REFLECTANCE AND ANGULAR LUMINANCE FOR MATERIAL RECOGNITION AND SEGMENTATION

by

JIA XUE

A dissertation submitted to the

School of Graduate Studies

Rutgers, The State University of New Jersey

In partial fulfillment of the requirements

For the degree of

Doctor of Philosophy

Graduate Program in Electrical and Computer Engineering

Written under the direction of

Kristin J. Dana

And approved by

New Brunswick, New Jersey

JANUARY, 2020

ABSTRACT OF THE DISSERTATION

Reflectance and Angular Luminance for Material Recognition and Segmentation

By JIA XUE

Dissertation Director:

Kristin J. Dana

Real world scenes consist of surfaces made of numerous materials, such as wood, marble, dirt, metal, ceramic and fabric, which contribute to the rich visual variation we find in images. Materials play a fundamental role in numerous applications including asphalt for automated driving, tree-cover in fire risk assessment, path material (grass vs. concrete) for robot navigation, and landcover albedo analysis for climate studies. This thesis is dedicated to developing compact and robust material and texture representations for material recognition and segmentation.

Material properties affect the spatial variation of surface appearance and the angular variation of reflectance with respect to both view and illumination. Modeling the apparent or latent characteristic appearance of different materials is essential to robustly recognize them in images. We build representations that capture the intrinsic invariant properties of the surface, which enables fine-grained material recognition and segmentation. In particular, this thesis develop the following methods:

1. Differential Angular Imaging: We present a new measurement method called differential angular imaging where a surface is imaged from a particular viewing angle v and then from an additional viewpoint $v + \delta$. The motivation for

this differential change in viewpoint is improved computation of the angular gradient of intensity $\partial I_v/\partial v$. We develop a framework for differential angular imaging, where small angular variations in image capture provide an enhanced appearance representation and significant recognition improvement. We build a large-scale material database, Ground Terrain in Outdoor Scenes (GTOS) database, geared towards real use for autonomous agents. The database consists of over 30,000 images covering 40 classes of outdoor ground terrain under varying whether and lighting conditions.

2. Deep Texture Manifold: For ground terrain recognition, many class boundaries are ambiguous. Therefore, it is of interest to find not only the class label but also the closest classes, or equivalently, the position in the manifold. We present a texture network called Deep Encoding Pooling Network (DEP) for the task of ground terrain recognition. Recognition of ground terrain is an important task in establishing robot or vehicular control parameters, as well as for localization within an outdoor environment. The architecture of DEP integrates orderless texture details and local spatial information. The resultant network shows excellent performance not only for GTOS-mobile, but also for more general databases (MINC and DTD). Based on DEP, we introduce a new texture manifold method, DEP-manifold, to find the relationship between newly captured images and images in dataset.

3. Texture Encoded Angular Network: We develop a novel approach for material recognition called texture-encoded angular network (TEAN) that combines deep encoding pooling of RGB information and differential angular images for angular-gradient features for the task of ground terrain recognition. With this novel network architecture, we extract characteristics of materials encoded in the angular and spatial gradients of their appearance. Our results show that TEAN achieves recognition performance that surpasses single view performance and standard (non-differential/large-angle sampling) multiview performance.

4. Angular Luminance: We utilize per-pixel *angular luminance distributions* as a key feature in discriminating the material of the surface. The angle-space sampling in a multiview image sequence is an unstructured sampling of the underlying reflectance function of the material. For real-world materials there is significant intra-class variation that can be managed by building a Angular Luminance Network (AngLNet). This network combines new angular reflectance cues from multiple images with more traditional spatial cues as in fully convolutional networks for semantic segmentation. We demonstrate the increased performance of AngLNet over prior state-of-the-art in material segmentation from drone video sequences and satellite imagery.

Acknowledgements

The past five years at Rutgers has been an unforgettable and invaluable experience for me. First of all, I would like to express my greatest gratitude to my advisor Kristin Dana. Kristin is an extremely kind, caring and supportive advisor that I could not have asked for more. I started to know Kristin through her robotic vision class in 2014, she brought me to the challenging and promising fields of computer vision, computational photography and artificial intelligence. I still remember the time when I started at Rutgers, I could barely speak fluent English, and knew little about this country. Kristin developed my skills in research, writing, and communication. I feel greatly fortunate to be her student.

It is my great honor to have Jorge Ortiz, Salim Rouayheb and Szymon Rusinkiewicz on my thesis committee. The work presented in this dissertation is very relevant to their research and I learned a lot from their works.

During my PhD, I have done three wonderful internships at Philips Research, Alibaba Group and Apple Inc. I thank my collaborators Zibo Meng, Karthik Katipally, Haibo Wang, Kees van Zon, Xiaofeng Ren, Feng Li and Jianping Zhou when I worked at these places.

Many thanks also go to my colleagues in Rutgers University, including Hang Zhang, Eric Wengrowski, Matthew Purri, Parneet Kaur, Thomas Shyr, Peri Akiva, He Zhang, Hansi Liu, Li Zhu, Zhenhua Jia, Jing Zhong, Yi Han, Xin Dong, Jian Zhou and Pei Peng. I learned a lot in many aspects from them during these years, I would never forget the enriched and joyful days together.

Lastly, a special thanks to my wife Lilin Zhang and my parents Changnian Xue and Guijie Guan. They made me who I am today and I never know how to pay them back. I hope that they are at least a little proud of me for what I have been through so far. Finally, I would like to thank all staffs in the Electrical and Computer Engineer Department at Rutgers University and all the other people who helped and discussed the thesis.

Dedication

To my family, for their unconditional love.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

Real world scenes consist of surfaces made of numerous materials, such as wood, marble, dirt, metal, ceramic and fabric, which contribute to the rich visual variation we find in images. Different materials are composed of different textures, the spatial arrangement of lines and colors gives appealing perceptual impression for humans. In computer vision, materials are interesting not only because of their perceptual response, but also because algorithms can characterize and quantify materials in novel and useful ways. For example, image in-painting use surrounding materials to generate materials of the cropped region; image compression uses patterns to efficiently summarize extensive content and robots uses materials to estimate the friction of surfaces and rigidity of objects.

The challenge of recognizing materials arises due to variation in material appearance for different contexts in which materials appear. The same material may appear as part of different objects with different colors, conversely, different materials may appear with similar color and similar objects. For example, as shown in Figure 1.1, from left to right, the first and second bottles are both made with plastic, but present with different color and different transparency. The cover of the third bottle is made of plastic, but the body part is made of steel. Modeling the apparent or latent characteristic appearance of different materials is essential for robust material recognition. Taking into account the list of desired material attributes, as discussed in [7], a set of material model requirements/guidelines is listed as follows.

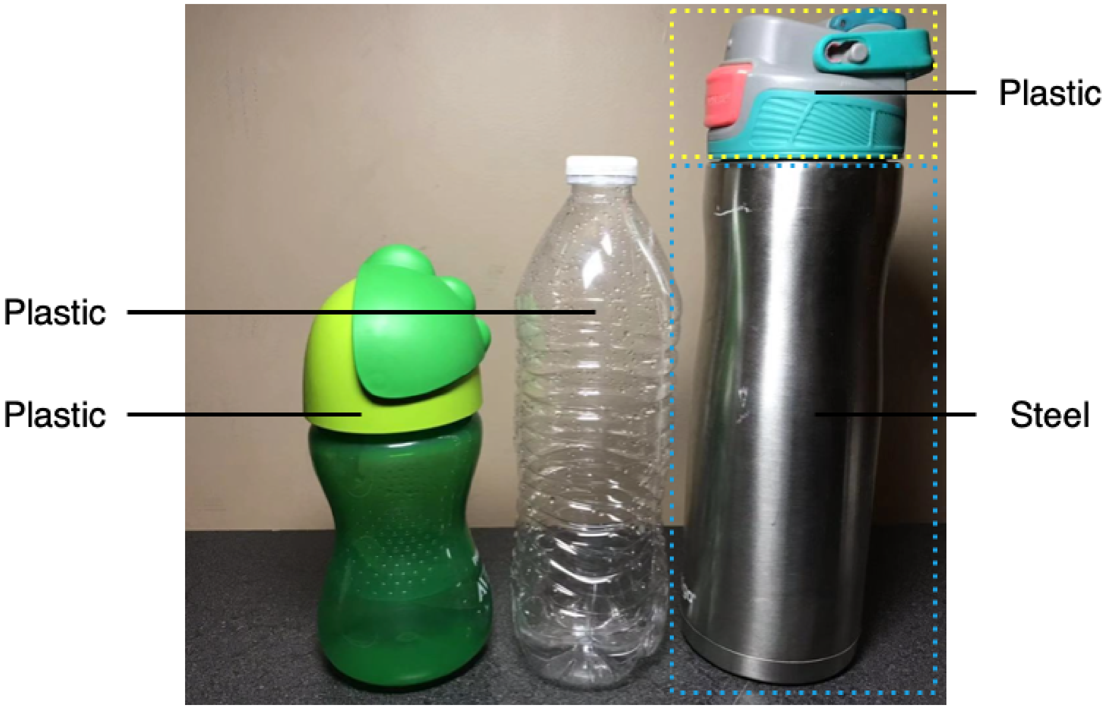


Figure 1.1: The same material may appear as part of different objects with different colors, and in turn different materials may appear with similar color and on the same objects. From left to right, the first and second bottles are both made with plastic, but present with different color and different transparency. The cover of the third bottle is made of plastic, but the body is made by steel.

- **Intraclass Invariance** — All examples of the same material class should give rise to similar representation values.

- **Illumination and Geometric Invariance** — The camera pose and environment illumination is arbitrary in real-world imaging conditions. The material model shall be robust to real-world imaging conditions.

A material model that exhibits the requirements listed above is required to recognize materials in real world.

1.2 Background

1.2.1 Material Dataset

Material properties affect the spatial variation of surface appearance and the angular variation of reflectance with respect to both view and illumination. Multiview observations of a scene provide an opportunity to use reflectance for recognition, since aligned imagery can provide a vector of reflectance samples per pixel.

Early studies of material appearance modeling largely concentrated on comprehensive lab-based measurements using dome systems, robots, or gonioreflectometers collecting measurements that are dense in angular space (such as BRDF, BTF) as summarized in a recent survey [8]. Wang *et al.* [9] constructed a hemispherical dome encircling the target object. A camera at the center of the dome records many images, each lit by a different lamp. This produces an image stack, where each pixel collects information for several lighting conditions for each color channel. Liu*et al.* [10] design and build an LED-based multi- spectral dome light for classifying raw materials based on a 2D slice of their spectral BRDFs. The dome has 25 LED clusters. Each LED cluster has six color LEDs which can be weighted individually to create a desired spectrum. They learn optimal illumination patterns from training samples, after projecting to which the spectral BRDFs

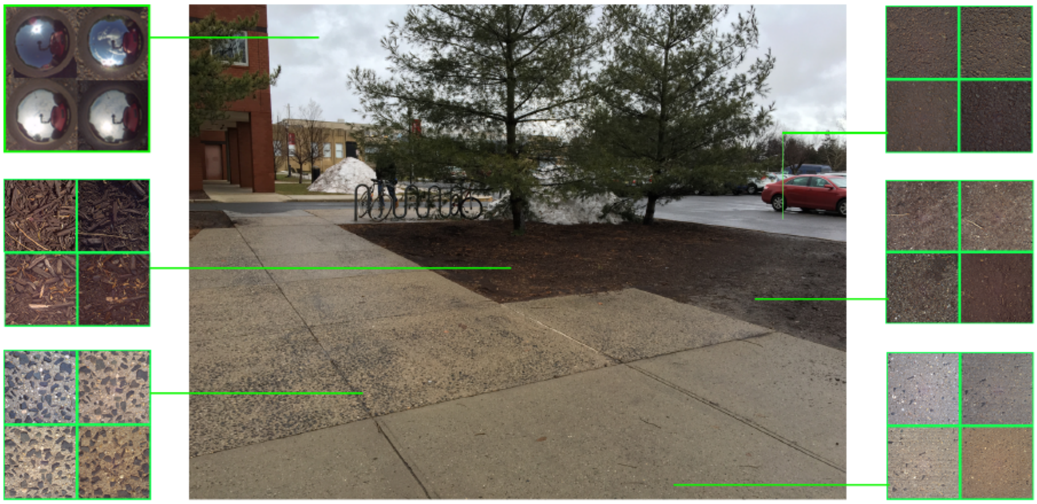


Figure 1.2: Example from GTOS dataset comprising outdoor measurements with multiple viewpoints, illumination conditions and angular differential imaging. The example shows scene-surfaces imaged at different illumination/weather conditions.

of different materials can be maximally separated. These reflectance-based studies have the advantage of capturing intrinsic invariant properties of the surface, which enables fine-grained material recognition.

Material recognition and segmentation for real-world outdoor surfaces has become increasingly important for artificial intelligence and computer vision to support its operation "in the wild." For instance, for applications of automated driving, robotics and human-computer interaction. Lab-based approaches to reflectance measurement can be cumbersome, time-consuming, and nonportable. Capturing a full BRDF is rarely practical in applications, the inflexibility of lab-based image capture, however, prevents widespread use in real world scenes, especially in the important class of outdoor scenes.

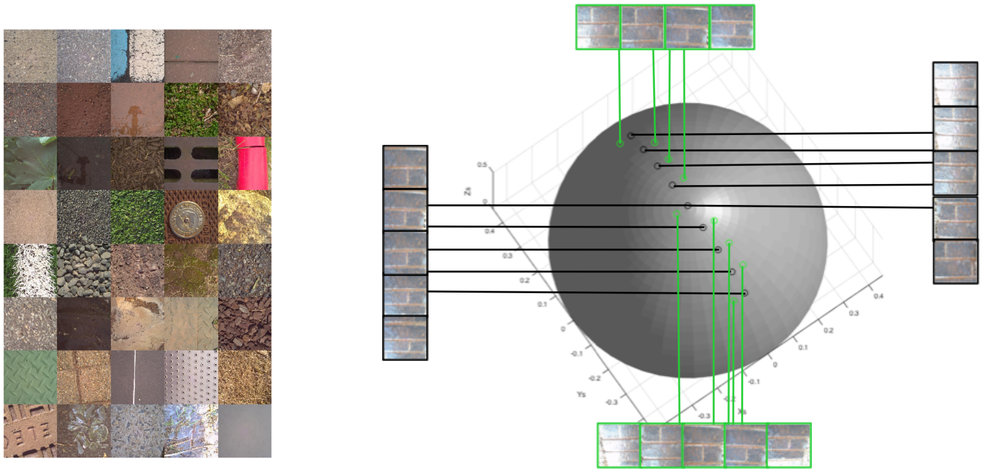
A fundamentally different approach to reflectance modeling is image-based appearance modeling where surfaces are captured with an single-view image in-scene or “in-the-wild.” Recent studies of image-based material recognition use single-view internet-mined images to train classifiers [11, 12, 13, 14] and can be applied to arbitrary images casually taken without the need of multiview reflectance information. For example, Cimpoi *et al.*[13] introduce the Describable Textures Dataset (DTD), which consists of 5,640 texture images jointly annotated with the 47 attributes. All of their images are captured in the wild by downloading them from the Internet rather than collecting them in a laboratory. In these methods, however, recognition is typically based more on context including object and scene cues, than intrinsic material appearance properties.

