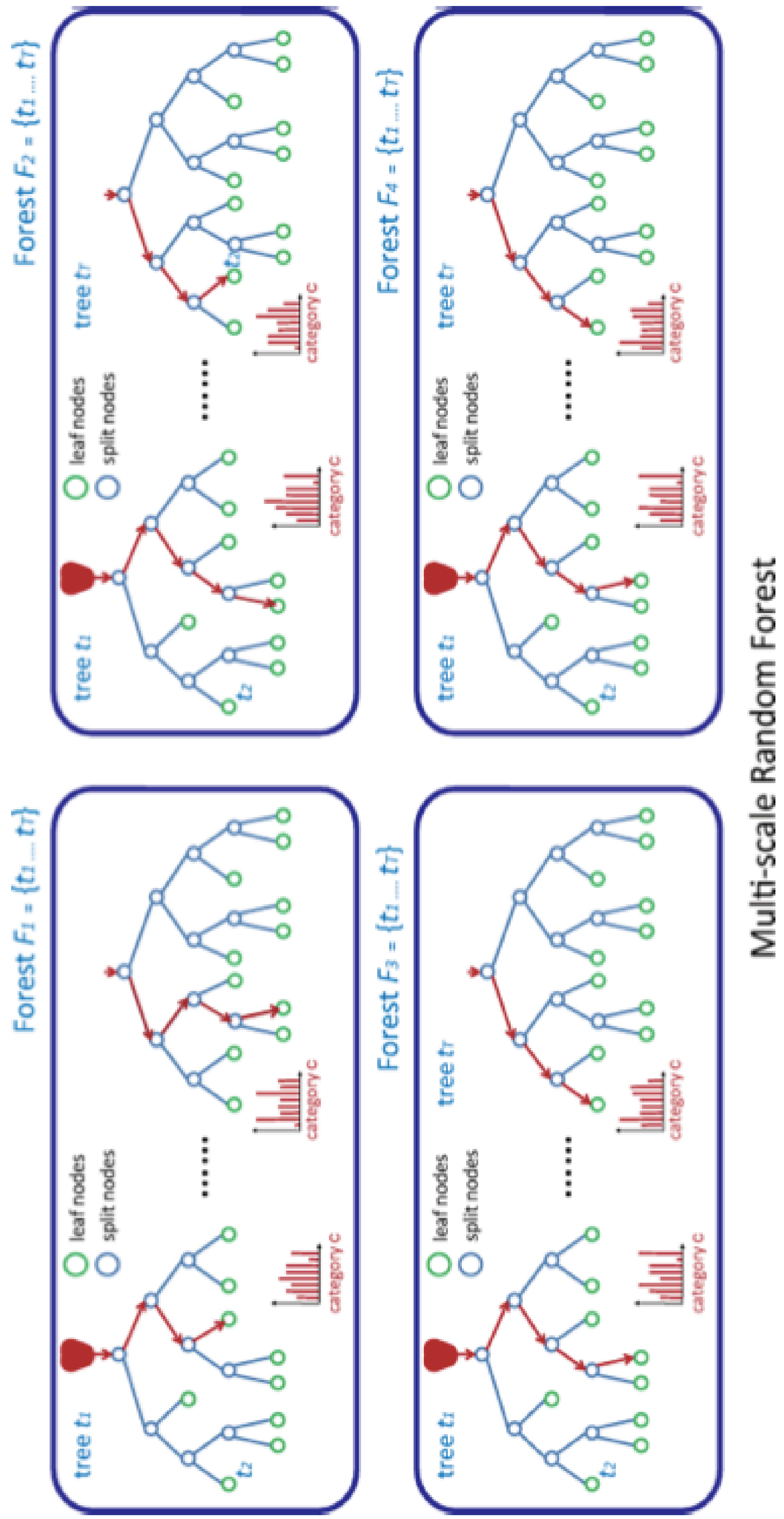


Figure 3.3: Multi-scale random forests[5]

3.4 Strategies for Scale Selection

In this section, we propose three approaches for our system to select a reasonable predicting result from multi-scale forests.

As introduced in 3.3, in multi-scale random forests $\mathcal{F}$ with S scale levels, average class distribution for k-th scale forest can be computed as (3.9). The computed average class distributions $P_k(c|L_k)$ will be utilised to predict class label $c = Class(i)$ of an input sample i. In this research, we only have 2 classes, $c = 0, 1$, where the case $(c = 0)$ means the input patch i belongs to this material trait class, and $(c = 1)$ means patch i is not a patch with this material trait.

In our experiment, we have $k = 4$ scales for the training and testing data, and also we trained 4 RDF classifiers for different scale. As results show in Section 4.3.2, the highest recognition accuracy will be obtained from classifier with the same scale as the input sample. However, in a real recognition task, we do not know to which scale the input patch should belong. Therefore, a reliable strategy of scale selection is very important for our system to make the final decision to recognize the input data.

We propose three strategies for scale selection:

Maximum Prediction Probability

The simplest model is to pick the highest probability from average class distributions of $(c = 0)$ obtained from four classifiers:

$$k_{chosen} = \arg\max_{\mathrm{k}}(\mathrm{P_k}(\mathrm{c} = 0|\mathrm{L_k}), \quad k = (1, 2, 3, 4) \tag{3.10}$$

With this approach, we just simply predict the input patch's class label by checking the highest prediction probability. However, the lowest probability, which plays a very important role for the prediction of negative label, is ignored in this model. Our experiment results prove that this model has a unstable performance and is sometimes error-prone.

Maximum Confidence

Based on the maximum probability model, we propose another model for scale selection by checking the confidence of each prediction. Here, confidence of the prediction made by each RDF classifier is defined as the distance between positive prediction probability with 0.5:

$$Distance_k(i) = |P_k(c = 0|L_k) - 0.5| \tag{3.11}$$

$$k_{chosen} = \arg\max_{\mathrm{k}} (\mathrm{Distance_k(i)}), \quad k = (1, 2, 3, 4) \tag{3.12}$$

With this model, both maximum probability and minimum probability are considered to make the final decision.

Average Probability

In addition, for this 2-class problem, we employ another simple model by computing the average probability of the positive class among all the forests:

$$P_{AVG}(c = 0|L_k) = \frac{1}{S}\sum_{k=1}^{S} P_k(c = 0|L_k), \quad k = (1, 2, 3, 4) \tag{3.13}$$

Then, instead of a strictly defined 'Scale Selection', we predict the class label of input sample i directly with this average probability:

$$Class(i) = \begin{cases} 0, & \text{if } P_{AVG}(c = 0|L_k) \geq 0.5 \\ 1, & \text{if } P_{AVG}(c = 0|L_k) < 0.5 \end{cases} \tag{3.14}$$

Chapter 4

Experiments

In this chapter, we introduce our experiments in four sections. Firstly, we introduce how we prepare training and testing data for the following steps. Then, our implementation to train convolutional auto-encoder filters and randomized decision forest classifiers with different scales will be introduced. Both of experiment settings and testing results will be given in these two sections. Finally, we discuss some strategies for scale selection.

4.1 Data Preparation

Since this research is to do per-pixel material traits recognition, our experimental objects should also be pixels instead of the whole material images. However, in this work, all the features we employ to train classifiers (e.g. Filter responses, Local Binary Pattern, Color histogram) require the neighbouring information around a pixel. Therefore, instead of a single pixel, we extract a small patch around a pixel, and utilize this small patch to represent its corresponding pixel.

In this work, we need to prepare two kinds of material patches:

1. Image patches of all material categories for training unsupervised convolutional material features;
2. Balaced sets of positive and negative **material trait** patches for training randomized decision forest (RDF) classifiers of each material trait and testing.

We have to state that all these patches contain 4 different scales, and all of them should be of a same image size.

Our setting is to prepare patches at scales of (16*16),(32*32),(48*48) and (64*64), noted as 1x, 2x, 3x, 4x, from images of original sizes in FMD (Table. 4.1). And then we down-sample all the patches larger than (16*16) back to size (16*16)

Table 4.1: Material patches with different scales

Scale No.	1	2	3	4
Original Size	(16,16)	(32,32)	(48,48)	(64,64)
Target Size	(16,16)	(16,16)	(16,16)	(16,16)
Note	1x	2x	3x	4x

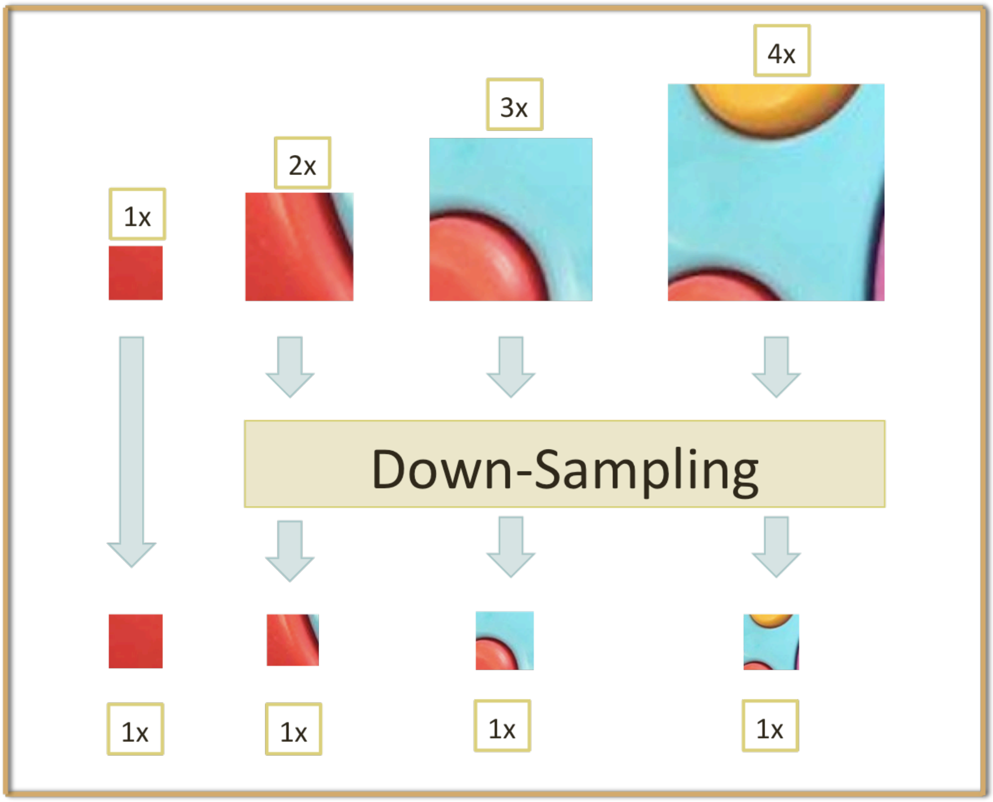


Figure 4.1: Preparation of material patches with different scales

for later utilization.(Figure 4.1) More details about the procedure of extracting each type of patches in our experiment will be introduced in the following sections.