Between the two approaches of reflectance-based and image-based material recognition, i.e. between comprehensive in-lab imaging and internet-mined images, we take an advantageous middle-ground. Specifically, we capture in-scene real-world surfaces but use multiple viewpoint angles for measurements that provide a partial reflectance sampling.

1.2.2 Material Recognition and Segmentation

In classic approaches for texture modeling, images are filtered with a set of hand-crafted filter banks followed by grouping the outputs into texton histograms [15, 16, 17, 18], or bag-of-words representation [19, 20]. For example, Zhang *et al.*[21] encode the discriminative optical characteristics of materials captured in the reflectance disks with a texton-based representation for material Recognition.

The success of deep learning methods in object recognition has also translated to the problem of material recognition, the classification and segmentation of material categories in arbitrary images. Bell *et al.*achieve per-pixel material category labeling by retraining the then state-of-the-art object recognition network [22] on a large dataset of material appearance [11]. Cimpoi *et al.*[23] achieves state-of-art results on FMD [24] and KTH-TIPS2 [25] using a Fisher vector representation computed on image features extracted with a CNN. Deep-TEN [4] ports the dictionary learning and feature pooling approaches into the CNN pipeline for an



(a) material classes

(b) one sample at multiple viewing directions

Figure 1.3: (a) The 40 material categories in the GTOS dataset introduced in this paper. (Right) The material surface observation points. Nine viewpoint angles separated along an arc spanning 80° are measured. For each viewpoint, a differential view is captured ±5° in azimuth from the original orientation (the sign is chosen based on robotic arm kinematics.)

end-to-end material/texture recognition network. Recognition algorithms that focus on texture details work well for images containing only a single material. But for "images in the wild", homogeneous surfaces rarely fill the entire field-of-view, and many materials exhibit regular structure. For material recognition, since surfaces are not completely orderless, local spatial order is an important cue for recognition.

In the field of material recognition and computational photography, one goal is to develop a ground terrain recognition system which can recognize ground terrain surfaces and find the relationship between newly captured material images and the corresponding images in the material dataset. For ground terrain recognition, many class boundaries are ambiguous. For example, "asphalt" class is similar to "stone-asphalt" which is an aggregate mix of stone and asphalt. The class "leaves" is similar to "grass" because most of the example images for "leaves" have grass

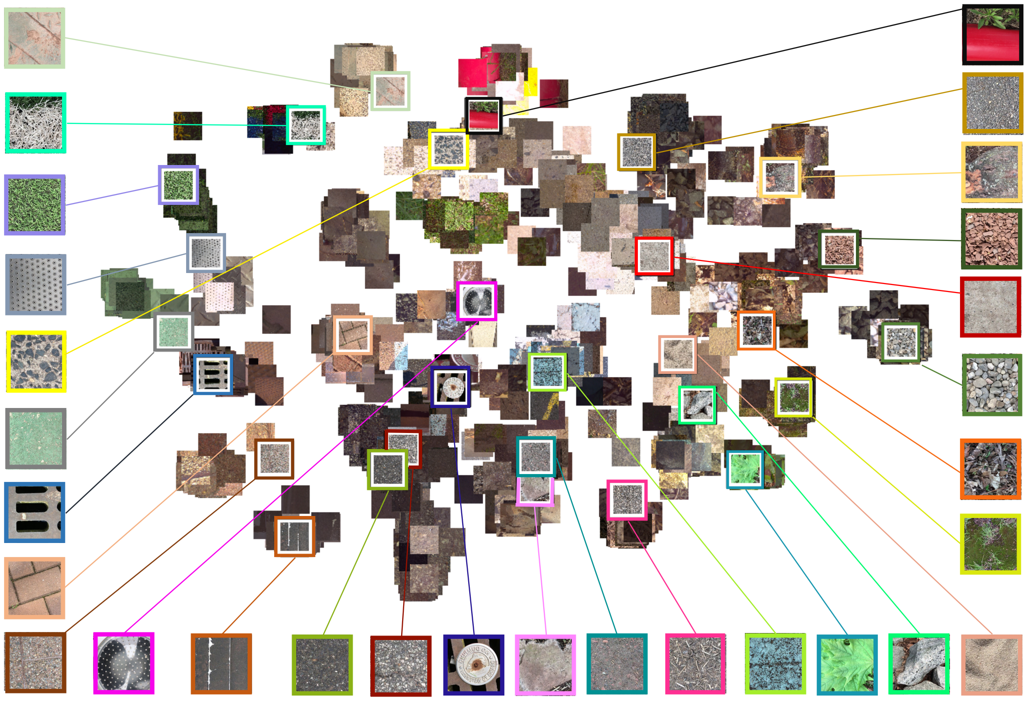


Figure 1.4: The result of texture manifold by DEP-manifold. Images with color frames are images in test set. The material classes are (from upper left to counter-clockwise): plastic cover, painted turf, turf, steel, stone-cement, painted cover, metal cover, brick, stone-brick, glass, sandpaper, asphalt, stone-asphalt, aluminum, paper, soil, mulch, painted asphalt, leaves, limestone, sand, moss, dry leaves, pebbles, cement, shale, roots, gravel and plastic. Not all classes are shown here for space limitations.

in the background. Similarly, the grass images contain a few leaves. Therefore, it is of interest to find not only the class label but also the closest classes, or equivalently, the position in the manifold.

Early studies in material recognition from reflectance characteristics largely concentrated on per-image recognition, which predicts one material class for the entire image or region. But pixel-wise prediction is required for segmentation, and characterizing material appearance with a BRDF for each pixel is an insurmountable task. Also, in many cases, we lack sufficient training data to formulate a probability distribution over the entire space of realizable BRDFs that fully capture the intraclass variation. The approach by Dror *et al.*[26] asserts that given a finite but arbitrary set of candidate reflectance functions, we can identify which one most closely represents an observed surface by a low-cost intensity distribution. These luminance histograms computed over a spatial region are powerful for material discrimination [27]. Integrating luminance histograms with color images can be helpful for material segmentation.

1.3 Thesis Overview

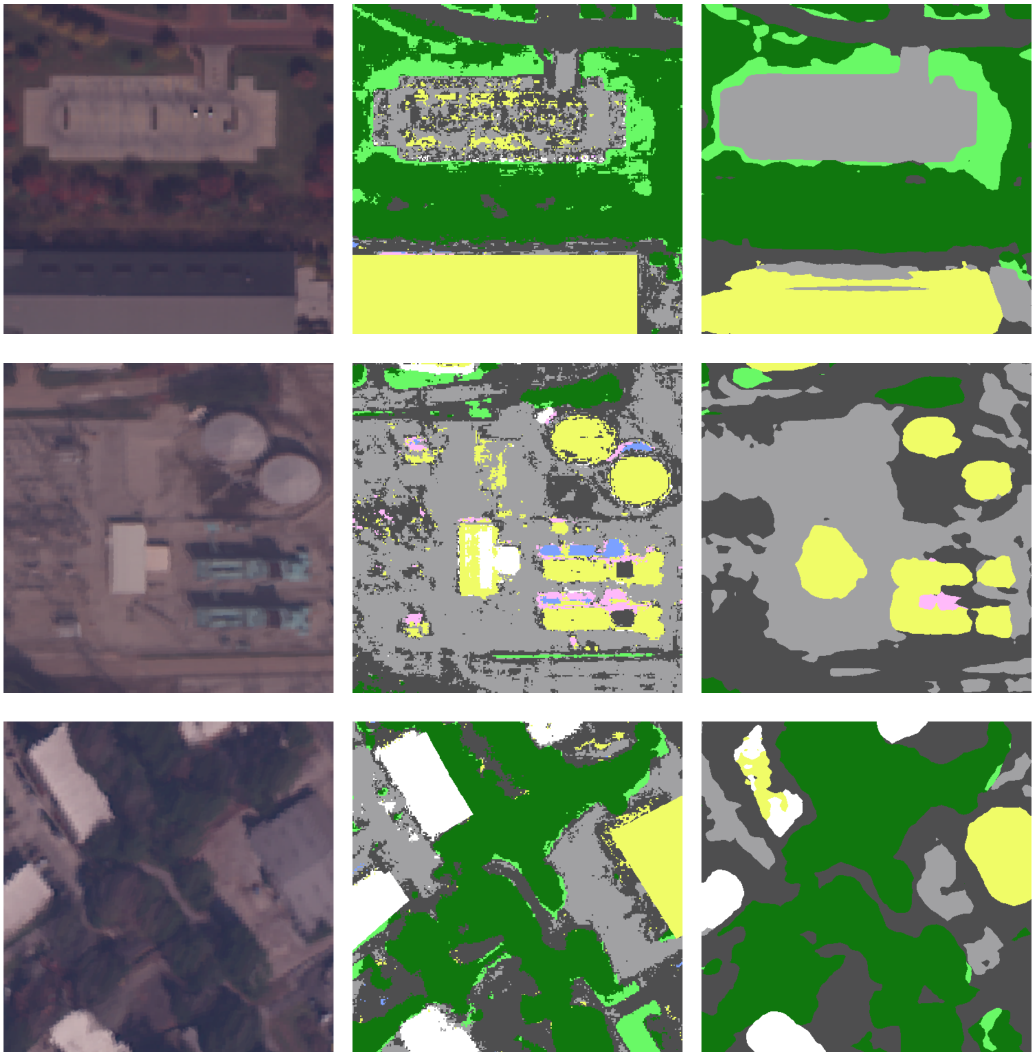
In this thesis, we are interested in using multiview observations for angular and spatial models of reflectance. For angular variations, we capture in-scene real-world surfaces with multiple viewpoint angles for measurements that provide a partial reflectance sampling. This leads to a very basic question: how do multiple viewing angles help in material recognition? More interestingly, we consider a novel question: Do small changes in viewing angles, *differential changes*, result in significant increases in recognition performance?

In Chapter 2, we present an approach called *angular differential imaging* that augments image capture for a particular viewing angle v a differential viewpoint $v + \delta$. Contrast this method, with lab-based reflectance measurements that often quantize the angular space measuring with domes or positioning devices with large angular spacing such as 22.5°. To capture material appearance in a manner that preserves the convenience of image-based methods but important angular information of reflectance-based methods, as shown in Figure 1.2 and Figure 1.3, we assemble a comprehensive, first-of-its-kind, *outdoor* material database that includes multiple viewpoints and multiple illumination directions (partial BRDF sampling), multiple weather conditions, a large set of surface material classes surpassing existing comparable datasets, multiple physical instances per surface class (to capture intra-class variability) and differential viewpoints to support the framework of differential angular imaging.

In Chapter 3, we separately address texture spatial variations by creating texture networks for material recognition (Deep-Ten). We explore combining both angular and spatial variations of reflectance in real world scenes. For ground terrain recognition, many class boundaries are ambiguous. For example, “asphalt” class is similar to “stone-asphalt” which is an aggregate mix of stone and asphalt. The class “leaves” is similar to “grass” because most of the example images for “leaves” in the GTOS database have grass in the background. Similarly, the grass images contain a few leaves. Therefore, it is of interest to find not only the class label but also the closest classes, or equivalently, the position in the manifold. As shown in Figure 1.4, we introduce a new texture manifold method, DEP-manifold, to find the relationship between newly captured images and images in dataset.

Adapting the DEP to the RGB image branch into the Differential Angular Imaging Network (DAIN), we introduce the Texture Encoded Angular Network (TEAN). We develop a two-stream convolutional neural network, one branch input is the differential angular image, representing the material reflectance information. The other branch input is the RGB image, representing the orderless texture details and ordered spatial information. For the color image branch, we utilize the Deep Encoding Pooling Network (DEP) to balance the orderless texture component and ordered spatial information. As in DAIN, we combine feature maps at both intermediate layer and final prediction layer. With the proposed Texture Encoded Angular Network (TEAN), we take advantage of material reflectance information, orderless texture details and ordered spatial information for ground terrain material recognition. This combination of angular cues, orderless spatial cues and ordered spatial cues leads to improved recognition results.

When adapting the multiple viewpoint angles techniques into large scale material segmentation as in Chapter 4, characterizing material appearance with a



(a) Image (b) Ground Truth (c) AngNet

Figure 1.5: The material segmentation results of AngLNet on the satellite dataset.

BRDF for each pixel is an insurmountable task. Early studies in material recognition from reflectance characteristics largely concentrated on per-image recognition, which predicts one material class for the entire image or region. But pixel-wise prediction is required for segmentation, and characterizing material appearance with a BRDF for each pixel is an insurmountable task. Also, in many cases, we lack sufficient training data to formulate a probability distribution over the entire space of realizable BRDFs that fully capture the intraclass variation. we make use of a per-pixel *angular luminance histogram* representing the distribution of intensities observed per-pixel from all viewing angles in the multiview sequence. As shown in Figure 1.5, integrated in a meaningful way within a large deep network, the angular histogram cue consistently provides a signficant performance boost for material-based segmentation. Moreover, in applications where multiview images are collected, the angular histogram is readily available with little additional cost.

Chapter 2

Differential Angular Imaging for Material Recognition

This chapter on Differential Angular Imaging for Material Recognition is based on our paper [28]. We propose to take a middle-ground approach for material recognition that takes advantage of both rich radiometric cues and flexible image capture. We realize this by developing a framework for differential angular imaging, where small angular variations in image capture provide an enhanced appearance representation and significant recognition improvement. We build a large-scale material database, Ground Terrain in Outdoor Scenes (GTOS) database, geared towards real use for autonomous agents. The database consists of over 30,000 images covering 40 classes of outdoor ground terrain under varying whether and lighting conditions. We develop a novel approach for material recognition called a Differential Angular Imaging Network (DAIN) to fully leverage this large dataset. With this novel network architecture, we extract characteristics of materials encoded in the angular and spatial gradients of their appearance. Our results show that DAIN achieves recognition performance that surpasses single view or coarsely quantized multiview images. These results demonstrate the effectiveness of differential angular imaging as a means for flexible, in-place material recognition.

2.1 Background

Real world scenes consist of surfaces made of numerous materials, such as wood, marble, dirt, metal, ceramic and fabric, which contribute to the rich visual variation we find in images. Material recognition has become an active area of research

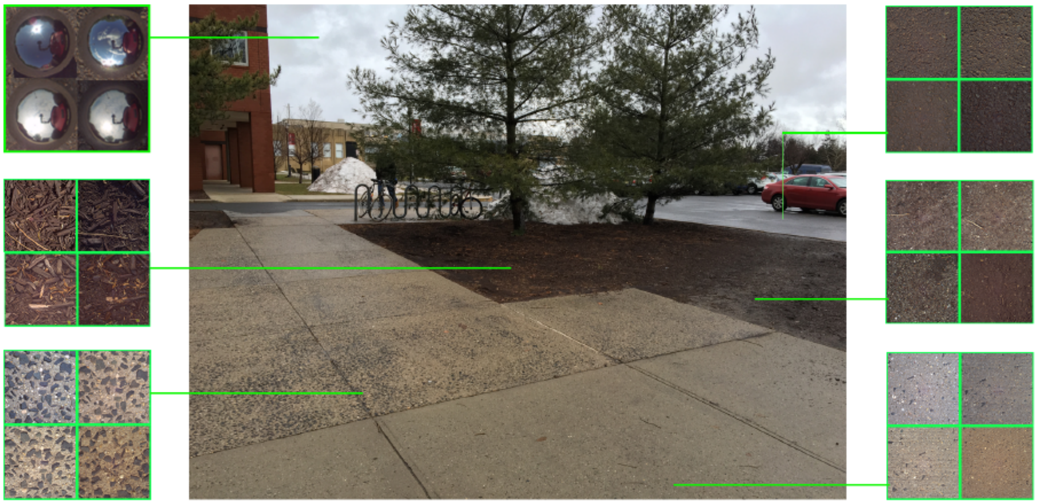
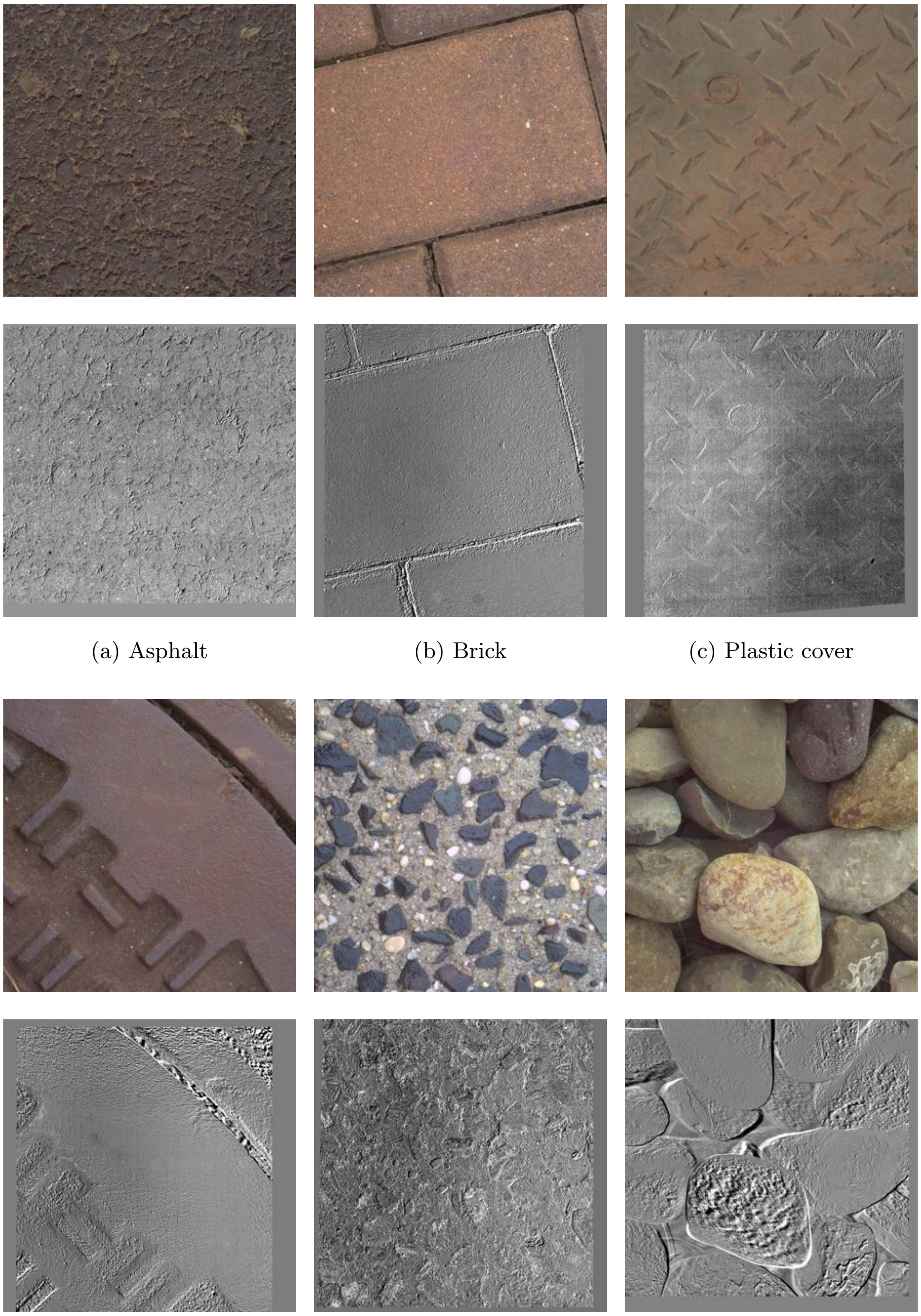


Figure 2.1: Example from GTOS dataset comprising outdoor measurements with multiple viewpoints, illumination conditions and angular differential imaging. The example shows scene-surfaces imaged at different illumination/weather conditions.

in recent years with the goal of providing detailed material information for applications such as autonomous agents and human-machine systems. Modeling the apparent or latent characteristic appearance of different materials is essential to robustly recognize them in images. Early studies of material appearance modeling largely concentrated on comprehensive lab-based measurements using dome systems, robots, or gonioreflectometers collecting measurements that are dense in angular space (such as BRDF, BTF). These reflectance-based studies have the advantage of capturing intrinsic invariant properties of the surface, which enables fine-grained material recognition [10, 29, 9]. The inflexibility of lab-based image capture, however, prevents widespread use in real world scenes, especially in the important class of outdoor scenes. A fundamentally different approach to reflectance modeling is image-based appearance modeling where surfaces are captured with an single-view image in-scene or “in-the-wild.” Recent studies of image-based material recognition use single-view internet-mined images to train



(a) Asphalt (b) Brick (c) Plastic cover

(a) Metal cover (b) Stone-cement (c) Pebble

Figure 2.2: Differential Angular Imaging. (Top) Examples of material surface images I_v. (Bottom) Corresponding differential images $I_\delta = I_v - I_{v+\delta}$ in our GTOS dataset. These sparse images encode angular gradients of reflection and 3D relief texture.

classifiers [11, 12, 23, 14] and can be applied to arbitrary images casually taken without the need of multiview reflectance information. In these methods, however, recognition is typically based more on context including object and scene cues, than intrinsic material appearance properties except for a few purely local methods [30, 31].

Between the two approaches of reflectance-based and image-based material recognition, i.e. between comprehensive in-lab imaging and internet-mined images, we take an advantageous middle-ground. Specifically, we capture in-scene real-world surfaces but use multiple viewpoint angles for measurements that provide a partial reflectance sampling. This leads to a very basic question: how do multiple viewing angles help in material recognition? More interestingly, we consider a novel question: Do small changes in viewing angles, *differential changes*, result in significant increases in recognition performance? Prior work has shown the power of angular filtering to complement spatial filtering in material recognition. These methods, however, relied on a mirror-based camera to capture a slice of the BRDF [32] or a lightfield camera to achieve multiple differential viewpoint variations [1] which limits their application in the wild due to its rigid imaging system setup and inadequacy for image capture at distance, respectively. We instead propose to capture surfaces with differential changes in viewing angles with an ordinary camera and compute discrete approximations of *angular gradients* from them. We present an approach called *angular differential imaging* that augments image capture for a particular viewing angle v a differential viewpoint $v + \delta$. Contrast this method, with lab-based reflectance measurements that often quantize the angular space measuring with domes or positioning devices with large angular spacing such as 22.5°. These coarse-quantized measurements have limited use in approximating angular gradients. Angular differential imaging can be implemented with a small-baseline stereo camera or a moving camera (e.g. handheld). We demonstrate that differential angular imaging provides key information about material reflectance properties while maintaining the flexibility of

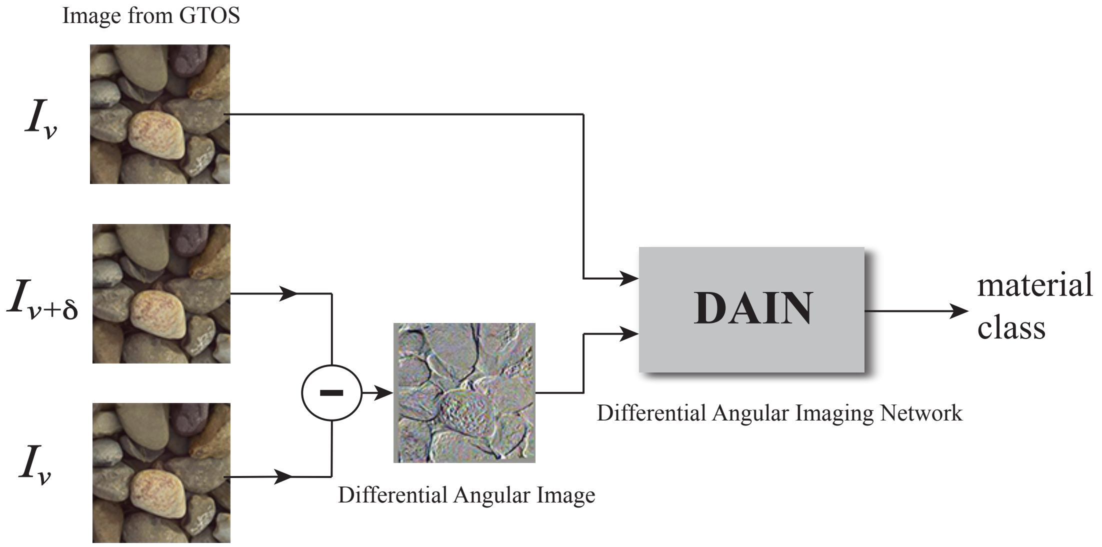


Image from GTOS
I_v
$I_{v+\delta}$
I_v
Differential Angular Image
DAIN
Differential Angular Imaging Network
material class

Figure 2.3: The Differential Angular Imaging Network (DAIN) for material recognition.

convenient in-scene appearance capture.

To capture material appearance in a manner that preserves the convenience of image-based methods but important angular information of reflectance-based methods, we assemble a comprehensive, first-of-its-kind, *outdoor* material database that includes multiple viewpoints and multiple illumination directions (partial BRDF sampling), multiple weather conditions, a large set of surface material classes surpassing existing comparable datasets, multiple physical instances per surface class (to capture intra-class variability) and differential viewpoints to support the framework of differential angular imaging. We concentrate on outdoor scenes because of the limited availability of reflectance databases for outdoor surfaces. We also concentrate on materials from ground terrain in outdoor scenes (GTOS) for applicability in numerous application such as automated driving, robot navigation and scene semantics. The 40 surface classes include ground terrain such as grass, gravel, asphalt, concrete, black ice, snow, moss, mud and sand (see Figure 2.2).

We build a recognition algorithm that leverages the strength of deep learning and differential angular imaging. The resulting method takes two image streams as input, the original image and a differential image as illustrated in Figure 2.3.

Datasets	samples	classes	views	illumination	in scene	scene image	camera parameters	year
CUReT[33]	61	61	205		N	N	N	1999
KTH-TIPS[25]	11	11	27	3	N	N	N	2004
UBO2014[34]	84	7	151	151	N	N	N	2014
Reflectance disk[32]	190	19	3	3	N	N	Y	2015
4D Light-field[1]	1200	12	1	1	Y	N	N	2016
NISAR[35]	100	100	9	12	N	N	N	2016
GTOS(ours)	**606**	**40**	**19**	**4**	**Y**	**Y**	**Y**	**2016**

Table 2.1: Comparison between GTOS dataset and some publicly available BRDF material datasets. Note that the 4D Light-field dataset[1] is captured by the Lytro Illum light field camera.

We optimize the two-stream configuration for material recognition performance and call the resulting network DAIN–differential angular imaging network. We make three significant contributions in this paper: 1) Introduction of differential angular imaging as a middle-ground between reflectance-based and image-based material recognition; 2) Collection of the GTOS database made publicly available with over 30000 in-scene outdoor images capturing angular reflectance samples with scene context over a large set of material classes; 3) The development of DAIN, a material recognition network with state-of-the-art performance in comprehensive comparative validation.

2.2 Related Work

Texture recognition, the classification of 3D texture images and bidirectional texture functions, traditionally relied on hand-designed 3D image features and multiple views [17, 36]. More recently, features learned with deep neural networks have outperformed these methods for texture recognition. Cimpoi *et al.*[23] achieves state-of-art results on FMD [24] and KTH-TIPS2 [25] using a Fisher vector representation computed on image features extracted with a CNN.

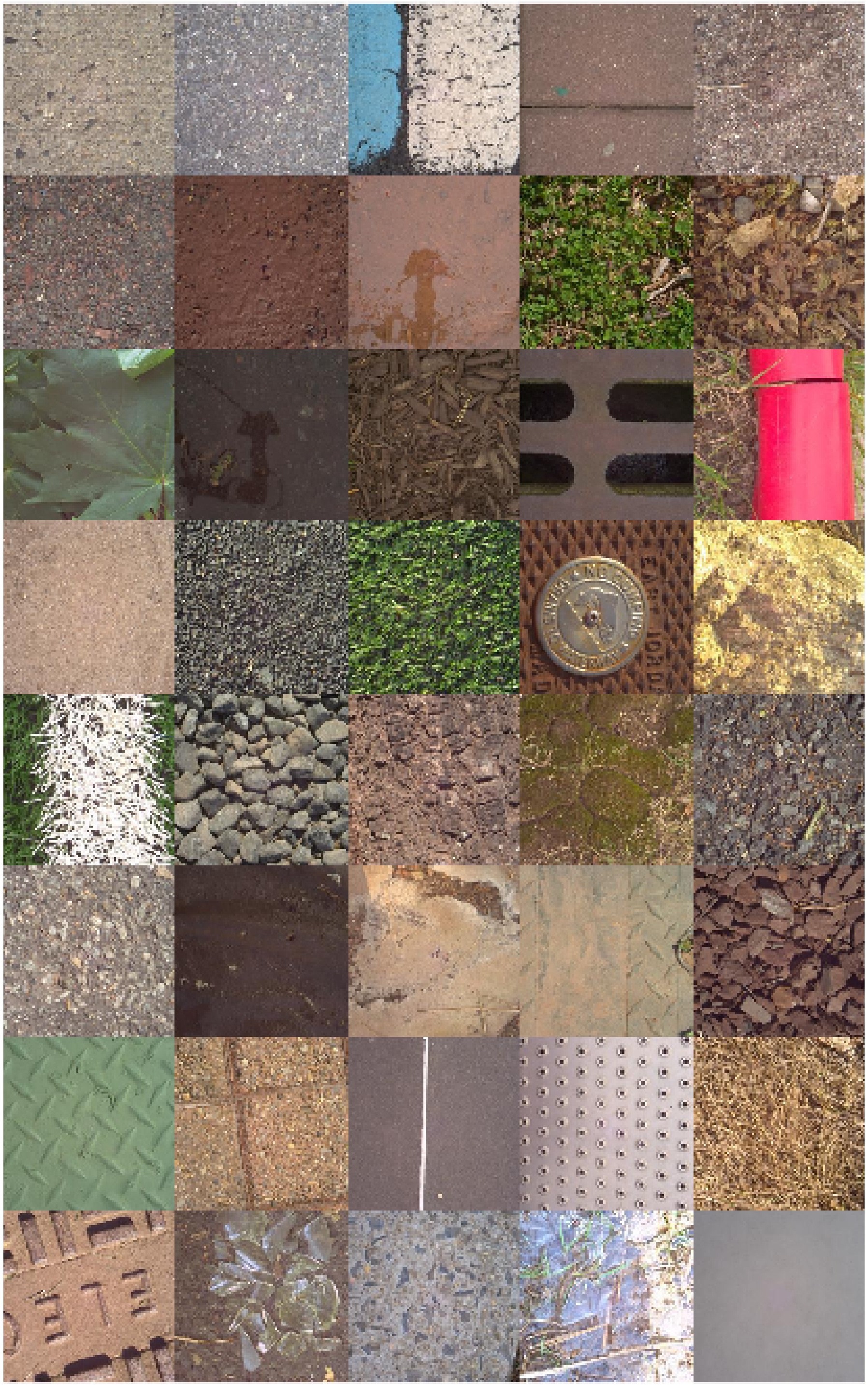


Figure 2.4: The 40 material categories in the GTOS dataset introduced in this paper.