4.1.1 Patches for Training CAE Filters

As we have introduced in Chapter 3, convolutional auto-encoder filters are good features to represent material traits for all the images. Therefore, patches for training CAE filters are not restricted to a certain material image group. As Figure 2.5 shows, in Flickr Material Database, material region in the original image has already been annotated by its corresponding mask. With these masks, patches for training CAE filters can be extracted directly from material regions.

In our experiment, we prepared four groups of 2048 patches for training CAE filters. Each group is extracted randomly from all the material images of FMD, with only one scale of 1x, 2x, 3x or 4x, as noted in Table 4.1. Then we will utilize these four groups of patches to train CAE filters with four scales.

4.1.2 Material Trait Samples for Training RDF classifiers

Different from training CAE filters which are a generalized representation for all the material traits, in this section, we will train Randomized Decision Forest classifiers to recognize each material trait. Therefore, positive and negative training samples should be individually prepared for different material traits. We prepare five material traits: Organic, Metallic, Liquid, Smooth and Woven for this experiment.

In FMD, regions of each material category have been annotated by masks, which are also provided in the database. However, annotations of different material traits are not given. Therefore, one work we have to do before training classifiers is to draw new masks for each material trait in the original images in FMD.

Figure 4.2 gives an example of annotating 'Smooth' region to show how we extract material trait regions from FMD images.

Then, we extract 3500 positive samples and 3500 negative samples from material trait regions and no material trait regions we have annotated at 4 scales. The settings of these patches are the same as shown in Table 4.1. Then, 2500 positive samples and 2500 negative samples are set as training samples and other 1000 positive, 1000 negative samples are set as testing samples.

4.1.3 Whitening

Before we utilize these patches in the following steps, all the training data should be properly whitened. For whitening operation, all the patch data should be reshaped to a 2-D matrix, with each row representing one patch, and each column representing a pixel value of one channel in this patch.

When we train on a raw data matrix with features in columns and data points in rows, information is always redundant, particularly when the original data is an image, in which adjacent pixel values are highly correlated. Therefore, before we train on such a data, we always have to do whitening to make the data less redundant. by making the features less correlated with each other with the same variance.

Usually we have two model to do the whitening, Principal Components Analysis (PCA) and Zero-Phase Whitening Filters (ZCA).[23] [**?**]

Assuming that we have an input data $\mathbf{I}$ with features in columns and data points in rows(in this work, rows index (16,16) patches and features in columns are pixel values in three channels $[\mathcal{R} :(16 * 16) + \mathcal{G} :(16 * 16) + \mathcal{B} :(16 * 16)]$). To whiten this data, we firstly have to make it become a centred data $\mathbf{X}$, by doing:

(a) The original image in category: Plastic

(b) Mask we drew to annotate 'Smooth' region

(c) 'Smooth' region extracted from the original image with the mask

Figure 4.2: An example of annotating 'Smooth' region in a 'Plastic' image

$$\mathbf{X} = \mathbf{I} - mean(\mathbf{I}) \tag{4.1}$$

Then we compute the covariance matrix $\mathbf{C}$, having eigenvectors in columns of $\mathbf{E}$ and eigenvalues on the diagonal of $\mathbf{D}$, so we obtain:

$$\mathbf{C} = conv(\mathbf{X}', \mathbf{X}) = \mathbf{E}\mathbf{D}\mathbf{E}^\top \tag{4.2}$$

Finally PCA whitened data is given by:

$$\mathbf{W}_{PCA} = \mathbf{D}^{-1/2}\mathbf{E}^\top \tag{4.3}$$

A PCA whitened data should have a property that it will still stay whitened after rotation. That means if we introduce an orthogonal matrix $\mathbf{R}$ to compute $\mathbf{W} = \mathbf{R}\mathbf{W}_{PCA}$, the result $\mathbf{W}$ will still be a whitened matrix. With this property, ZCA whitening is introduced, letting $\mathbf{E}$ be the orthogonal matrix and finally we will obtain:

$$\mathbf{W}_{ZCA} = \mathbf{E}\mathbf{D}^{-1/2}\mathbf{E}^\top = \mathbf{E}\mathbf{W}_{PCA} \tag{4.4}$$

After images whitening, ZCA-whitened images will still resemble the input raw images, while the PCA-whitened data will appear completely different from the original data. In this research, since we are going to utilize whitened data to train convolutional features, the local properties of original images are also important. So here we have to choose ZCA to whiten our material image data.

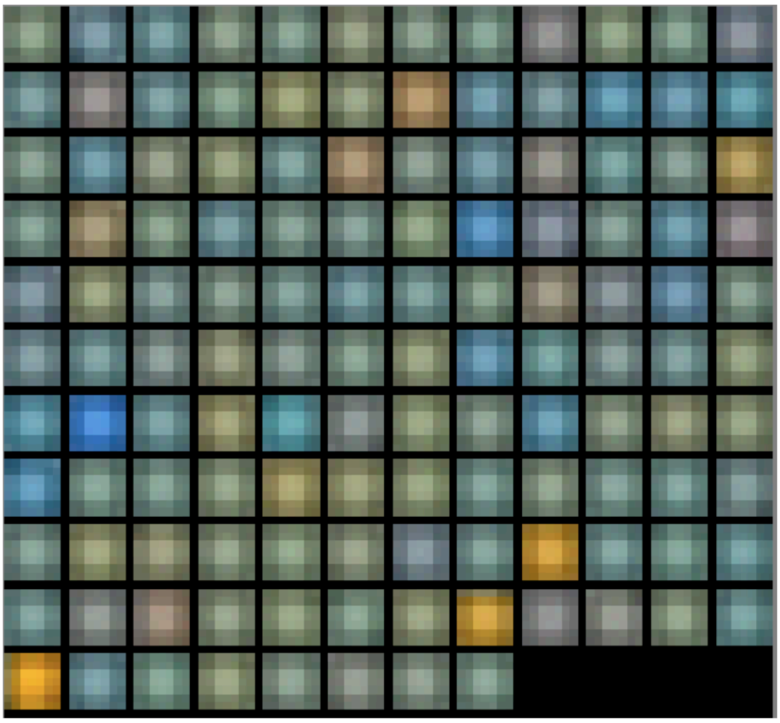
Figure 4.3 shows the results of training filters from data with and without ZCA whitening. Filters trained from original raw data show identical appearance and it is also proved that such filters have little contribution in a classifier.

4.2 Train Multi-scale CAE filters

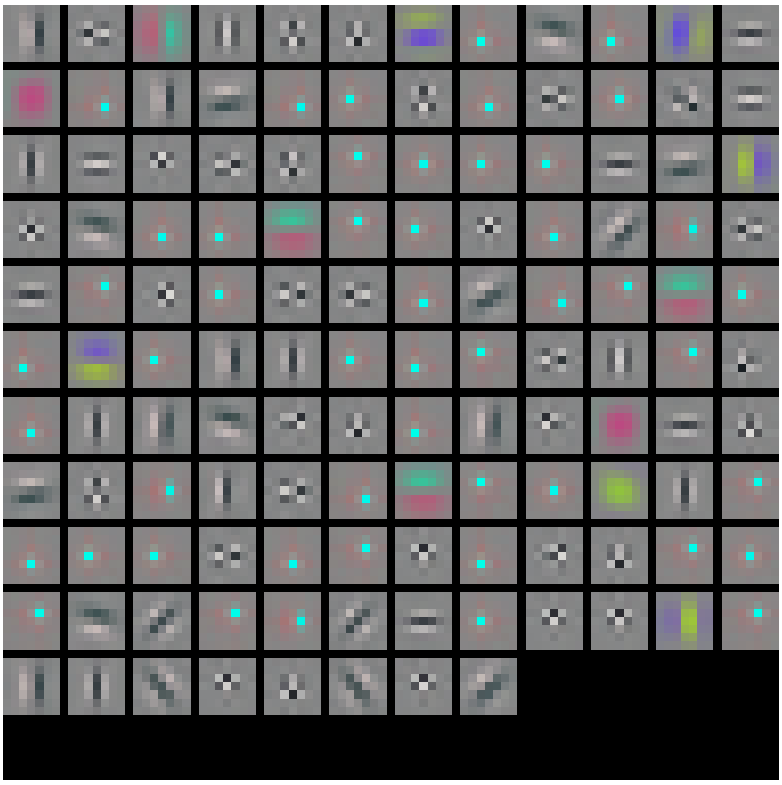
4.2.1 Experiment Settings

In our experiment, we utilize four groups of patches with only one scale 1x, 2x, 3x or 4x to train four sets of CAE filters. Each patch has a size of (16,16) with 3 color channels (RGB). For each scale, our target is to train 128 filters with a size of (7,7), and also these filters contain 3 color channels (RGB). Since we employ Python with a library Theano to do this work, both the patch data and filter data are stored in a 4-D tuple with a shape which can be accepted by Theano, as shown in Table 4.2.

We have to state that training CAE filters is quite a time-consuming work. We tried to compute only one set of CAE filters by using an Intel Core i7-2600



(a) Filters trained from original raw data



(b) Filters trained from whitened data

Figure 4.3: Filters trained from data with and without ZCA whitening

Table 4.2: Experiment Setting for training CAE filters

Patch Size	Patch Number	Patch Color Channels	Tuple Shape
(16,16)	2048	3	(2048, 3, 16, 16)
Filter Size	Filter Number	Filter Color Channels	Tuple Shape
(7,7)	128	3	(128, 3, 7, 7)

Table 4.3: Material patches with randomly mixed scales

Scale No.	5	6	7
Original Size	(16,16),(32,32)	(16,16),(32,32),(48,48)	(16,16),(32,32),(48,48),(64,64)
Target Size	(16,16)	(16,16)	(16,16)
Note	12x	123x	1234x