The success of deep learning methods in object recognition has also translated to the problem of material recognition, the classification and segmentation of material categories in arbitrary images. Bell *et al.*, achieve per-pixel material category labeling by retraining the then state-of-the-art object recognition network [22] on a large dataset of material appearance [11]. This method relies on large image patches that include object and scene context to recognize materials. In contrast, Schwartz and Nishino [30, 31] learn material appearance models from small image patches extracted inside object boundaries to decouple contextual information from material appearance. To achieve accurate local material recognition, they introduced intermediate material appearance representations based on their intrinsic properties (e.g., “smooth” and “metallic”).

In addition to the apparent appearance, materials can be discerned by their radiometric properties, namely the bidirectional reflectance distribution function (BRDF) [37]and the bidirectional texture function (BTF) [33], which essentially encode the spatial and angular appearance variations of surfaces. Materials often exhibit unique characteristics in their reflectance offering detailed cues to recognize the difference of subtle variations in them (e.g., different types of metal [10] and paint [9]). Reflectance measurements, however, necessitate elaborate image capture systems, such as a gonioreflectometer [37, 38], robotic arm [39], or a dome with cameras and light sources [40, 10, 9]. Recently, Zhang *et al.*introduced the use of a one-shot reflectance field capture for material recognition [32]. They adapt the parabolic mirror-based camera developed by Dana and Wang [41] to capture the reflected radiance for a given light source direction in a single shot, which they refer to as a reflectance disk. More recently, Zhang *et al.*showed that the reflectance disks contain sufficient information to accurately predict the kinetic friction coeffcient of surfaces [42]. These results demonstrate that the angular appearance variation of materials and their gradients encode rich cues for their recognition. Similarly, Wang *et al.*[1] uses a light field camera and combines angular and spatial filtering for material recognition. In strong alignment with these recent advances in material recognition, we build a framework of spatial and angular appearance filtering. In sharp contrast to past methods, however, we use image information from standard cameras instead of a multilens array as in Lytro. We explore the difference of using a large viewing angle range (with samples coarsely quantized in angle space) by using differential changes in angles which can easily be captured by a two-camera system or small motions of a single ordinary camera.

Deep learning has achieved major success in object classification [43, 44, 45], segmentation [46, 47, 48], and material recognition [23, 42, 3]. In our goal of combining spatial and angular image information to account for texture and reflectance, we are particularly motivated by the two-stream fusion framework [49, 22] which achieves state-of-art results in UCF101[50] action recognition dataset.

Datasets: Datasets to measure reflectance of real world surfaces have a long history of lab-based measurements including: CUReT database[33], KTH-TIPS database by Hayman *et al.*[25], MERL Reflectance Database [51], UBO2014 BTF Database [34], UTIA BRDF Database [52], Drexel Texture Database [53] and ICCERTH Fabric Database [54]. In many of these datasets, dense reflectance angles are captured with special image capture equipment. Some of these datasets have limited instances/samples per surface category (different physical samples representing the same class for intraclass variability) or have few surface categories, and all are obtained from indoor measurements where the sample is removed from the scene. More recent datasets capture materials and texture in-scene, (a.k.a. in-situ, or in-the-wild). A motivation of moving to in-scene capture is to build algorithms and methods that are more relevant to real-world applications. These recent databases are from internet-mined databases and contain a single view of the scene under a single illumination direction. Examples include the the Flickr Materials Database by Sharan *et al.*[24] and the Material in Context Database by Bell *et al.*[11]. Recently, DeGol *et al.*released GeoMat Database[55] with 19 material categories from outdoor sites and each category has between 3 and 26 physical surface instances, with 8 to 12 viewpoints per surface. The viewpoints in this dataset are irregularly sampled in angle space.

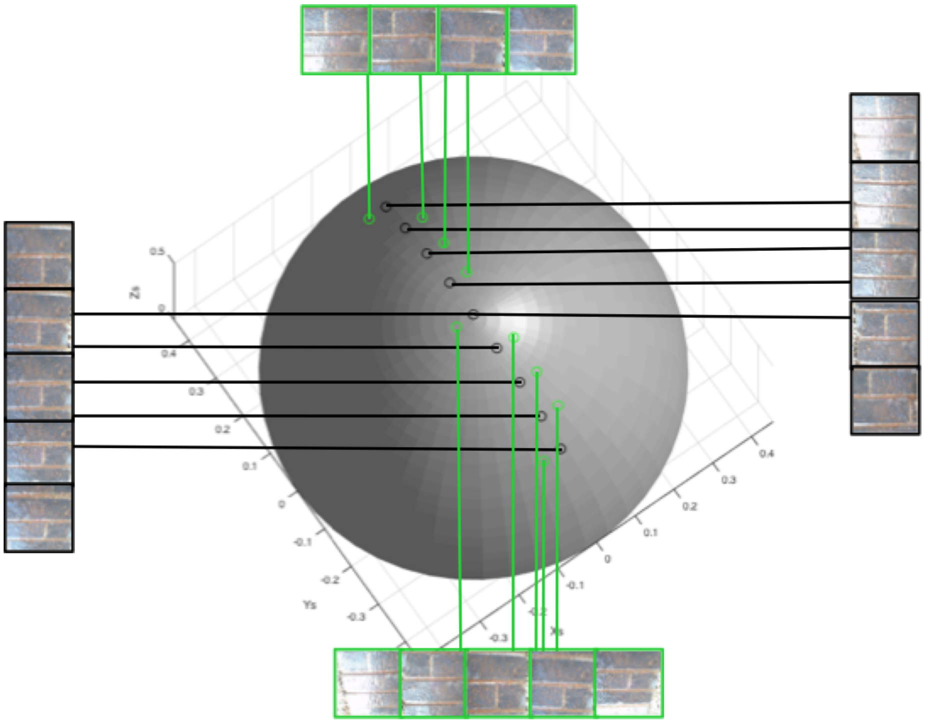


Figure 2.5: The material surface observation points. Nine viewpoint angles separated along an arc spanning 80° are measured. For each viewpoint, a differential view is captured ±5° in azimuth from the original orientation (the sign is chosen based on robotic arm kinematics.

2.3 GTOS Dataset

We collect the GTOS database, a first-of-its-kind in-scene material reflectance database, to investigate the use of spatial and angular reflectance information of outdoor ground terrain for material recognition. We capture reflectance systematically by imaging a set of viewing angles comprising a partial BRDF with a mobile exploration robot. Differential angular images are obtained by also measuring each of $N_v = 9$ base angles $v = (\theta_v, \phi_v)$, $\theta_v \in [-40^\circ, -30^\circ, \ldots, 40]$, and

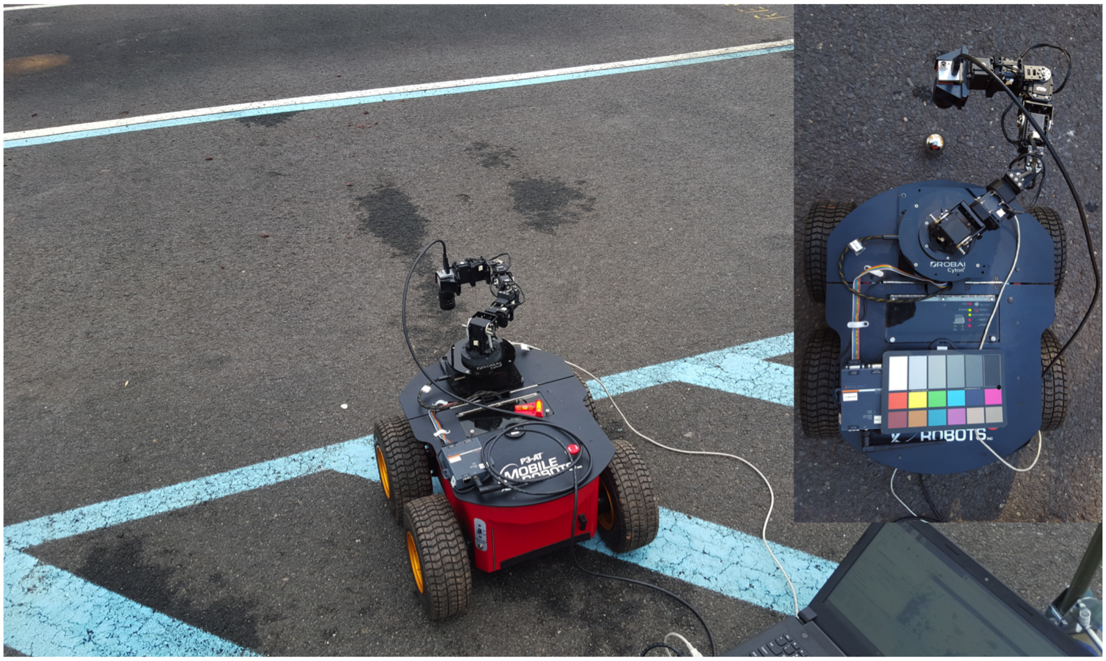
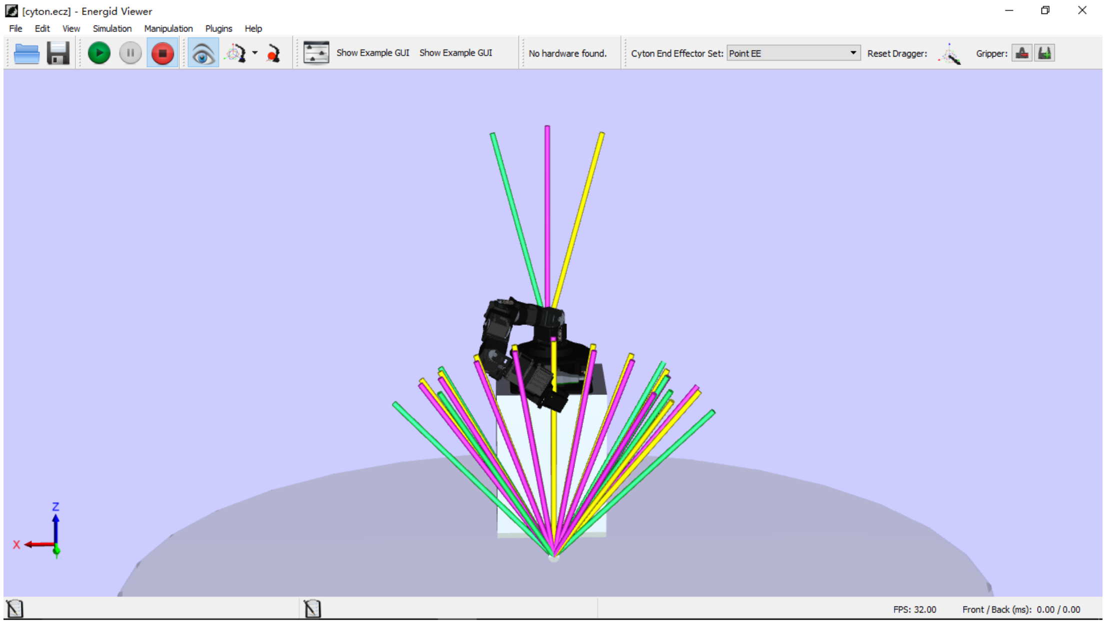


Figure 2.6: The measurement equipment for the GTOS database: Mobile Robots P3-AT robot, Cyton gamma 300 robot arm, Basler aca2040-90uc camera with Edmund Optics 25mm/F1.8 lens, DGK 18% white balance and color reference card, and Hardened 440C Stainless Steel Tight-Tolerance Sphere.

a differential angle variation of $\delta = (0, 5^\circ)$ resulting in 18 viewing directions per sample as shown in Figure 2.5 (b).

Example surface classes are depicted in Figure 2.4 (a). The class names are (in order of top-left to bottom-right): cement, asphalt, painted asphalt , brick, soil, muddy stone, mud, mud-puddle, grass, dry leaves, leaves, asphalt-puddle, mulch, metal grating, plastic, sand, stone, artificial turf, aluminum, limestone, painted turf, pebbles, roots, moss, loose asphalt-stone, asphalt-stone, cloth, paper, plastic cover, shale, painted cover, stone-brick, sandpaper, steel, dry grass, rusty cover, glass, stone-cement, icy mud, and snow. The $N_c = 40$ surface classes mostly have between 4 and 14 instances (samples of intra-class variability) and each instance is imaged not only under N_v viewing directions but also under multiple natural light illumination conditions. As illustrated in Figure 2.1, sample appearance depends on the weather condition and the time of day. To capture this variation,



No hardware found.

Cyton End Effector Set: Point EE

FPS: 32.00

Front / Back (ms): 0.00 / 0.00

Figure 2.7: The user interface to control the robot arm and design poses to manipulate robot arm actions. Each rod represents a preset pose, the 3 rods behind the robot arm are the poses designed to image for the steel sphere. In our dataset collection, some poses are not directly reachable from other poses. That is, it was not possible to sweep the arm over a full 90 degree viewing range ($-45°$ to $45°$). To address this problem, we designed intermediate poses for the robot arm path that allowed us the Cyton gamma arm to achieve a 90 degree viewing angle range.

we image the same region with $N_i = 4$ different weather conditions (cloudy dry, cloudy wet, sunny morning, and sunny afternoon). We capture the samples with 3 different exposure times to enable high dynamic range imaging. Additionally, we image a mirrored sphere to capture the environment lighting of the natural sky. In addition to surface images, we capture a scene image to show the global context.

The robot measurement device is depicted in Figure 2.6. We use Cyton Viewer to control the robot arm, the user interface to control the robot arm and design poses to manipulate robot arm actions is shown in Figure 2.7. Each rod represents a preset pose, the 3 rods behind the robot arm are the poses designed

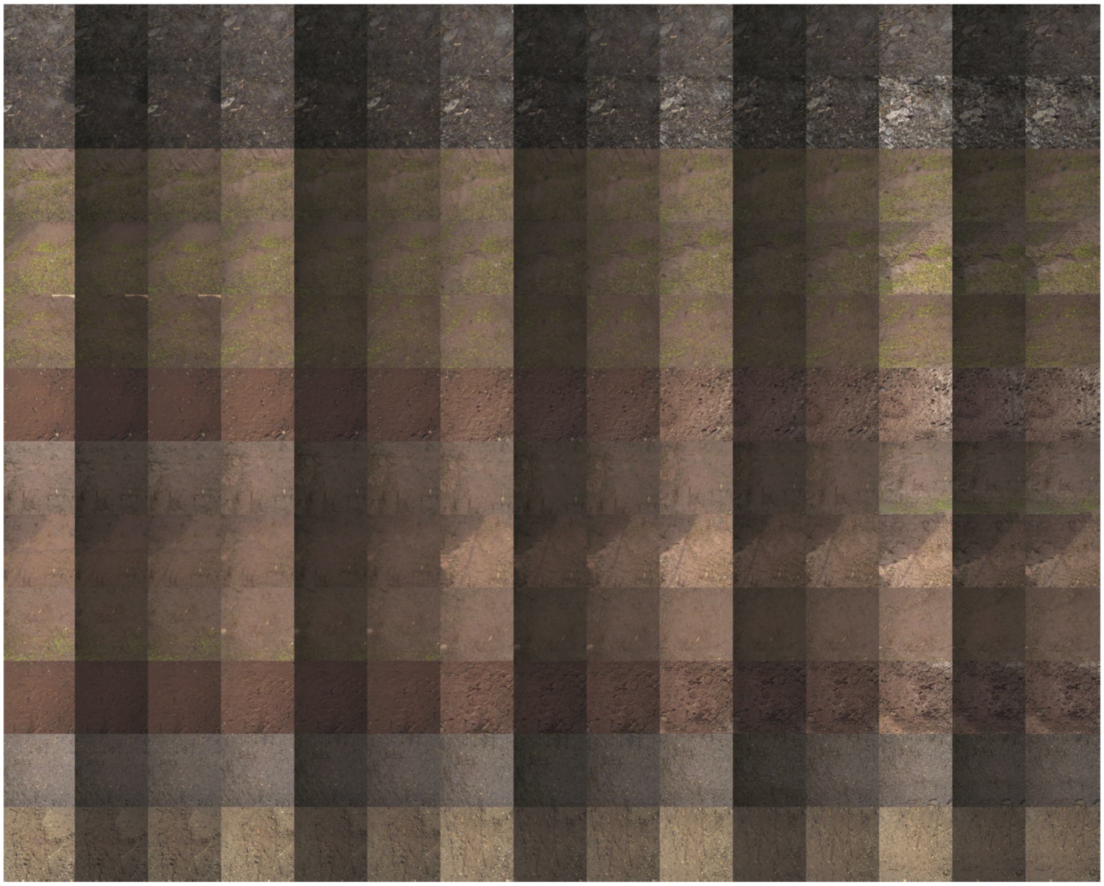


Figure 2.8: Sample images for the mud class are shown. Each row shows a multiview image set (5 viewing angles, each imaged with 3 exposures). Each set of 2-4 rows shows the same physical surface under different weather/illumination condition. Multiple instances of mud (different physical surfaces) are shown.

to image for the steel sphere. In our dataset collection, some poses are not directly reachable from other poses. That is, it was not possible to sweep the arm over a full 90 degree viewing range ($-45°$ to $45°$). To address this problem, we designed intermediate poses for the robot arm path that allowed us the Cyton gamma arm to achieve a 90 degree viewing angle range. For more information about robot arm controlling and the code we used to integrate robot arm and Basler camera, please see Section 2.7. Although, the database measurements were obtained with robotic positioning for precise angular measurements, our recognition results are based on subsets of these measurements so that an articulated arm would not be required for an in-field system.

The total number of surface images in the database is 34,243. As shown in Table 2.1, this is the most extensive outdoor in-scene multiview material database to date. Figure 2.8 shows the sample images for the mud class in the dataset. Each row shows a multiview image set (5 viewing angles, each imaged with 3 exposures). Each set of 2-4 rows shows the same physical surface under different weather/illumination condition. Multiple instances of mud (different physical surfaces) are shown. For more dataset examples, please see Section 2.7.

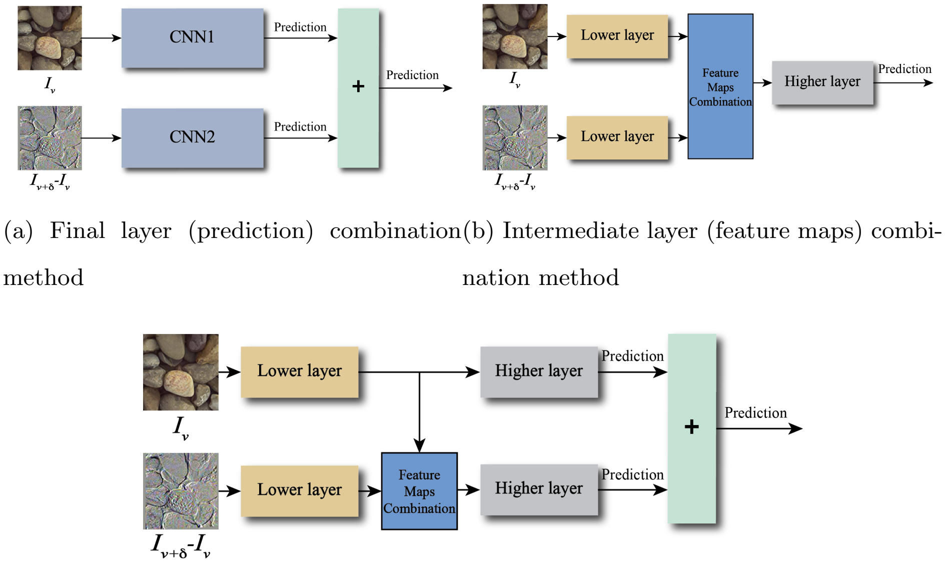
2.4 Methods

2.4.1 Differential Angular Imaging

We present a new measurement method called differential angular imaging where a surface is imaged from a particular viewing angle v and then from an additional viewpoint $v + \delta$. The motivation for this differential change in viewpoint is improved computation of the angular gradient of intensity $\partial I_v / \partial v$. Intensity gradients are the basic building block of image features and it is well known that discrete approximations to derivatives have limitations. In particular, spatial gradients of intensities for an image I are approximated by $I(x+\Delta) - I(x)$ and this approximation is most reasonable at low spatial frequencies and when Δ is small.

For angular gradients of reflectance, the discrete approximation to the derivative is a subtraction with respect to the viewing angle. Angular gradients are approximated by $I(v+\delta) - I(v)$ and this approximation requires a small δ. Consequently, differential angular imaging provides more accurate angular gradients.

The differential images as shown in Figures 2.3 and 2.2 have several characteristics. First, the differential image reveals the gradients in BRDF/BTF at the particular viewpoint. Second, relief texture is also observable in the differential image due to non-planar surface structure. Finally, the differential images are sparse. This sparsity can provide a computational advantage within the network.



(a) Final layer (prediction) combination method

(b) Intermediate layer (feature maps) combination method

(c) DAIN (differential angular image network)

Figure 2.9: Methods to combine two image streams, the original image I_v and the differential image $I_\delta = I_{v+\delta} - I_v$. The best performing configuration is the architecture in (c), which we refer to as differential angular imaging network (DAIN).

2.4.2 Differential Angular Imaging Network (DAIN)

Consider the problem of in-scene material recognition with images from multiple viewing directions (multiview). We develop a two-stream convolutional neural network to fully leverage differential angular imaging for material recognition. The differential image I_δ sparsely encodes reflectance angular gradients as well as surface relief texture. The spatial variation of image intensity remains an important recognition cue and so our method integrates these two streams of information. A CNN is used on both streams of the network and then combined for the final prediction result. The combination method and the layer at which the combination takes place leads to variations of the architecture.

We employ the ImageNet [56] pre-trained VGG-M model [44] as the prediction unit (labeled CNN in Figure 2.9). The first input branch is the image I_v at a specific viewing direction v. The second input branch is the differential image I_δ. The first method of combination shown in Figure 2.9 (a) is a simple averaging of

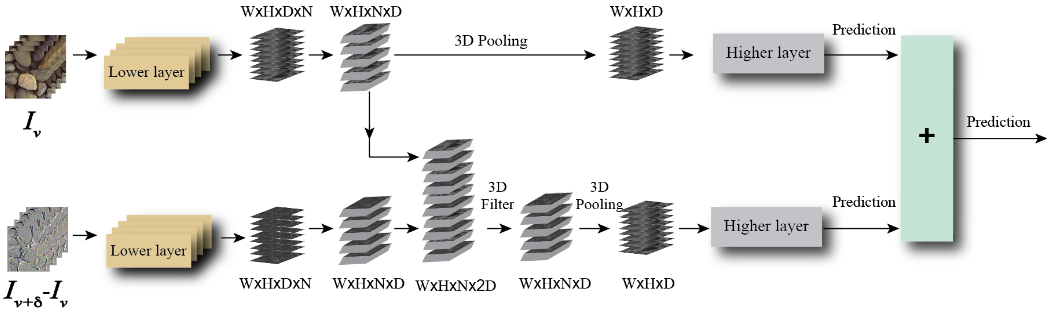


Figure 2.10: Multiview DAIN. The 3D filter + pooling method to combine two streams (original and differential image) from multiple viewing angles. W, H, and D are the width, height, and depth of corresponding feature maps, N is the number of view points.

the output prediction vectors obtained by the two branches. The second method combines the two branches at the intermediate layers of the CNN, i.e. the feature maps output at layer M are combined and passed forward to the higher layers of the CNN, as shown Figure 2.9 (b). We empirically find that combining feature maps generated by Conv5 layer after ReLU performs best. A third method (see Figure 2.9 (c)) is a hybrid of the two architectures that preserves the original CNN path for the original image I_v by combining the layer M feature maps for both streams *and* by combining the prediction outputs for both streams as shown in Figure 2.9 (c). This approach is the best performing architecture of the three methods and we call it the differential angular imaging network (DAIN).

For combining feature maps at layer M, consider features maps x_a and x_b from the two branches that have width W, height H, and feature channel depth D. The output feature map y will be the same dimensions $W \times H \times D$. We can combine feature maps by: (1) *Sum:* pointwise sum of x_a and x_b, and (2) *Max:* pointwise maximum of x_a and x_b. In Section 2.5 we evaluate the performance of these methods of combining lower layer feature maps.

2.4.3 Multiple Views

Our GTOS database has multiple viewing directions on an arc (a partial BRDF sampling) as well as differential images for each viewing direction. We evaluate our recognition network in two modes: (1) **Single view DAIN**, with inputs from I_v and I_δ, with v representing a single viewing angle; (2) **Multi view DAIN**, with inputs I_v and I_δ, with $v \in [v1, v2, ..., vN]$. For our GTOS databse, $v1, v2, ..., vN$ are viewing angles separated by 10° representing a $N \times 10^\circ$ range of viewing angles. We empirically determine that $N = 4$ viewpoints are sufficient for recognition. For a baseline comparison we also consider non-differential versions: **Single View** with only I_v for a single viewing direction and **Multi View** with inputs I_v, $v \in [v1, v2, ..., vN]$.

To incorporate multi view information in DAIN we use three methods: (1) voting (use the predictions from each view to vote), (2) pooling (pointwise maximum of the combined feature maps across viewpoints), (3) 3D filter + pooling (follow [57] to use a $3 \times 3 \times 3$ learned filter bank to convolve the multi view feature maps). See Figure 2.10. After 3D filtering, pooling is used (pointwise maximum across viewpoints). The computational expense of this third method due to learning the filter weights is significantly higher.

2.5 Experiments

In this section, we evaluate the DAIN framework for material recognition and compare the results on GTOS with several state-of-the-art algorithms. The first evaluation determines which structure of the two stream networks from Figure 2.9 works best on the GTOS dataset, leading to the choice in (c) as the DAIN architecture. The second evaluation considers recognition performance with different variations of DAIN recognition. The third experimental evaluation compares three other state-of-the-art approaches on our GTOS-dataset, concluding that multiview DAIN works best. Finally, we apply DAIN to a lightfield dataset to

Method	Final Layer Combination	Intermediate Layer Combination	DAIN
Accuracy	$77.0_{\pm 2.5}$	$74.8_{\pm 3.4}$	$79.4_{\pm 3.4}$

Table 2.2: Comparison of accuracy from different two stream methods as shown in Figure 2.9. The feature-map combination method for (b) and (c) is Sum at Conv5 layers after ReLU. The reported result is the mean accuracy and the subscript shows the standard deviation over 5 splits of the data. Notice that the architecture in (c) gives the best performance and is chosen for the differential angular imaging network (DAIN).

show performance in another multiview material dataset.

2.5.1 Training procedure

We design 5 training and testing splits by assigning about 70% of ground terrain surfaces of each class to training and the rest 30% to testing. Note that, to ensure that there is no overlap between training and testing sets, if one sample is in the training set, all views and illumination conditions for that sample is in the training set.

Each input image from our GTOS database is resized into 240 × 240. Before training a two branch network, we first fine-tune the VGG-M model separately with original and differential images with batch size 196, dropout rate 0.5, momentum 0.9. We employ the augmentation method that horizontally and vertically stretch training images within ±10%, with an optional 50% horizontal mirror flips. The images are randomly cropped into 224 × 224 material patches. All images are pre-processed by subtracting a per color channel mean and normalizing for unit variance. The learning rate for the last fully connected layer is set to 10 times of other layers. We first fine-tune only the last fully connected layer with learning rate 5×10^{-2} for 5 epochs; then, fine-tune all the fully connected layers with learning rate 10^{-2} for 5 epochs. Finally we fine-tune all the layers with leaning rate starting at 10^{-3}, and decrease by a factor of 0.1 when the training accuracy saturates. Since the snow class only has 2 samples, we omit them from experiments.

For the two branch network, we employ the fine-tuned two-branch VGG-M model with batch size 64 and learning rate starting from 10^{-3} which is reduced by a factor of 0.1 when the training accuracy saturates. We augment training data with randomly stretch training images by ±25% horizontally and vertically, and also horizontal mirror flips. The images are randomly cropped to 224 × 224 material patches. We first backpropagate only to feature maps combination layer for 3 epochs, then fine tunes all layers. We employ the same augmentation method for the multiview images of each material surface. We randomly select the first viewpoint image, then subsequent $N = 4$ view point images are selected for experiments.

2.5.2 Evaluation for DAIN Architecture

Table 2.2 shows the mean classification accuracy of the different three branch combination methods depicted in Figure 2.9. Inputs are single view images (I_v) and single view differential images (I_δ). Combining the two streams at the final prediction layer (77% accuracy) is compared with the intermediate layer combination (74.8%) or the hybrid approach in Figure 2.9 (c) (79.4%) which we choose as the differential angular imaging network. The combination method used is Sum and the feature maps are obtained from Conv5 layers after ReLU.