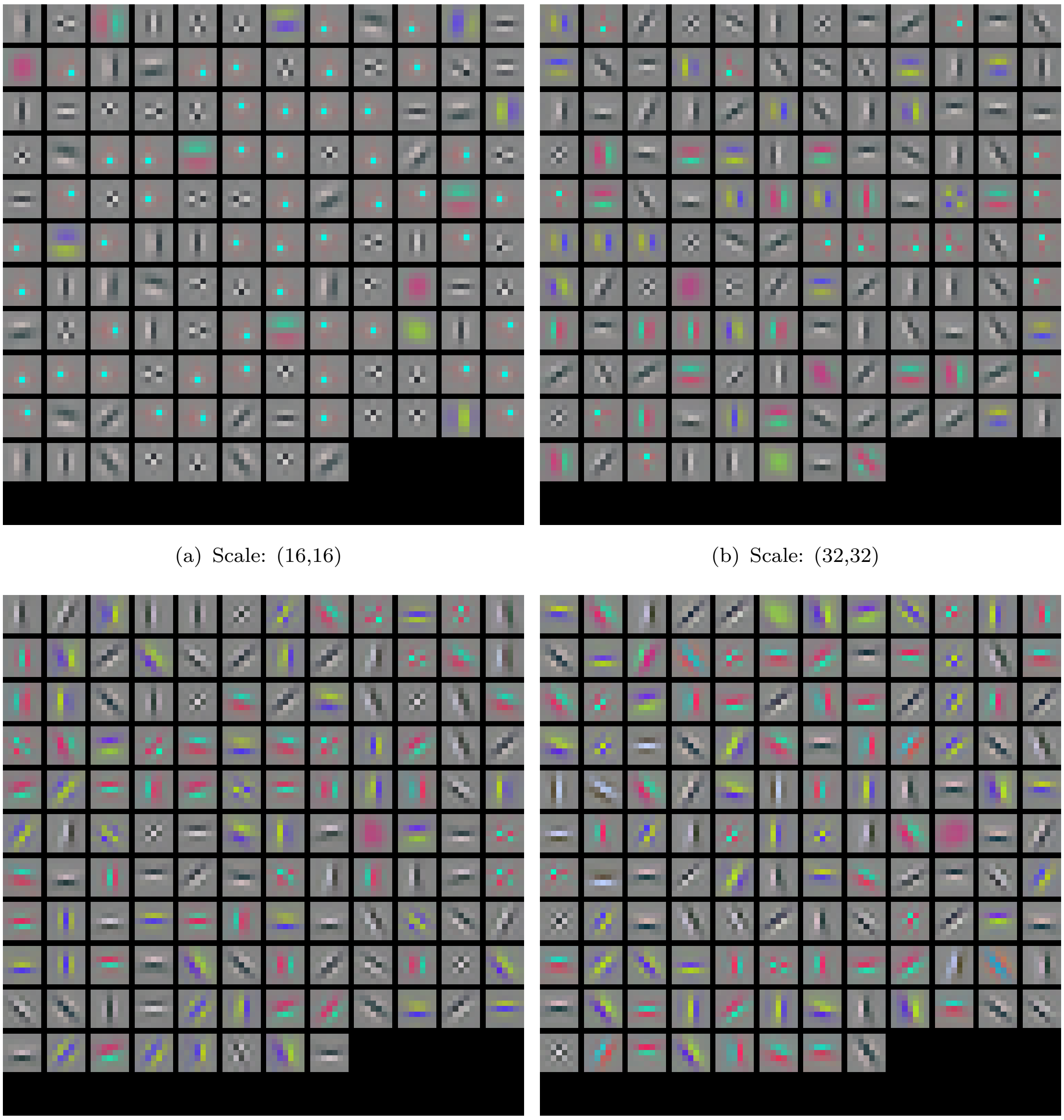
3.4GHz CPU, and the whole time it cost was around 6300s. Fortunately, the Python library Theano provides us an access to compute these filters with GPU instead of CPU. The time cost of computing 128 filters from 2048 patches with size of (16,16) was around 1160s by an NVIDIA GeForce GT530 GPU. However, this setting has been the limit of our device. With the patch number and patch size increasing, a GPU with larger memory and higher compute capability is necessary. If your device is powerful enough, more image patches and a larger patch size can be considered for this experiment. Utilizing some cloud computing services, which provide instances with powerful high-end GPU at a very low cost, to do this work is also a recommended good idea.

4.2.2 Results and Discussions

Figure 4.4 shows four sets of CAE filters we have trained from patches with four different scales. In each group, we train 128 filters which have been visualized as 128 blocks with 3 channels in Figure 4.4. We can find that filters trained from patches with different scales are quite dissimilar in appearance with each other.

Therefore, we can assume that filters trained from scale N-x should be the best representation for patches with the same scale N-x. Meanwhile, it may not be a good representation for patches with other scales. In order to prove this, three scale sets are created by randomly extracting material patches from (1x and 2x), from (1x, 2x and 3x) and from (1x, 2x, 3x, 4x). (Table. 4.3) Then we train extra three sets of filters for these mixed-scale patches.

Figure 4.5 shows an example to compare filters trained with scale 123x with single-scale filters trained separately from patches with scale 1x, 2x and 3x. We can find that filters of scale 123x are a collection of filters from each single scale. Also, filters with scale 1234x contain some different filters which appear separately in filters with single-scale 1x, 2x, 3x and 4x. With this property, we can utilize filters with different scales as a discriminative feature to train multi-scale classifiers, which can classify different material traits with different scales.



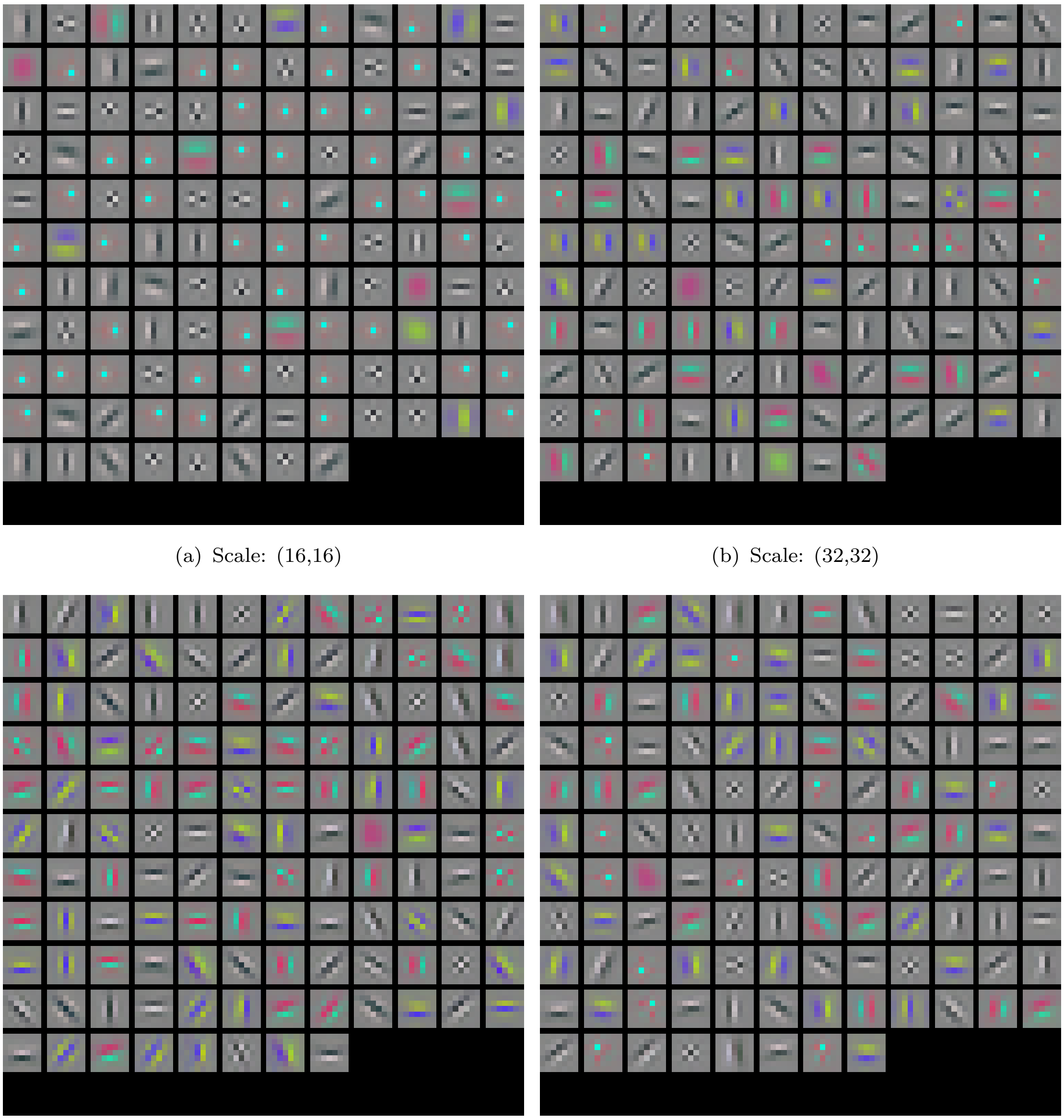
(a) Scale: (16,16)

(b) Scale: (32,32)

(c) Scale: (48,48)

(d) Scale: (64,64)

Figure 4.4: CAE filters trained from multi-scale patches



(a) Scale: (16,16)

(b) Scale: (32,32)

(c) Scale: (48,48)

(d) Random Scale: (16,16),(32,32),(48,48)

Figure 4.5: CAE filters trained from single-scale patches and randomly mixed scale patches

4.3 Train Multi-scale RDF Classifiers

4.3.1 Experiment Settings

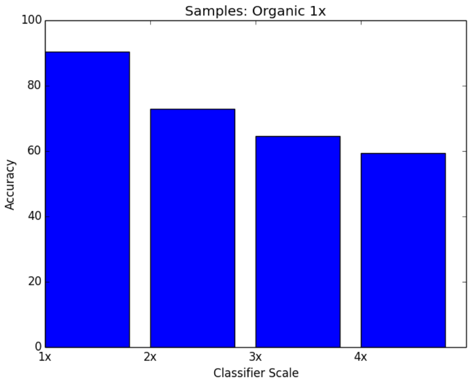
After preparing positive and negative samples with 4 scales for each material trait, we extract proper features from training samples to train randomizied decision forest (RDF) classifiers. In this experiment, we employ four kinds of features, which are CAE filter responses (CAE), Histogram of Oriented Gradients (HOG) [24], Local Binary Pattern (LBP)[25], and Color histogram, to train RDF classifiers for material trait with 4 scales.

It is important to make sure that only CAE filters with the same scale as training data can be utilized to generate filter response features for training. For instance, when training classifier of scale 2x, we should utilize training samples with scale 2x Also, all the other features should be extracted from training data with the same scale.

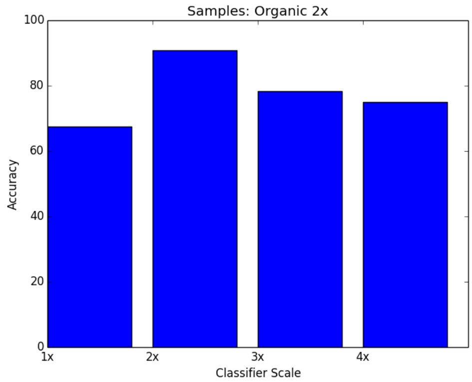
4.3.2 Results and Discussions

Figure 4.6 shows an example of recognition for Trait 'Organic' patches with 4 scales via multi-scale RDF classifiers. In these figures, we can observe that the highest accuracy for each sample is obtained from the classifier with the same scale as the testing sample. Figure 4.7 shows this interesting result more intuitively with a confusion matrix of the accuracy. The highest accuracy is always located in the diagonal, which means the combination of testing sample and classifier with the same scale. Then we check the results of other four material traits shown in Figure 4.8 - Figure 4.12, and we observe the same results as Organic, without any outliers.