2.5.3 DAIN Recognition Performance

We evaluate DAIN recognition performance for single view input (and differential image) and for multiview input from the GTOS database. Additionally, we compare the results to recognition using a standard CNN without a differential image stream. For all multiview experimental results we choose the number of

Method	Final Layer Combination	Intermediate Layer Combination	DAIN
Accuracy	$77.0_{\pm 2.5}$	$74.8_{\pm 3.4}$	$79.4_{\pm 3.4}$

Table 2.3: Results comparing performance of standard CNN recognition without angular differential imaging (first three rows) to our single-view DAIN (middle three rows) and our multi-view DAIN (bottom three rows). I_v denotes the image from viewpoint v, $I_{v+\delta}$ is the image obtained from viewpoint $v + \delta$, and $I_\delta = I_v - I_{v+\delta}$ is the differential image. The differential angular imaging network (DAIN) has superior performance over CNN even when comparing single view DAIN to multiview CNN. Multiview DAIN provides the best recognition rates.

viewpoints $N = 4$, separated by 10° with the starting viewpoint chosen at random (and the corresponding differential input). Table 2.3 shows the resulting recognition rates (with standard deviation over 5 splits shown as a subscript). The first three rows shows the accuracy *without* differential angular imaging, using both single view and multiview input. Notice the recognition performance for these non-DAIN results are generally lower than the DAIN recognition rates in the rest of the table. The middle three rows show the recognition results for single view DAIN. For combining feature maps we evaluate both Sum and Max which have comparable results. Notice that single view DAIN achieves better recognition accuracy than multiview CNN with voting (79.4% vs. 78.1%). This is an important result indicating the power of using the differential image. Instead of four viewpoints separated by 10° a single viewpoint and its differential image achieves a better recognition. These results provide design cues for building imaging systems tailored to material recognition. We also evaluate whether using inputs from the two viewpoints directly (i.e. I_v and $I_{v+\delta}$) is comparable to using I_v and the differential image I_δ. Interestingly, the differential image as input has an advantage (79.4% over 77.5%). The last three rows of Table 2.3 show that recognition performance using multiview DAIN beats the performance of both single view DAIN and CNN methods with no differential image stream. To further analyze DAIN performance for each material classes, we construct the confusion matrix of GTOS dataset with multiview DAIN(Sum/pooling) model as shown in Figure 2.11. We evaluate different ways to combine the multiview image set including voting, pooling, and the 3D filter+pooling illustrated in Figure 2.10.

The CNN module of our DAIN network can be replaced by other state-of-the-art deep learning methods to further improve results. To demonstrate this, we change the CNN module in a single view DAIN (Sum) (with inputs I_v, I_δ) to ImageNet pre-trained ResNet-50 model[45] on split1. Combining feature maps generated from the Res4 layer (the fourth residual unit) after ReLU with training batch size 196, recognition rate improves from 77.5% to 83.0%.

Table 2.4 shows the recognition rates for multiview DAIN that outperforms three other multi-view classification method: FV+CNN[13], FV-N+CNN+N3D [55], and MVCNN[58]. The table shows recognition rates for a single split of the GTOS database with images resized to 240 × 240. All experiments are based on the same pre-trained VGG-M model. We use the same fine-tuning and training procedure as in the MVCNN[58] experiment. For FV-N+CNN+N3D applied to GTOS, 10 samples (out of 606) failed to get geometry information by the method provided in [55] and we removed these samples from the experiment. The patch size in [55] is 100 × 100, but the accuracy for this patch size for GTOS was only 43%, so we use 240 × 240. We implement FV-N+CNN+N3D with linear mapping instead of homogeneous kernel map[59] for SVM training to save memory with this larger patch size.

DAIN on 4D Light Field Dataset

We tested our multiview DAIN (Sum + pooling) method on a recent 4D light field (Lytro) dataset [1]. ResNet-50 is used as the CNN module. The recognition accuracy with full images on 5 splits is $83.0_{\pm 2.1}$ which outperforms the results (80%) reported for the 4D filter method [1].

The Lytro dataset has $N = 49$ views, from the 7 × 7 lenslet array, where each lenslet

Architecture	Accuracy
FV+CNN[13]	75.4%
FV-N+CNN+N3D [55]	58.3%
MVCNN[58]	78.1%
multiview DAIN (3D filter), pooling	**81.4%**

Table 2.4: Comparison with the state of art algorithms on GTOS dataset. Notice that our method, multiview DAIN, achieves the best recognition accuracy.

corresponds to a different viewing direction. Using (i, j) as an index into this array, we employ the viewpoints indexed by $(4, 1), (4, 3), (4, 5), (4, 7)$ as the 4 views in multiview DAIN. We use the viewpoint indexed by $(3, 1), (5, 3), (3, 5), (5, 7)$ as the corresponding differential views. This is an approximation of multiview DAIN; the lightfield dataset does not capture the range of viewing angles to exactly emulate multiple viewpoints and small angle variations of these viewpoints. Instead of using all $N = 49$ viewpoints as in [1], we generate comparable recognition accuracy by only 8 viewpoints.

2.6 Summary and Conclusion

In summary, there are three main contributions of this work: 1) Differential Angular Imaging for a sparse representation of the spatial distribution of angular gradients that provides key cues for material recognition; 2) The GTOS Dataset with ground terrain imaged by systematic in-scene measurement of partial reflectance instead of in-lab reflectance measurements. The database contains 34,243 images with 40 surface classes, 18 viewing directions, 4 illumination conditions, 3 exposure settings per sample and several instances/samples per class. 3) We develop and evaluate an architecture for using differential angular imaging, showing superior results for differential inputs as compared to original images. Our work in measuring and modeling outdoor surfaces has important implications for applications such as robot navigation (determining control parameters based on current ground terrain) and automatic driving (determining road conditions by partial real time reflectance measurements). We believe our database and

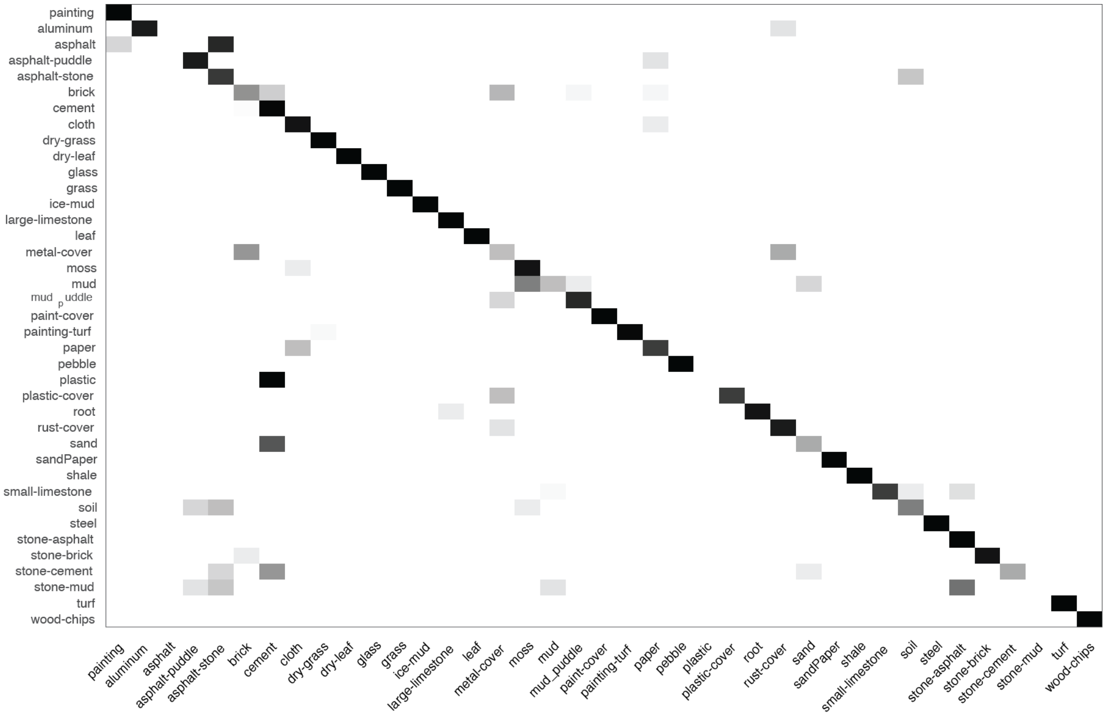


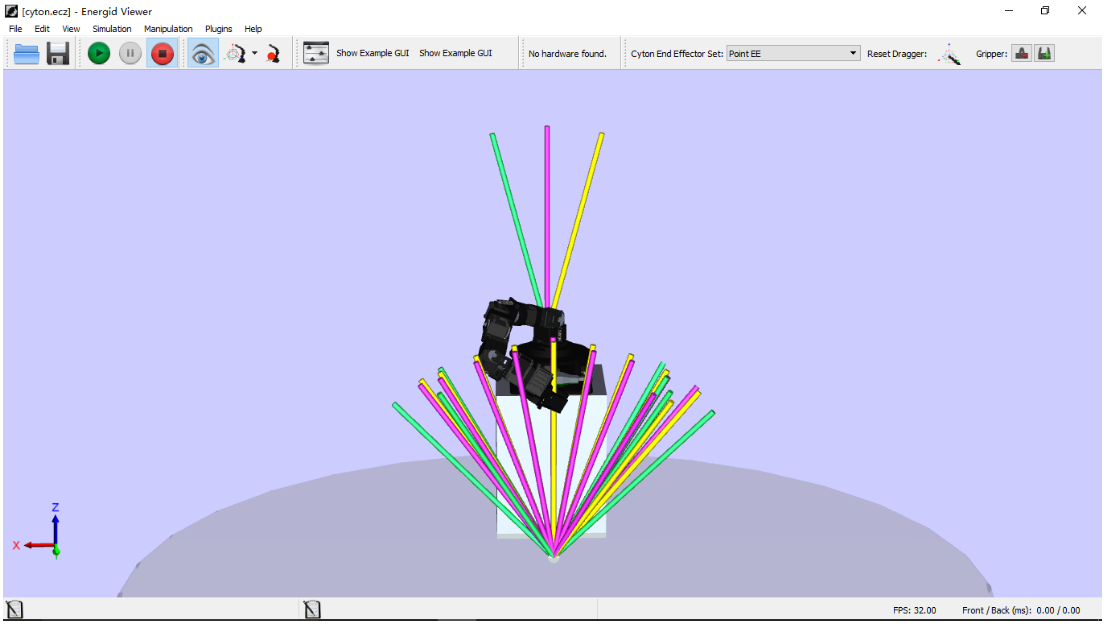
Figure 2.11: Confusion matrix of GTOS dataset with multiview DAIN(Sum/pooling) model. (Darker values are higher values).

methods will provide a sound foundation for in-depth studies on material recognition in the wild.

2.7 Appendix

2.7.1 Measurement Devices

Cyton Viewer is the software we used to control the robot arm, it is designed by Robai. The Cyton Viewer can be used to both simulate motion of the robot and to directly control the robot. It has many powerful features that allow for real time end-effector or joint level control the Cython hardware. To control robot arm image samples with the same multiview observation points, we first design poses, which are coordinate system transformations for robot arm. Figure 2.12 is the front view and aerial view of the user interface to control the robot arm and design poses to manipulate robot arm actions. Each rob is a preset pose, the 3 robs behind the robot arm are the poses



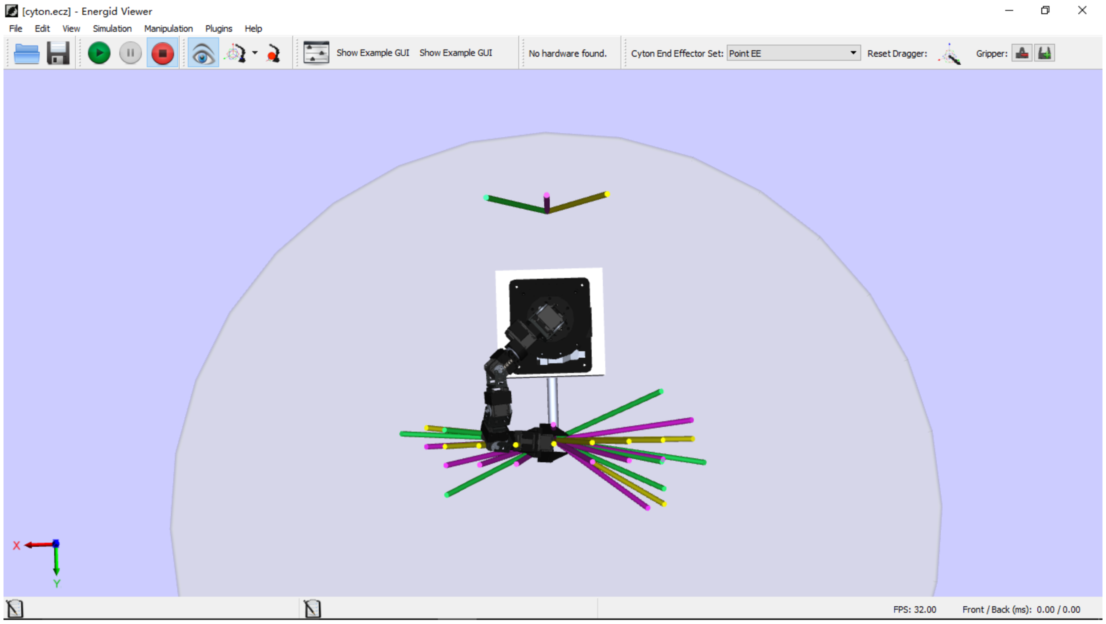
No hardware found.

Cyton End Effector Set: Point EE

FPS: 32.00

Front / Back (ms): 0.00 / 0.00

(a) The front view



No hardware found.

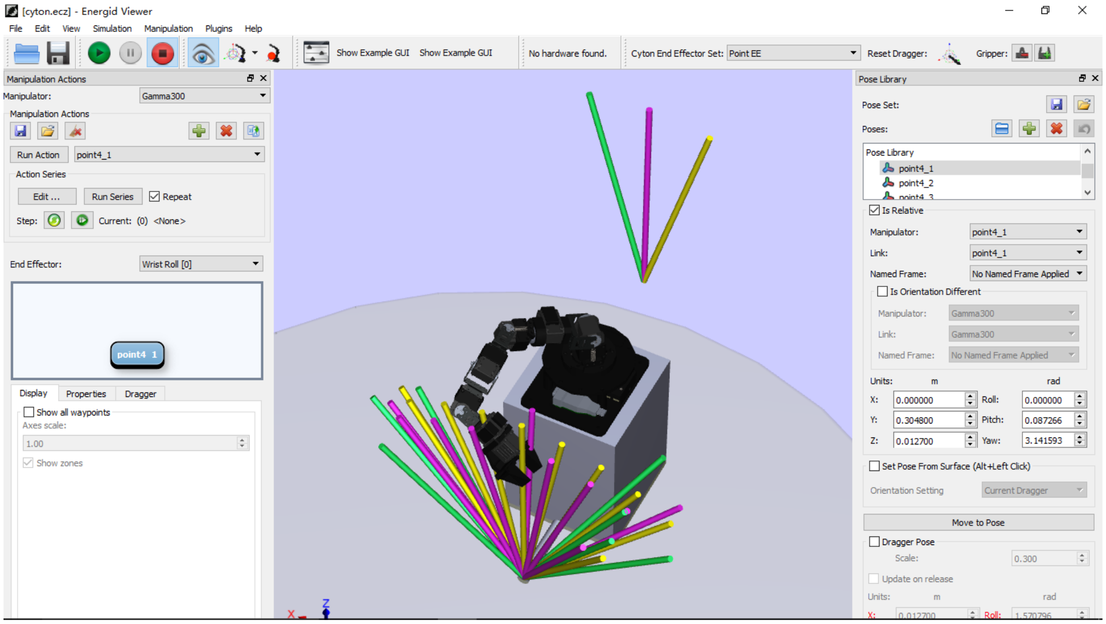
Cyton End Effector Set: Point EE

FPS: 32.00

Front / Back (ms): 0.00 / 0.00

(a) The aerial view

Figure 2.12: The front view (top) and aerial view (down) of the user interface to control the robot arm and design poses to manipulate robot arm actions. Each rod represents a preset pose, the 3 rods behind the robot arm are the poses designed to image for the steel sphere.