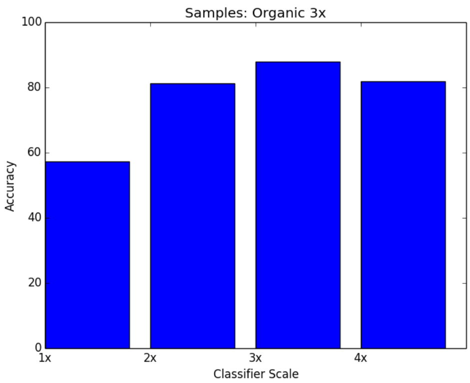
Results in this experiment sufficiently prove that scale changes have a very important impact on the accuracy in material trait recognition. In our multi-scale recognition system, if the scale of testing samples can be correctly selected, the general results of recognition will be efficiently improved.



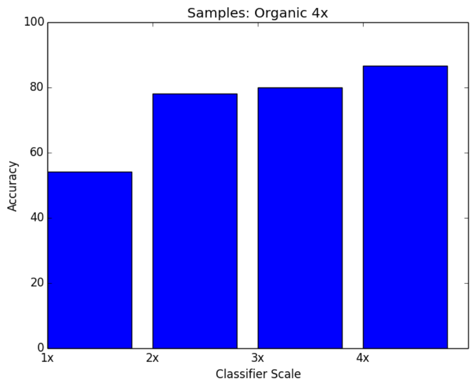
(a) Recognition Accuracy: Sample_1x



(b) Recognition Accuracy: Sample_2x



(c) Recognition Accuracy: Sample_3x



(d) Recognition Accuracy: Sample_4x

Figure 4.6: Recognition accuracy of 'Organic' samples with 4 scales

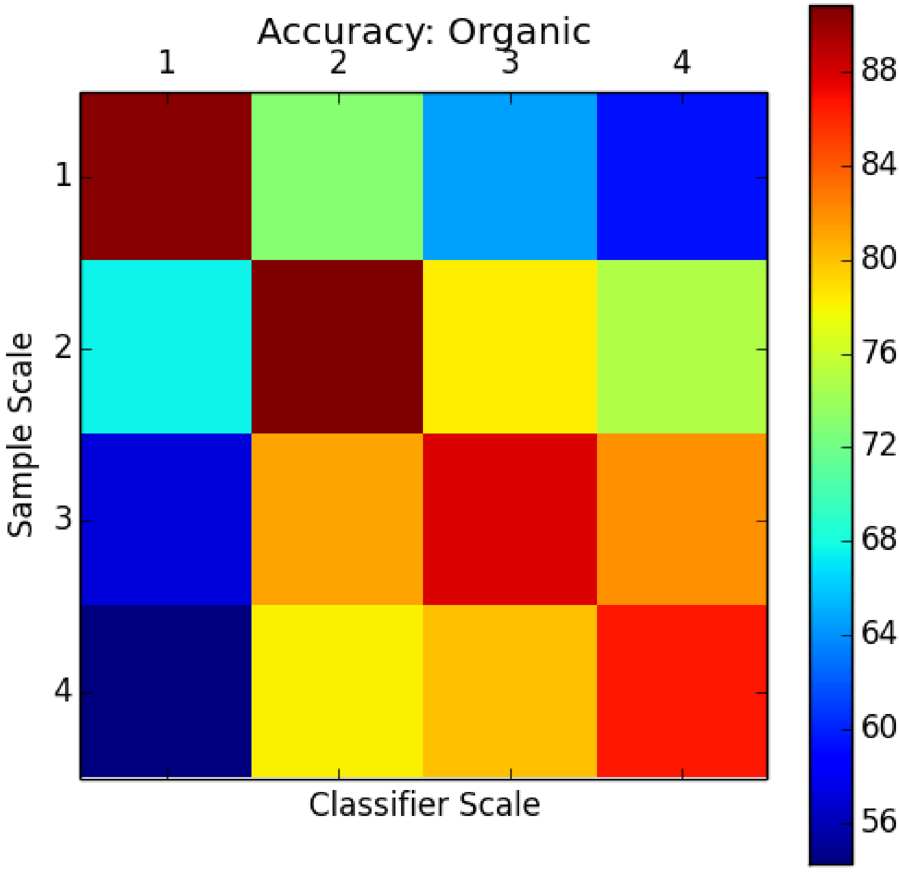
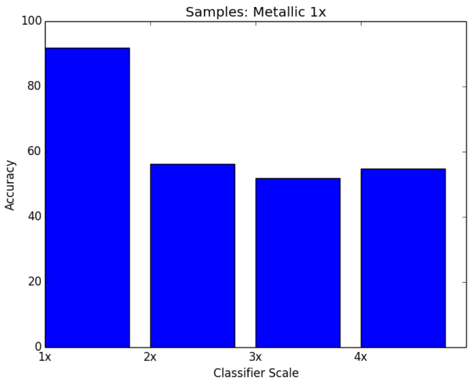
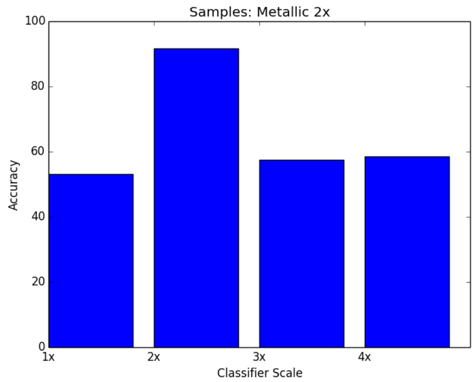


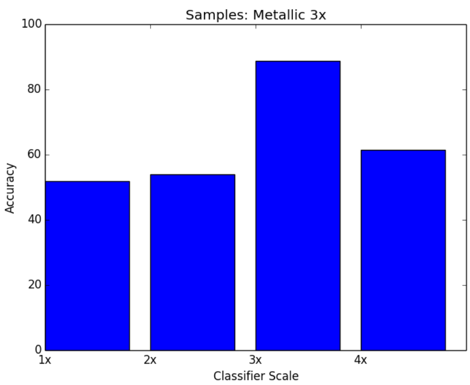
Figure 4.7: Confusion Matrix: Recognition accuracy of multi-scale 'Organic' samples via multi-scale classifiers



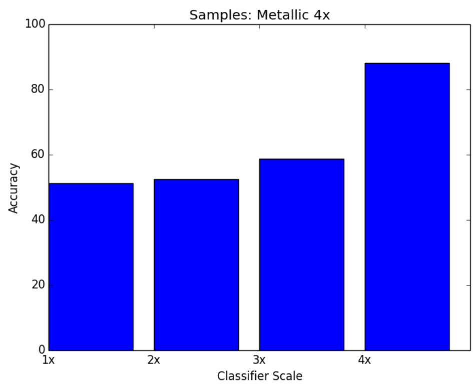
(a) Recognition Accuracy: Sample_1x



(b) Recognition Accuracy: Sample_2x

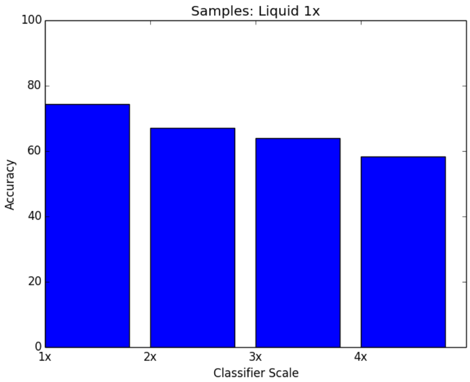


(c) Recognition Accuracy: Sample_3x

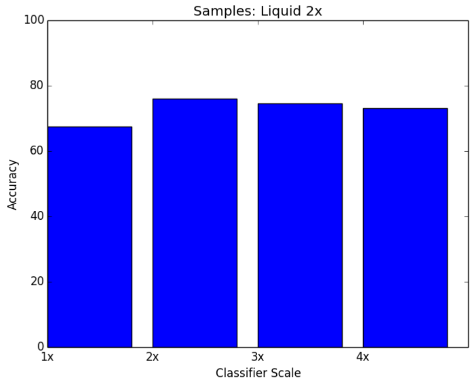


(d) Recognition Accuracy: Sample_4x

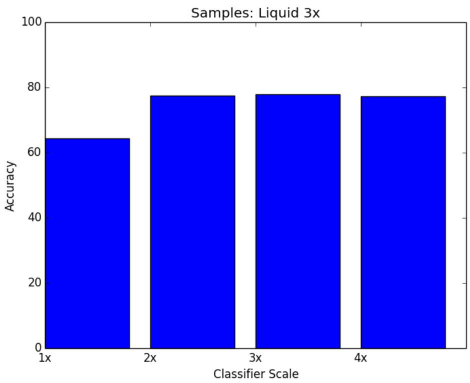
Figure 4.8: Recognition accuracy of 'Metallic' samples with 4 scales



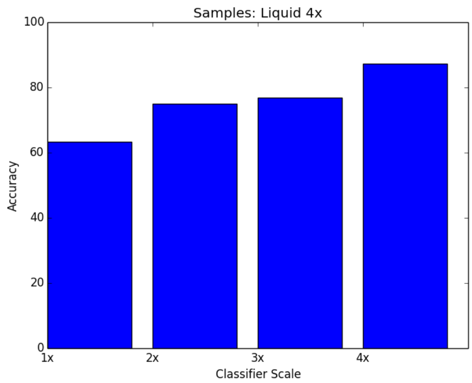
(a) Recognition Accuracy: Sample_1x



(b) Recognition Accuracy: Sample_2x

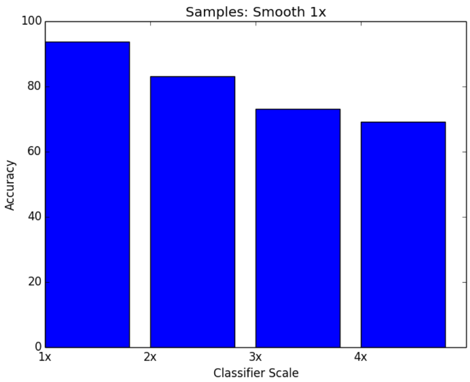


(c) Recognition Accuracy: Sample_3x

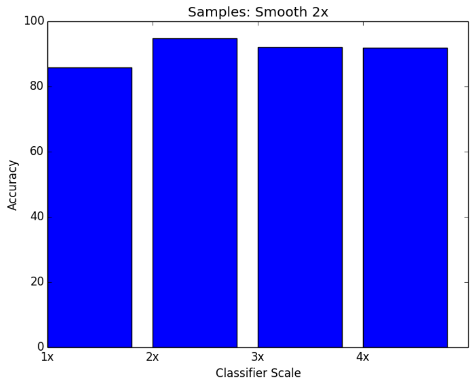


(d) Recognition Accuracy: Sample_4x

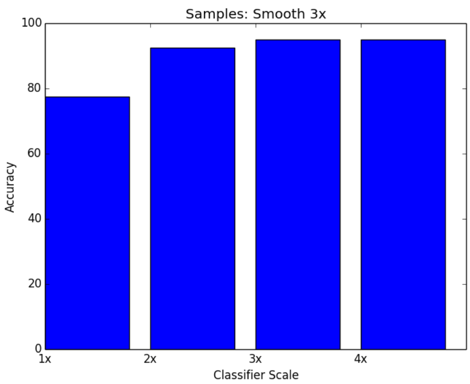
Figure 4.9: Recognition accuracy of 'Liquid' samples with 4 scales



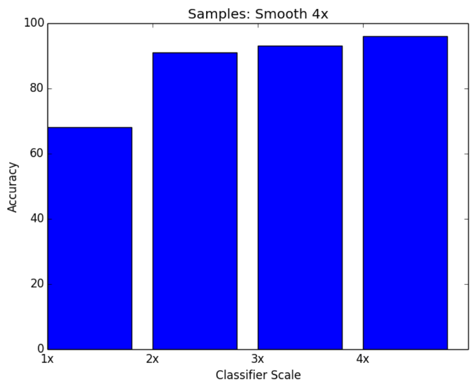
(a) Recognition Accuracy: Sample_1x



(b) Recognition Accuracy: Sample_2x

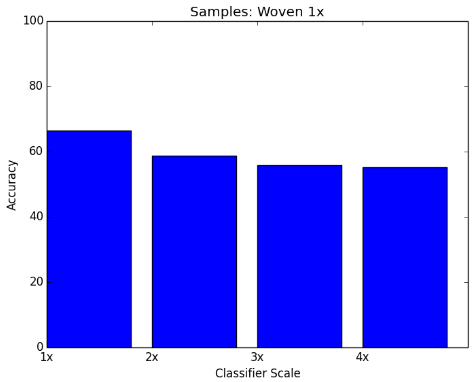


(c) Recognition Accuracy: Sample_3x

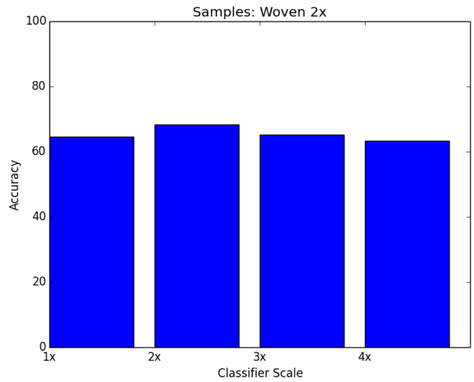


(d) Recognition Accuracy: Sample_4x

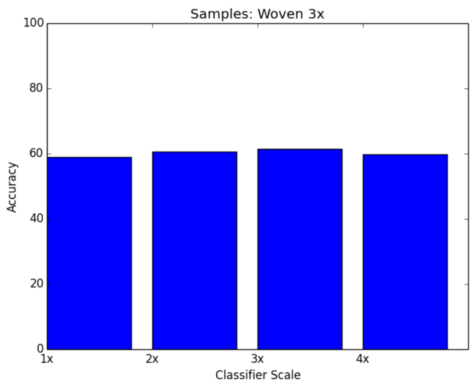
Figure 4.10: Recognition accuracy of 'Smooth' samples with 4 scales



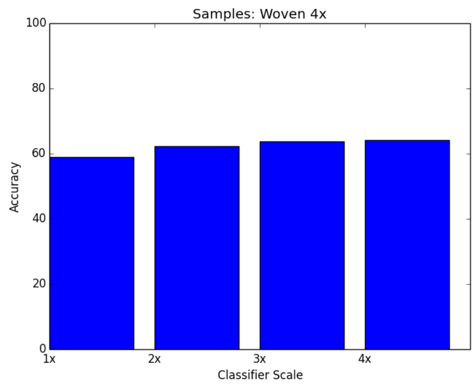
(a) Recognition Accuracy: Sample_1x



(b) Recognition Accuracy: Sample_2x

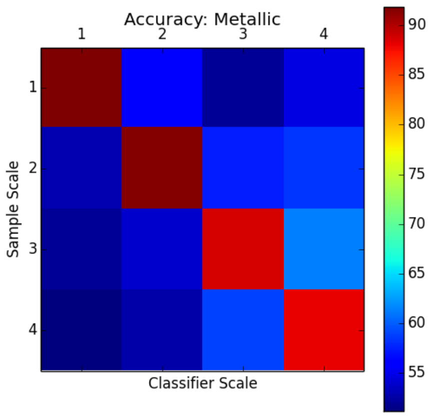


(c) Recognition Accuracy: Sample_3x

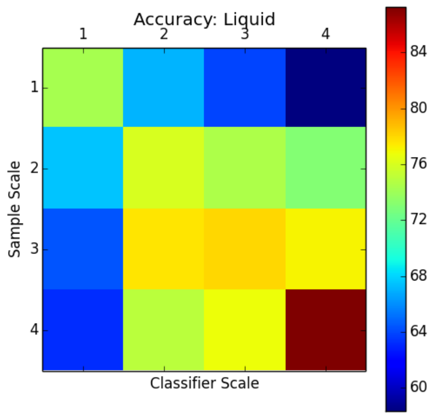


(d) Recognition Accuracy: Sample_4x

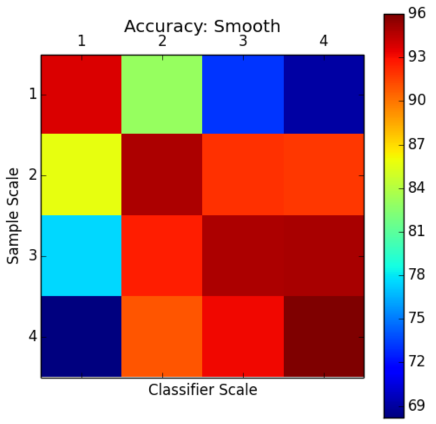
Figure 4.11: Recognition accuracy of 'Woven' samples with 4 scales



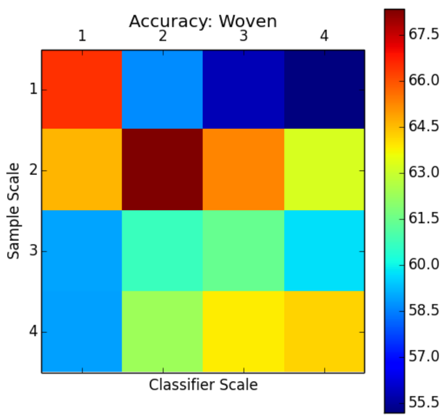
(a) Recognition Accuracy: Metallic



(b) Recognition Accuracy: Liquid



(c) Recognition Accuracy: Smooth



(d) Recognition Accuracy: Woven

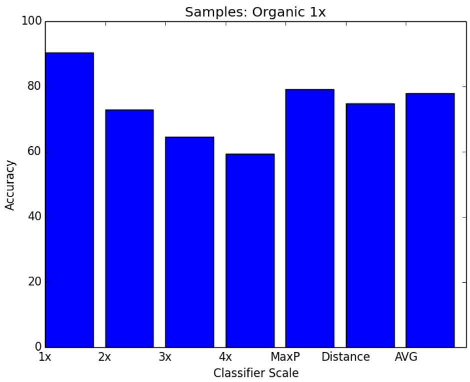
Figure 4.12: Confusion Matrix: Recognition accuracy of multi-scale samples via multi-scale classifiers

4.4 Scale Selection: Results and Discussions

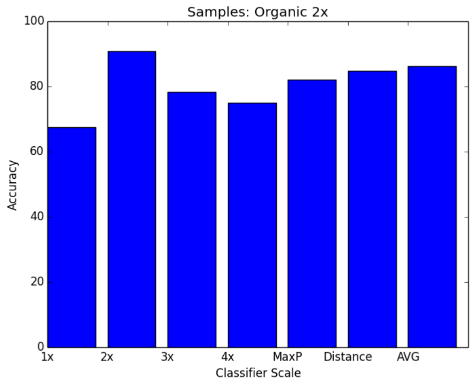
In this section, we utilize the same samples and the same setting as Section4.3.2 to do testing with multi-scale classifiers. Besides, we add three strategies for scale selection in this experiment, and finally we will compare the accuracy obtained from three scale selection strategies with results obtained from each single-scale classifier.

Figure 4.13 shows results of a testing for 'Organic' samples with 4 scales. In these four figures, we can find that strategies of Maximum Confidence and Average Probability lead to good results, with both higher accuracy and stability than an arbitrary single-scale classifier. The method of find Maximum Probability sometimes has good performance, while it is not stable for all the 4 scales. Figure 4.14 gives a statistics of the frequency that each classifier is selected by the Maximum Confidence Strategy. We can see a reasonable results that classifier with the same scale of testing samples are mostly selected. And then we check the other three results of Metallic, Liquid and Smooth samples, the same results as Organic samples can be also observed in Figure 4.15 - Figure 4.20. The only outlier is the trait of 'Woven', in whose results classifier scale can not be correctly selected. (Figure 4.22) This result may be caused by two results:

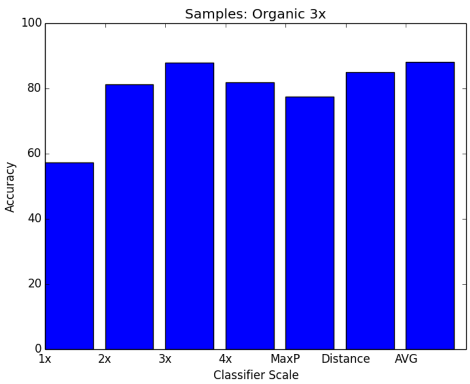
1. The source data contain some problems, making it very difficult to train a good classifier for these training samples
2. In Figure 4.22, we can find that the performance of each single-scale classifier is similar, making the prediction probabilities very confusing for Maximum Confidence strategy to select a correct scale for testing samples.