[cyton.ecz] - Energid Viewer

No hardware found.

Cyton End Effector Set: Point EE

Reset Dragger:

Gripper:

Manipulator: Gamma300

Run Action: point4_1

☑ Repeat

Step: Current: (0) <None>

End Effector: Wrist Roll [0]

□ Show all waypoints

Axes scale: 1.00

☑ Show zones

Pose Library: point4_1, point4_2, point4_3

☑ Is Relative

Manipulator: point4_1

Link: point4_1

Named Frame: No Named Frame Applied

□ Is Orientation Different

Manipulator: Gamma300

Link: Gamma300

Named Frame: No Named Frame Applied

Units:	m		rad
X:	0.000000	Roll:	0.000000
Y:	0.304800	Pitch:	0.087266
Z:	0.012700	Yaw:	3.141593

□ Set Pose From Surface (Alt+Left Click)

Orientation Setting: Current Dragger

□ Dragger Pose

Scale: 0.300

□ Update on release

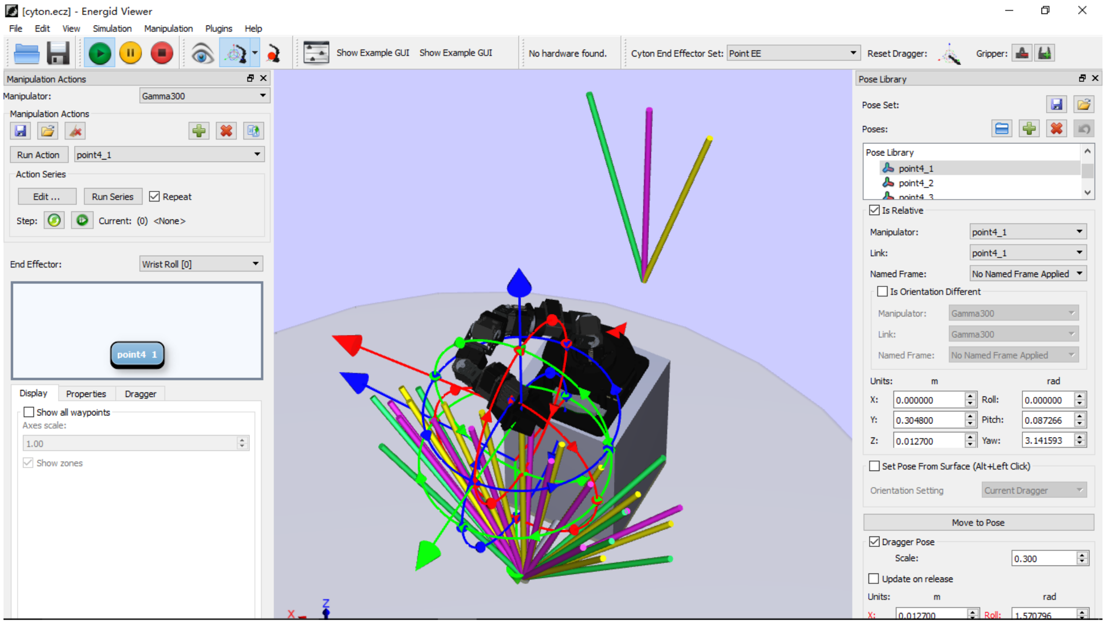
Units:	m		rad
X:	0.012700	Roll:	1.570796

Figure 2.13: The interface for editing poses and manipulation actions. We edit the poses and moving pose sequences with Manipulation → Edit Poses and Manipulation → Edit Manipulation Actions.

designed to image for the steel sphere. We edit the poses and moving pose sequences with Manipulation → Edit Poses and Manipulation → Edit Manipulation Actions, the interface for pose editing is shown in Figure 2.13. As shown in Figure 2.14, we can set poses based on rotation and use mouse to change the view of the robot arm. In our dataset collection, some points are not directly reachable, so we design some auxiliary poses for robot arm to get a 90 degree viewing angles.

When we take multiview observing images. To avoid control robot arm to each position and take image manually for each observing point, we design pose sequences with manipulation actions for all observation points and create plugin to combine robot arm and Basler camera into one image processing. In manipulation actions, the designed program is to take image at each point. For points that are not directly reachable, we need to drag auxiliary poses into End Effector to run series poses for target point, a example point is shown in Figure 2.15. After actions manipulation, click Run Series to make sure the robot arm is not stuck in the series.

We use Basler aca2040-90uc camera with Edmund Optics 25mm/F1.8 lens to image



No hardware found.

Cyton End Effector Set: Point EE

Manipulator: Gamma300

Run Action: point4_1

☑ Repeat

Current: (0) <None>

End Effector: Wrist Roll [0]

□ Show all waypoints

Axes scale: 1.00

☑ Show zones

Pose Library: point4_1, point4_2, point4_3

☑ Is Relative

Manipulator: point4_1

Link: point4_1

Named Frame: No Named Frame Applied

□ Is Orientation Different

Manipulator: Gamma300

Link: Gamma300

Named Frame: No Named Frame Applied

Units: m rad

X: 0.000000 Roll: 0.000000

Y: 0.304800 Pitch: 0.087266

Z: 0.012700 Yaw: 3.141593

□ Set Pose From Surface (Alt+Left Click)

Orientation Setting: Current Dragger

☑ Dragger Pose

Scale: 0.300

□ Update on release

Units: m rad

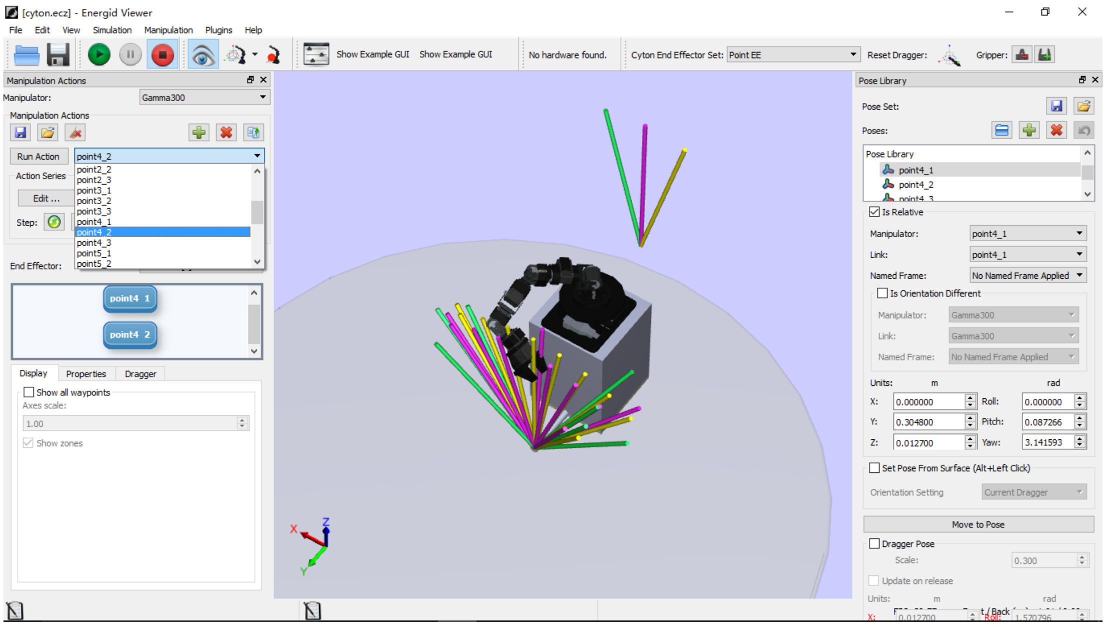
X: 0.012700 Roll: 1.570796

Figure 2.14: we can set poses based on rotation and use mouse to change the view of the robot arm.

material samples. The Basler camera supports pylon Camera Software Suite, which enables to build Basler pylon based C++ applications. The source code to declare Basler camera and define camera settings (gain, while balance and exposure) in C++ is shown below

```
#include <pylon/usb/BaslerUsbInstantCamera.h>
typedef Pylon::CBaslerUsbInstantCamera Camera_t;
using namespace Basler_UsbCameraParams;
typedef Camera_t::GrabResultPtr_t GrabResultPtr_t;
void AutoGainOnce(Camera_t& camera);
void AutoWhiteBalance(Camera_t& camera);
void AutoExposureContinuous(Camera_t& camera);
```

The material sample imaging logic is that we first put on calibration card. By observing calibration card, camera will continuous adjust gain, exposure and white balance. When camera find the correct gain, exposure and white balance, we will fix camera parameters and image for the material sample. The source code is shown below.



No hardware found.

Cyton End Effector Set: Point EE

Reset Dragger:

Gripper:

Manipulator: Gamma300

End Effector:

point4_2

point2_2
point2_3
point3_1
point3_2
point3_3
point4_1
point4_2
point4_3
point5_1
point5_2

□ Show all waypoints

Axes scale: 1.00

☑ Show zones

Pose Library
point4_1
point4_2
point4_3

☑ Is Relative

Manipulator: point4_1

Link: point4_1

Named Frame: No Named Frame Applied

□ Is Orientation Different

Manipulator: Gamma300

Link: Gamma300

Named Frame: No Named Frame Applied

Units:	m		rad
X:	0.000000	Roll:	0.000000
Y:	0.304800	Pitch:	0.087266
Z:	0.012700	Yaw:	3.141593

□ Set Pose From Surface (Alt+Left Click)

Orientation Setting: Current Dragger

□ Dragger Pose

Scale: 0.300

□ Update on release

Units:	m		rad
X:	0.012700	Roll:	1.570796

Figure 2.15: The example pose to drag auxiliary poses into End Effector to run series poses for target point.

```
CDeviceInfo info;
info.SetDeviceClass(Camera_t::DeviceClass());
/* Create an instant camera object with the first found
camera device that matches the specified device class.*/
Camera_t camera(CTlFactory::GetInstance().CreateFirstDevice(info));
QMessageBox::information(this, tr("Info"),
tr(camera.GetDeviceInfo().GetModelName()));
// Open the camera to allow parameter changes
camera.Open();
camera.TestImageSelector = TestImageSelector_Off;
// This smart pointer will receive the grab result data.
CGrabResultPtr ptrGrabResult;
QMessageBox::information(this, tr("Info"), tr("Put on calibration card"));
AutoGainContinuous(camera);
AutoExposureContinuous(camera);
// Carry out white balance using the balance white auto function.
```

```
AutoWhiteBalance(camera);
AutoGainOnce(camera);
AutoExposureOnce(camera);
AutoWhiteBalance(camera);

QMessageBox::information(this, tr("Info"), tr("Image for sample"));
for (int ii = 0; ii < numActions; ++ii)
{
  m_pPlugin->runManipulationAction(manipActionList[ii].toStdString());
  // Set up the stream grabber to start acquisition
  camera.StartGrabbing();
  // Execute a trigger so the camera can acquire
  camera.ExecuteSoftwareTrigger();
  /* Wait for an image and then retrieve it.
  A timeout of 5000ms is used.*/
  camera.RetrieveResult(5000, ptrGrabResult, \
  TimeoutHandling_ThrowException);
  // Image grabbed successfully?
  if (ptrGrabResult->GrabSucceeded())
  {
    n = sprintf(buffer, "sample%s.bmp", setWitTwo(ii).c_str());
    filename = buffer;
    CImagePersistence::Save(ImageFileFormat_Bmp, filename, ptrGrabResult);
  }
  else
  {
    QMessageBox::information(this, tr("Info"), tr("Error: "\
    + ptrGrabResult->GetErrorCode() + ' ' + \
    ptrGrabResult->GetErrorDescription()));
  }
  // end the command for the stream grabber to stop acquisition
```

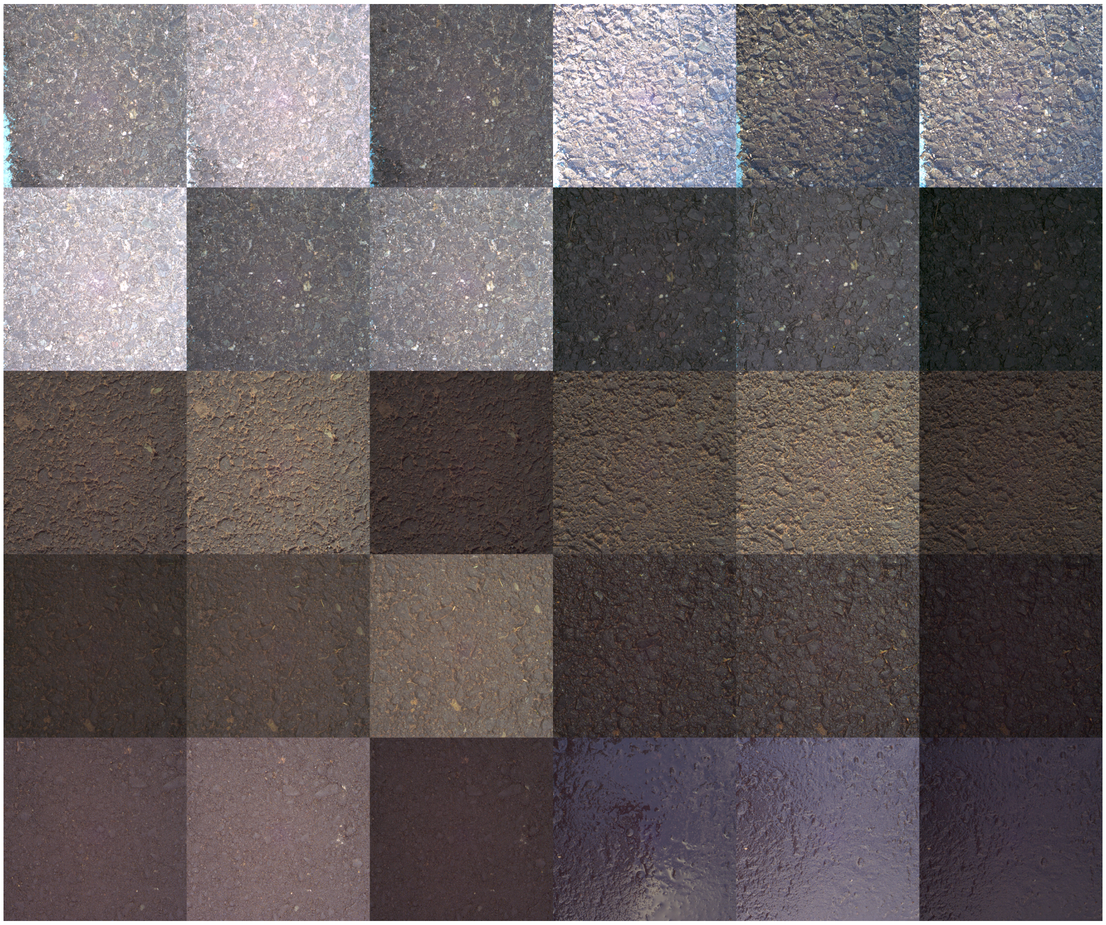


Figure 2.16: Sample images for the asphalt class are shown. Each row shows the same physical surface under different weather/illumination condition. Multiple instances of asphalt (different physical surfaces) are shown.

```
  camera.StopGrabbing();
}
```