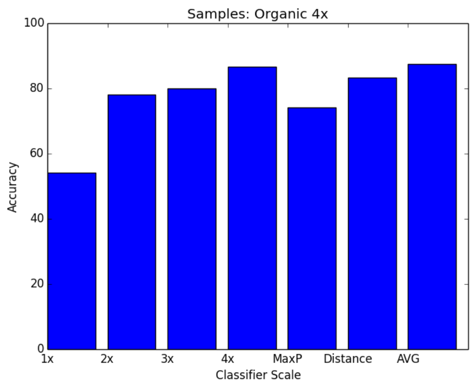
(a) Recognition Accuracy: Sample_1x



(b) Recognition Accuracy: Sample_2x

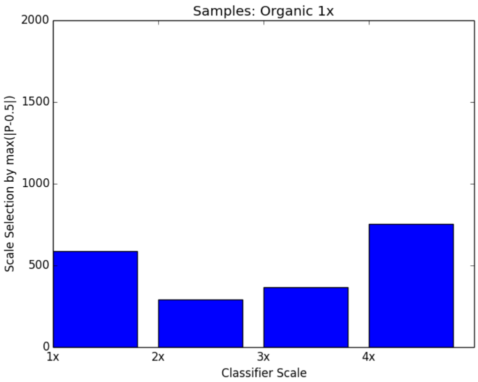


(c) Recognition Accuracy: Sample_3x

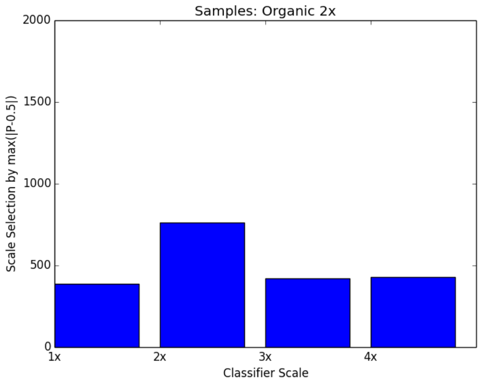


(d) Recognition Accuracy: Sample_4x

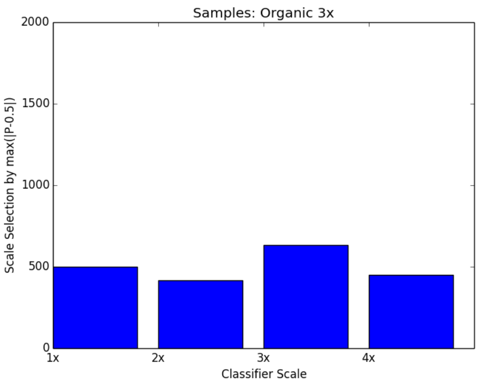
Figure 4.13: Recognition accuracy of 'Organic' samples with 4 scales and 3 approaches for scale selection



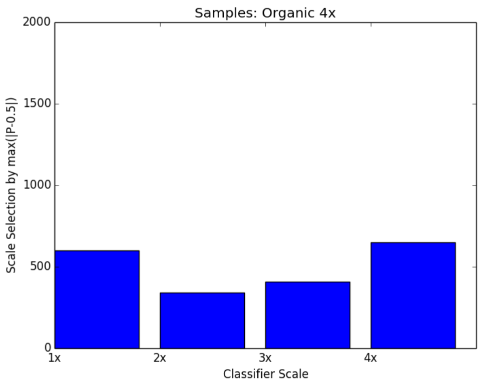
(a) Scale Selection: Sample_1x



(b) Scale Selection: Sample_2x

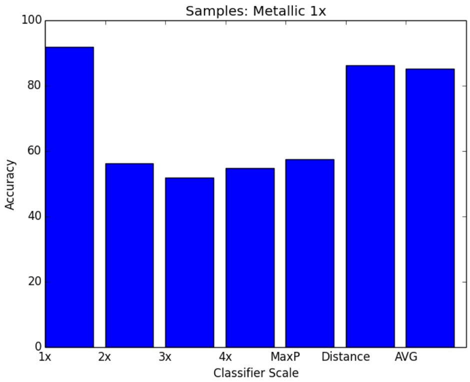


(c) Scale Selection: Sample_3x

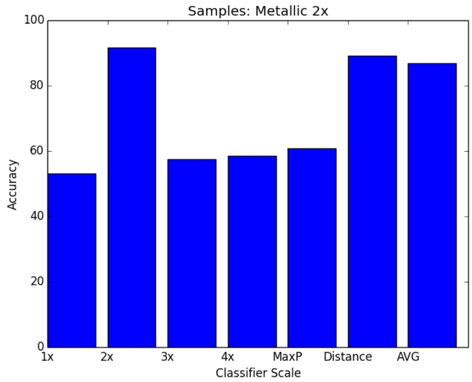


(d) Scale Selection: Sample_4x

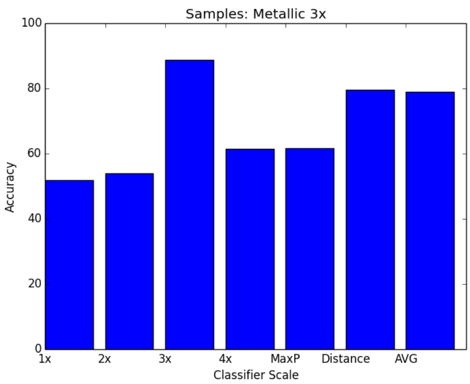
Figure 4.14: Scale Selection by Maximum Confidence of 'Organic' samples with 4 scales



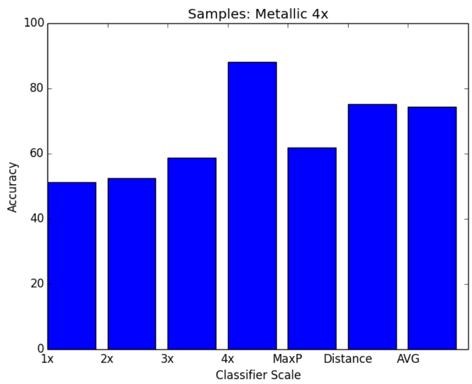
(a) Recognition Accuracy: Sample_1x



(b) Recognition Accuracy: Sample_2x

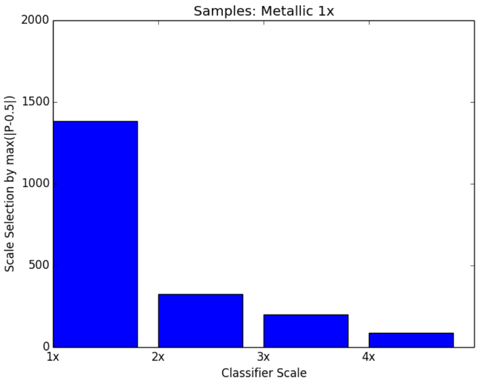


(c) Recognition Accuracy: Sample_3x

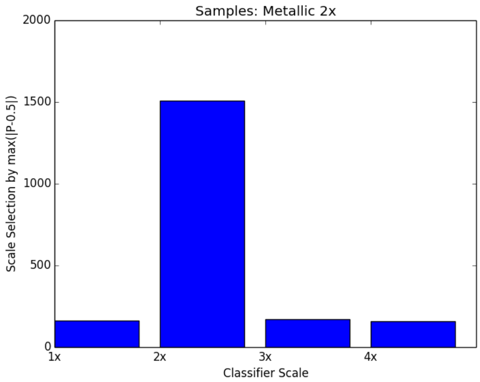


(d) Recognition Accuracy: Sample_4x

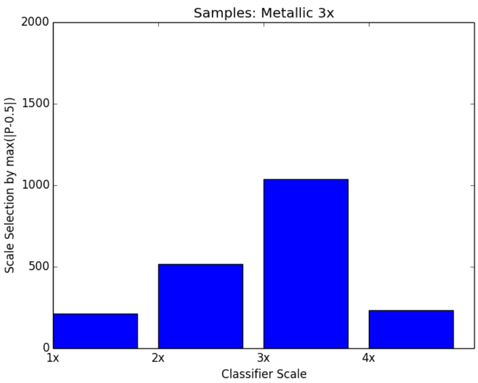
Figure 4.15: Recognition accuracy of 'Metallic' samples with 4 scales and 3 approaches for scale selection



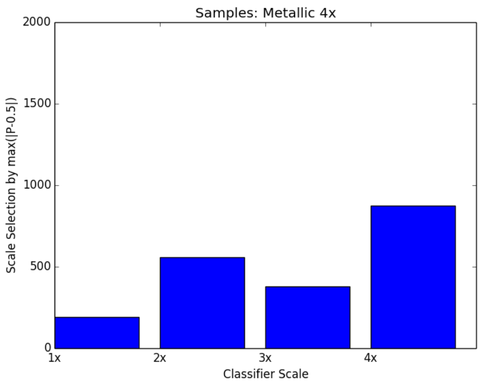
(a) Scale Selection: Sample_1x



(b) Scale Selection: Sample_2x

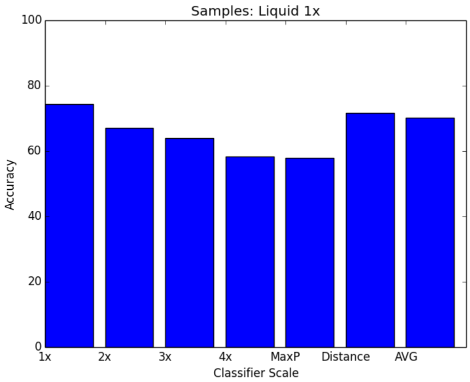


(c) Scale Selection: Sample_3x

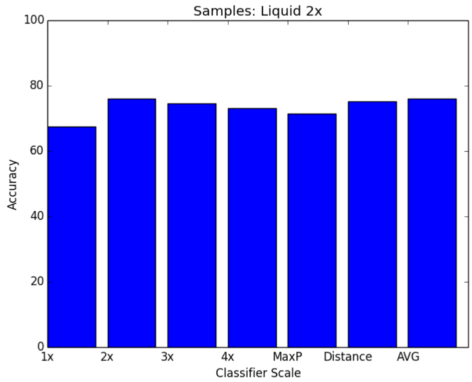


(d) Scale Selection: Sample_4x

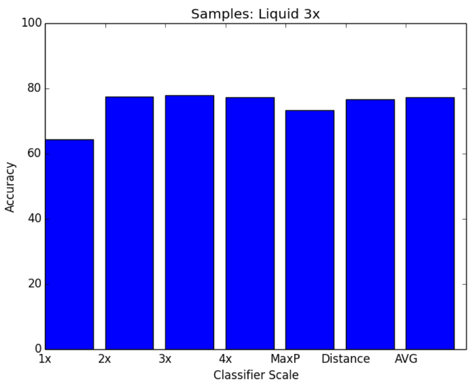
Figure 4.16: Scale Selection by Maximum Confidence of 'Metallic' samples with 4 scales



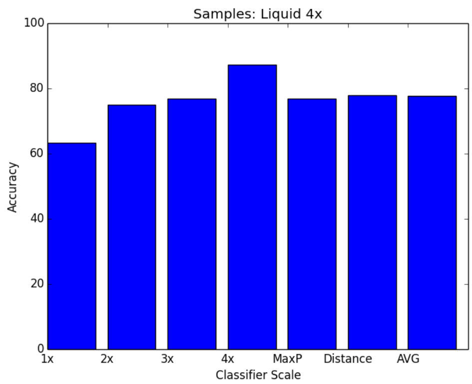
(a) Recognition Accuracy: Sample_1x



(b) Recognition Accuracy: Sample_2x

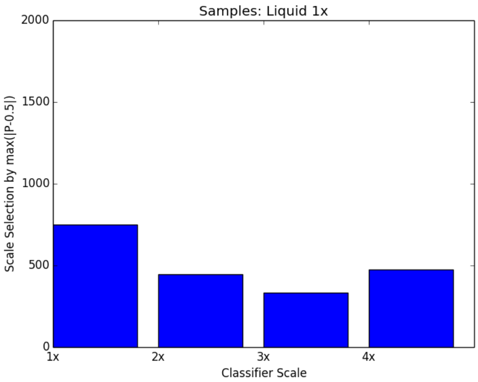


(c) Recognition Accuracy: Sample_3x

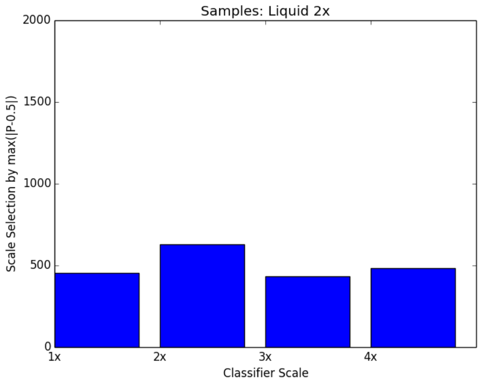


(d) Recognition Accuracy: Sample_4x

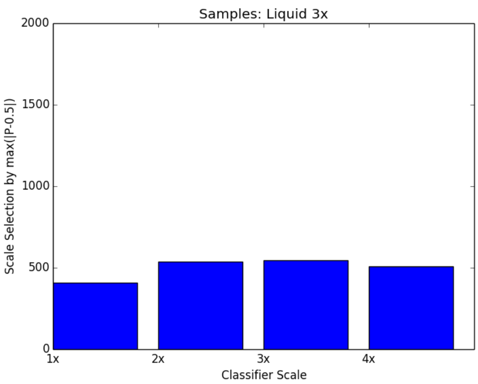
Figure 4.17: Recognition accuracy of 'Liquid' samples with 4 scales and 3 approaches for scale selection



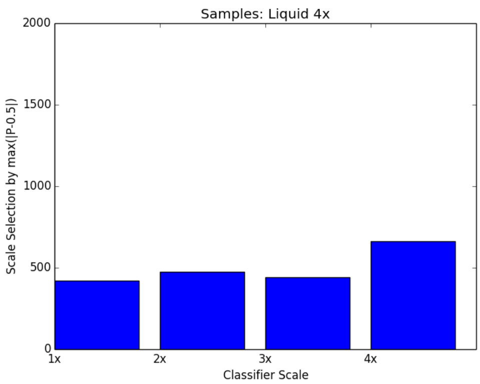
(a) Scale Selection: Sample_1x



(b) Scale Selection: Sample_2x

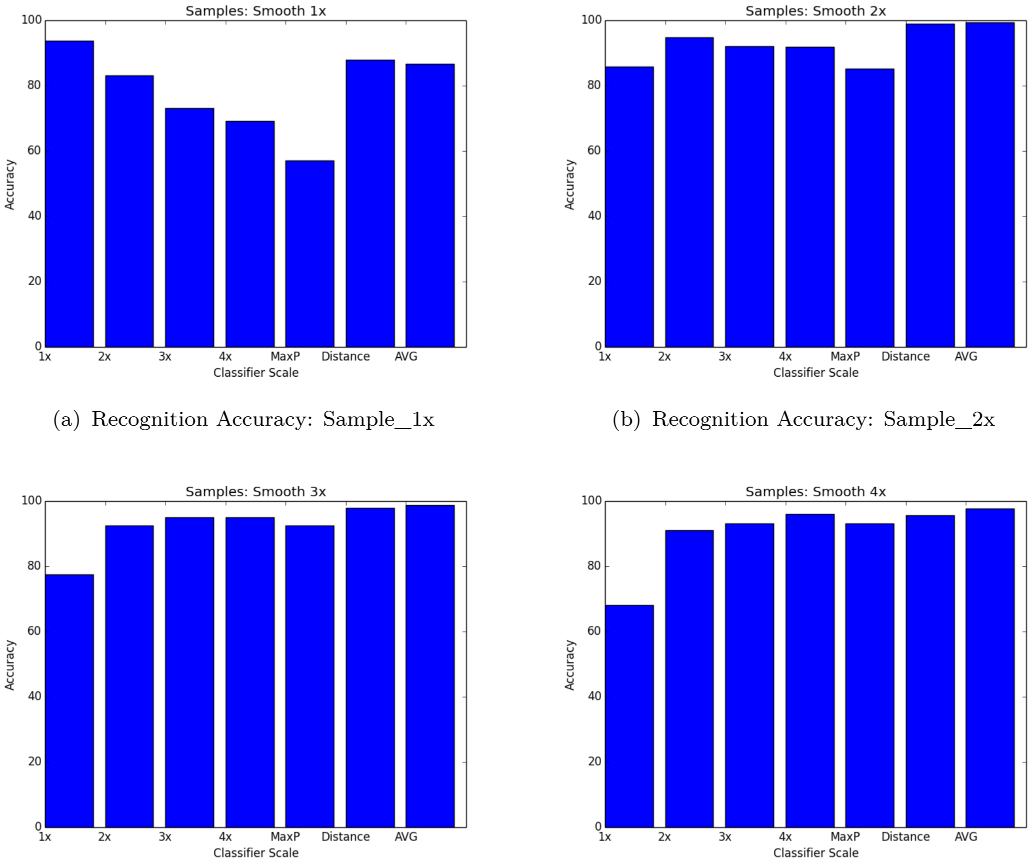


(c) Scale Selection: Sample_3x



(d) Scale Selection: Sample_4x

Figure 4.18: Scale Selection by Maximum Confidence of 'Liquid' samples with 4 scales



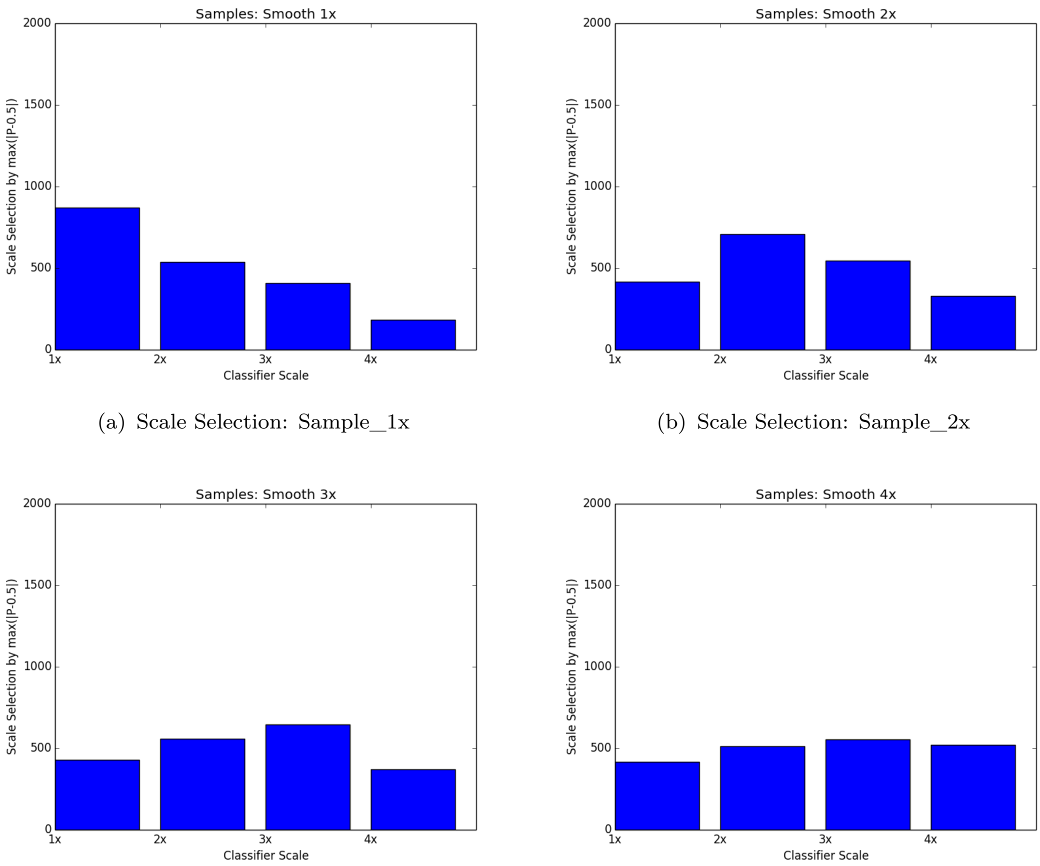
(a) Recognition Accuracy: Sample_1x

(b) Recognition Accuracy: Sample_2x

(c) Recognition Accuracy: Sample_3x

(d) Recognition Accuracy: Sample_4x

Figure 4.19: Recognition accuracy of 'Smooth' samples with 4 scales and 3 approaches for scale selection



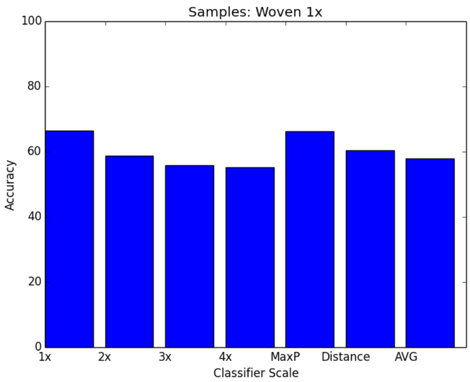
(a) Scale Selection: Sample_1x

(b) Scale Selection: Sample_2x

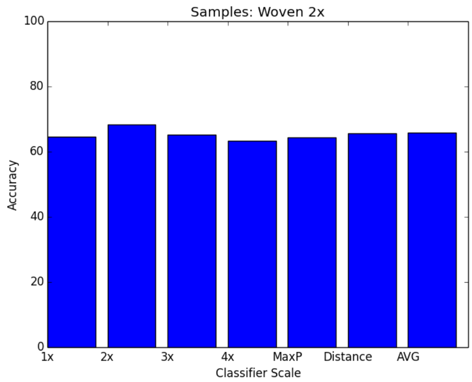
(c) Scale Selection: Sample_3x

(d) Scale Selection: Sample_4x

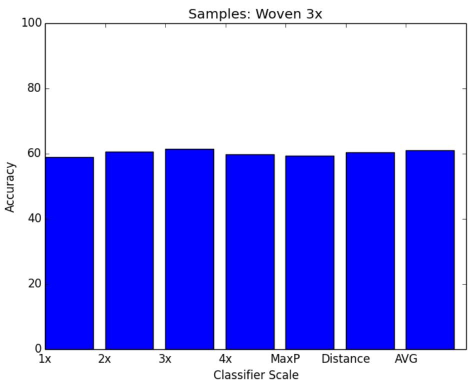
Figure 4.20: Scale Selection by Maximum Confidence of 'Smooth' samples with 4 scales



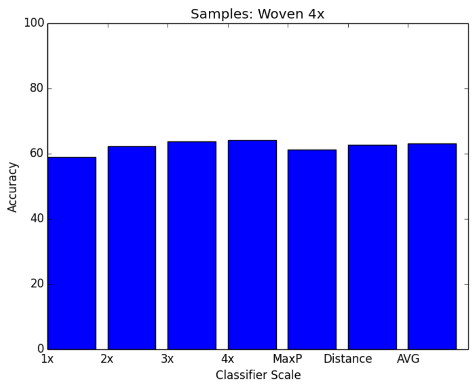
(a) Recognition Accuracy: Sample_1x



(b) Recognition Accuracy: Sample_2x

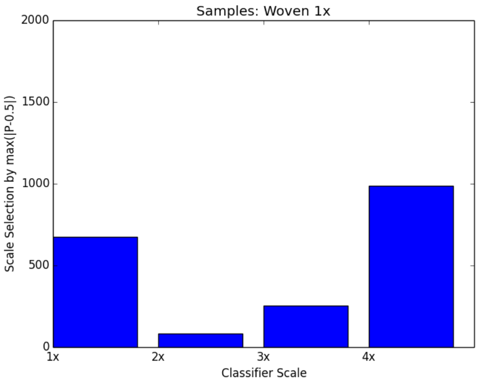


(c) Recognition Accuracy: Sample_3x

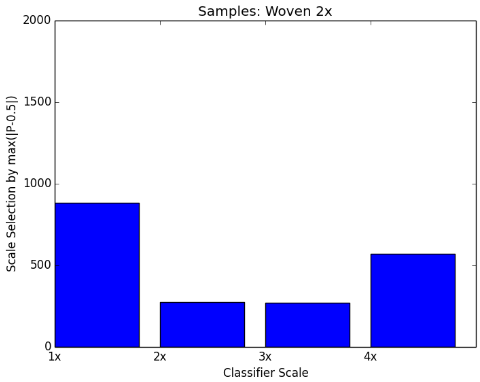


(d) Recognition Accuracy: Sample_4x

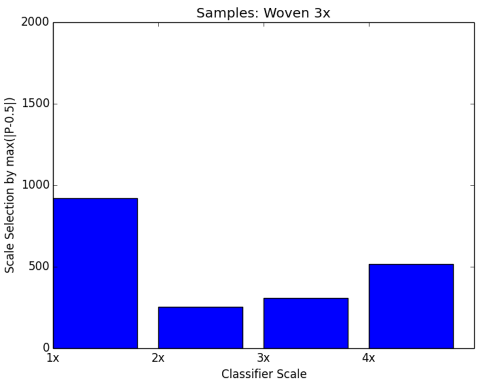
Figure 4.21: Recognition accuracy of 'Woven' samples with 4 scales and 3 approaches for scale selection



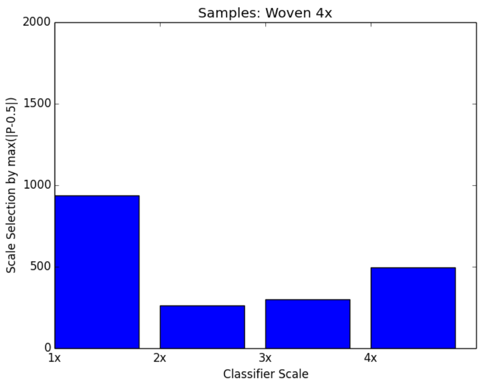
(a) Scale Selection: Sample_1x



(b) Scale Selection: Sample_2x



(c) Scale Selection: Sample_3x



(d) Scale Selection: Sample_4x

Figure 4.22: Scale Selection by Maximum Confidence of 'Woven' samples with 4 scales

Chapter 5

Summary

In [7], Schwartz *et al.*had proposed an approach to recognize materials by utilizing material traits (e.g. smooth, shiny, organic). This is quite different from methods in conventional research that employ some low-level features to do the recognition. Actually, recognizing materials with perceptual properties as an intermediate-level feature, is much more similar to the way human beings recognize material. They experiment results also show that it is reliable to recognize materials by learning material traits. However, in their research, scale change in images, which has an very important impact on the accuracy, has not been taken into consideration.

In this thesis, we propose an approach to recognize multi-scale material traits. We firstly train Convolutional Auto-encoder (CAE) Filters, which can represent the features well under an unsupervised learning, for material image patches with S scales. These filters will later been proven to be very important in our classifiers. We also prepare annotated material trait patches with 4 different scales, including patches with a particular material trait(e.g. smooth, metallic, organic) present or not present. For each trait, we take balanced sets of positive and negative examples to train multi-scale classifiers. For feature generating, we take pooled convolutional responses of prepared positive and negative trait patches with trained CAE filters at each scale. Besides we also employ some non-linear features (e.g. HOG, LBP, Color Histogram), and generate them with CAE filter responses as a huge Feature Matrix. We utilize feature matrices obtained at S different scales to train S Randomized Decision Forest Classifiers. With these classifiers, material traits with different scales can be directly recognized. Finally we propose two effective and stable approaches of scale selection for these classifiers to make a final decision of the testing sample's class label: One is searching maximum confidence among the predictions given by multi-scale classifiers, and the other one is computing the average prediction. Both these two approaches are proved to have a positive contribution to recognition accuracy.

Our results show that scale change has a very important impact on the accuracy when we recognize material traits. If the scale of object samples can be correctly selected, accuracy of recognition will have an obvious improvement. However, until now we have just found this interesting phenomenon and bring some assumptions to explain the results. Searching the inner relationship between different scales within the same material trait should be an promising topic in the future work. Besides, scale selection strategy with a better performance should be another feasible direction of this work in the future.

Bibliography

[1] L. Sharan, C. Liu, R. Rosenholtz, and E. H. Adelson, "Recognizing materials using perceptually inspired features," *International journal of computer vision*, vol. 103, no. 3, pp. 348–371, 2013.

[2] O. Wang, P. Gunawardane, S. Scher, and J. Davis, "Material classification using brdf slices," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on.* IEEE, 2009, pp. 2805–2811.

[3] M. Varma and A. Zisserman, "A statistical approach to texture classification from single images," *International Journal of Computer Vision*, vol. 62, no. 1-2, pp. 61–81, 2005.

[4] J. Shotton, M. Johnson, and R. Cipolla, "Semantic texton forests for image categorization and segmentation," in *Computer vision and pattern recognition, 2008. CVPR 2008. IEEE Conference on.* IEEE, 2008, pp. 1–8.

[5] Y. Kang and S. Akihiro, "Texton clustering for local classification using scene-context scale," in *Frontiers of Computer Vision,(FCV), 2013 19th Korea-Japan Joint Workshop on.* IEEE, 2013, pp. 26–30.

[6] R. W. Fleming, C. Wiebel, and K. Gegenfurtner, "Perceptual qualities and material classes," *Journal of vision*, vol. 13, no. 8, p. 9, 2013.

[7] G. Schwartz and K. Nishino, "Visual material traits: Recognizing per-pixel material context," in *Computer Vision Workshops (ICCVW), 2013 IEEE International Conference on.* IEEE, 2013, pp. 883–890.

[8] E. H. Adelson, "On seeing stuff: the perception of materials by humans and machines," in *Photonics West 2001-Electronic Imaging.* International Society for Optics and Photonics, 2001, pp. 1–12.

[9] C. Liu, G. Yang, and J. Gu, "Learning discriminative illumination and filters for raw material classification with optimal projections of bidirectional texture functions," in *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on.* IEEE, 2013, pp. 1430–1437.

[10] R. Shiradkar, L. Shen, G. Landon, S. H. Ong, and P. Tan, "A new perspective on material classification and ink identification," in *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on.* IEEE, 2014, pp. 2275–2282.

[11] S. Lombardi and K. Nishino, "Single image multimaterial estimation," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 238–245.

[12] M. Varma and A. Zisserman, "A statistical approach to material classification using image patch exemplars," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 31, no. 11, pp. 2032–2047, 2009.

[13] Y. Yacoob, "Matching dry to wet materials," in *Computer Vision (ICCV), 2013 IEEE International Conference on*. IEEE, 2013, pp. 2952–2959.

[14] W. Li and M. Fritz, "Recognizing materials from virtual examples," in *Computer Vision–ECCV 2012*. Springer, 2012, pp. 345–358.

[15] M. Weinmann, J. Gall, and R. Klein, "Material classification based on training data synthesized using a btf database," in *Computer Vision–ECCV 2014*. Springer, 2014, pp. 156–171.

[16] T. Kadir and M. Brady, "Saliency, scale and image description," *International Journal of Computer Vision*, vol. 45, no. 2, pp. 83–105, 2001.

[17] W. Li, "Learning multi-scale representations for material classification," in *Pattern Recognition*. Springer, 2014, pp. 757–764.

[18] L. Sharan, R. Rosenholtz, and E. Adelson, "Material perception: What can you see in a brief glance?" *Journal of Vision*, vol. 9, no. 8, pp. 784–784, 2009.

[19] C. Liu, L. Sharan, E. H. Adelson, and R. Rosenholtz, "Exploring features in a bayesian framework for material recognition," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2010, pp. 239–246.

[20] K. J. Dana, B. Van Ginneken, S. K. Nayar, and J. J. Koenderink, "Reflectance and texture of real-world surfaces," *ACM Transactions on Graphics (TOG)*, vol. 18, no. 1, pp. 1–34, 1999.

[21] J. Masci, U. Meier, D. Cireşan, and J. Schmidhuber, "Stacked convolutional auto-encoders for hierarchical feature extraction," in *Artificial Neural Networks and Machine Learning–ICANN 2011*. Springer, 2011, pp. 52–59.

[22] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[23] A. J. Bell and T. J. Sejnowski, "Edges are the" independent components" of natural scenes," in *NIPS*, 1996, pp. 831–837.

[24] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1. IEEE, 2005, pp. 886–893.

[25] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 24, no. 7, pp. 971–987, 2002.

[26] S. Blunsden and L. Atallah, "Investigating the effects of scale in mrf texture classification," 2005.

[27] D. Hu, L. Bo, and X. Ren, "Toward robust material recognition for everyday objects." in *BMVC*, vol. 13. Citeseer, 2011, p. 14.

[28] J. Gu and C. Liu, "Discriminative illumination: Per-pixel classification of raw materials based on optimal projections of spectral brdf," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on.* IEEE, 2012, pp. 797–804.