2.7.2 GTOS Dataset

Figure 2.17 is another example to show scene-surfaces imaged at different illumination/weather conditions. Figure 2.16 shows the sample images for the asphalt class. Each row shows the same physical surface under different weather/illumination condition. Multiple instances of asphalt (different physical surfaces) are shown. Figure 2.18 shows the sample images for the sphere. Each row shows the same physical surface

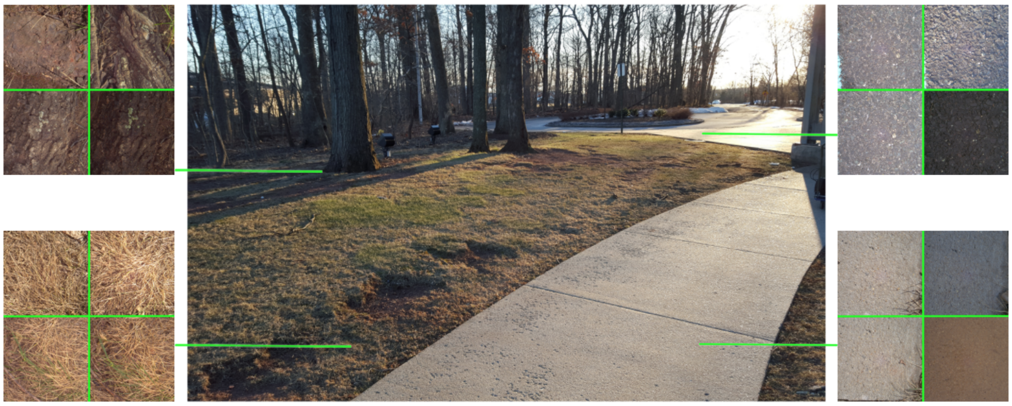


Figure 2.17: Example from GTOS dataset comprising outdoor measurements with multiple viewpoints, illumination conditions and angular differential imaging. The example shows scene-surfaces imaged at different illumination/weather conditions.

under different weather/illumination condition. Figure 2.19 is the multiview images for an asphalt sample and Figure 2.20 is the multiview images for an painting sample. The dataset can be downloaded in `http://computervision.engr.rutgers.edu`, there are 3 different image resolution datasets in the webpage, 1. 256×256 resolution images for material recognition. 2. 512×512 resolution images for material reconstruction. 3. 2048×2048 full resolution images. The 512×512 resolution and the 2048×2048 full resolution datasets also include images for steel sphere (as shown in Figure 2.18), which can be employed to capture weather condition, and text files which record the camera parameters (gain, while balance and exposure).

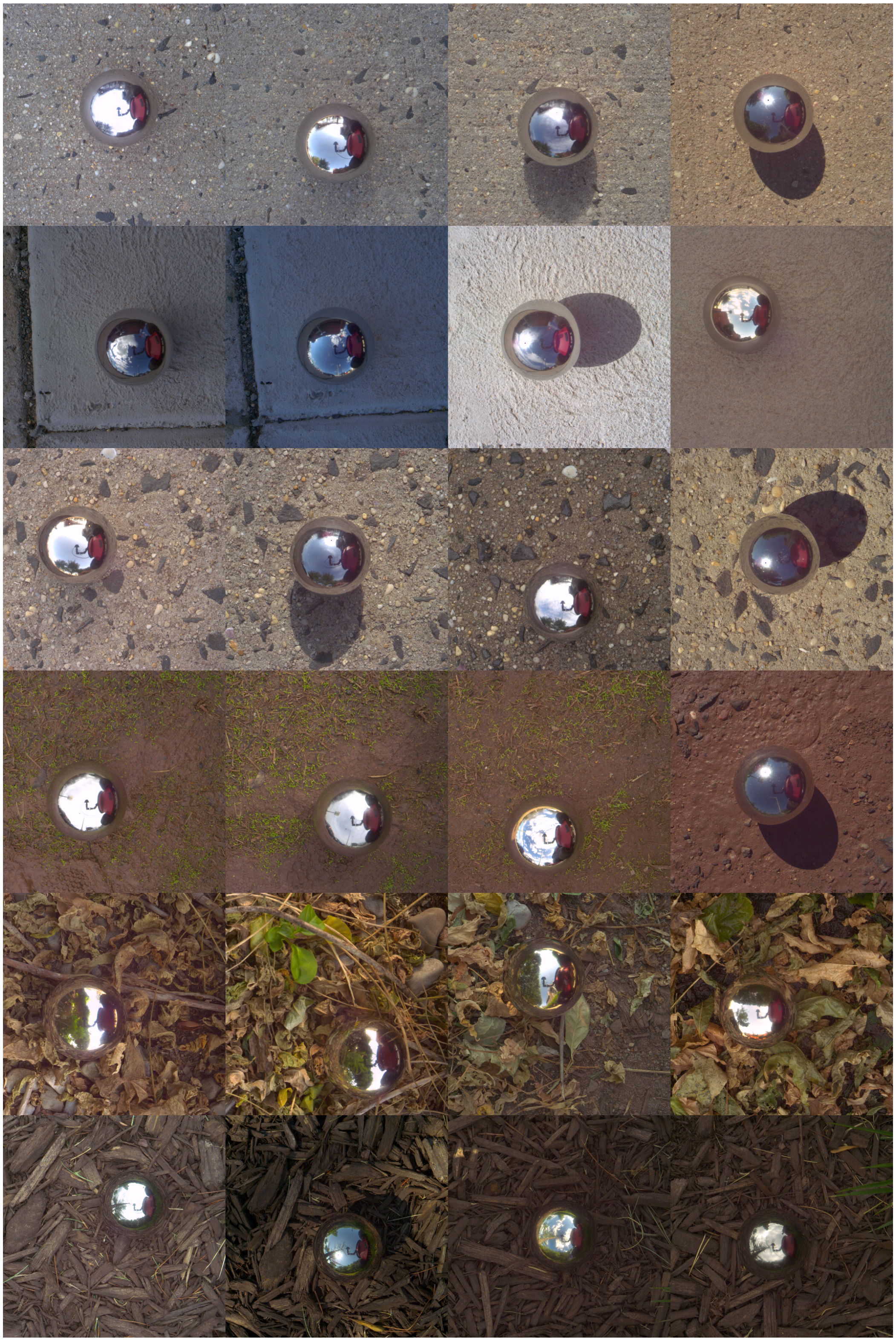


Figure 2.18: Sample images for the sphere. Each row shows the same physical surface under different weather/illumination condition.

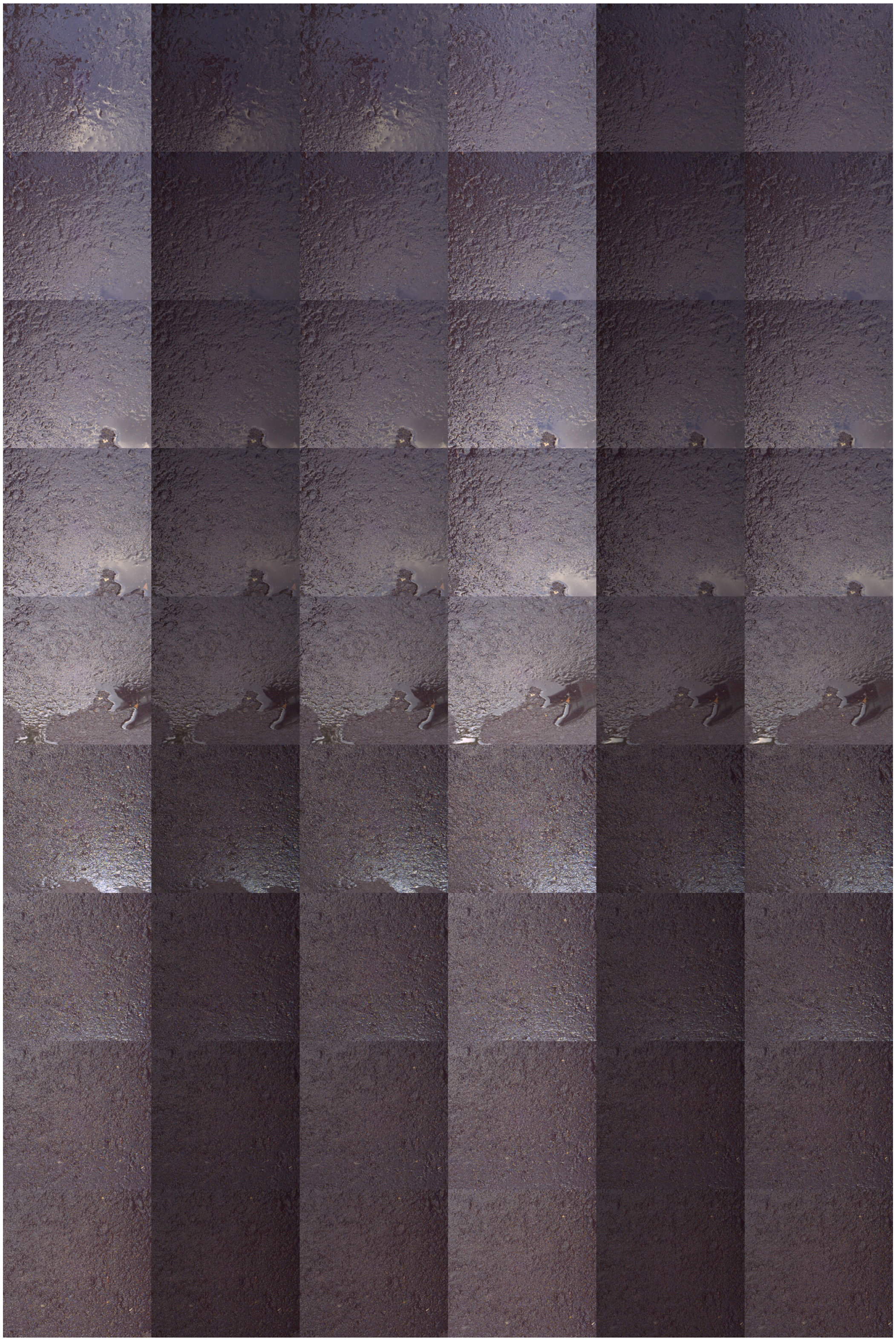


Figure 2.19: The multiview images for an asphalt sample.

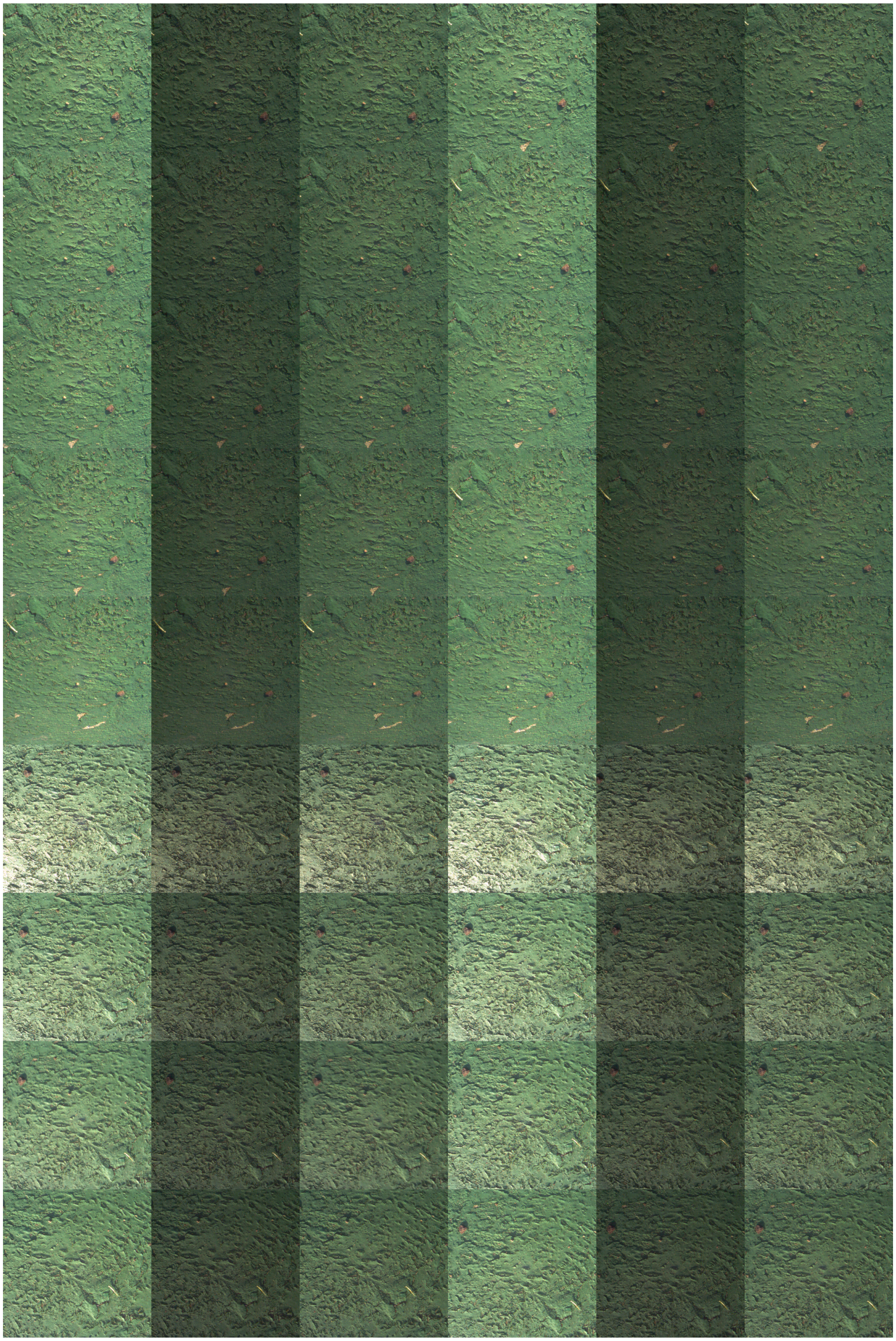


Figure 2.20: The multiview images for an painting sample.

Chapter 3

Deep Texture Manifold for Ground Terrain Recognition

This chapter on Deep Texture Manifold for Ground Terrain Recognition is based on our paper [60]. We present a texture network called Deep Encoding Pooling Network (DEP) for the task of ground terrain recognition. Recognition of ground terrain is an important task in establishing robot or vehicular control parameters, as well as for localization within an outdoor environment. The architecture of DEP integrates orderless texture details and local spatial information and the performance of DEP surpasses state-of-the-art methods for this task. The GTOS database (comprised of over 30,000 images of 40 classes of ground terrain in outdoor scenes) enables supervised recognition. For evaluation under realistic conditions, we use test images that are not from the existing GTOS dataset, but are instead from hand-held mobile phone videos of similar terrain. This new evaluation dataset, GTOS-mobile, consists of 81 videos of 31 classes of ground terrain such as grass, gravel, asphalt and sand. The resultant network shows excellent performance not only for GTOS-mobile, but also for more general databases (MINC and DTD). Leveraging the discriminant features learned from this network, we build a new texture manifold called DEP-manifold. We learn a parametric distribution in feature space in a fully supervised manner, which gives the distance relationship among classes and provides a means to implicitly represent ambiguous class boundaries.

3.1 Background

Ground terrain recognition is an important area of research in computer vision for potential applications in autonomous driving and robot navigation. Recognition with CNNs have achieved success in object recognition and the CNN architecture balances preservation of relative spatial information (with convolutional layers) and aggregation

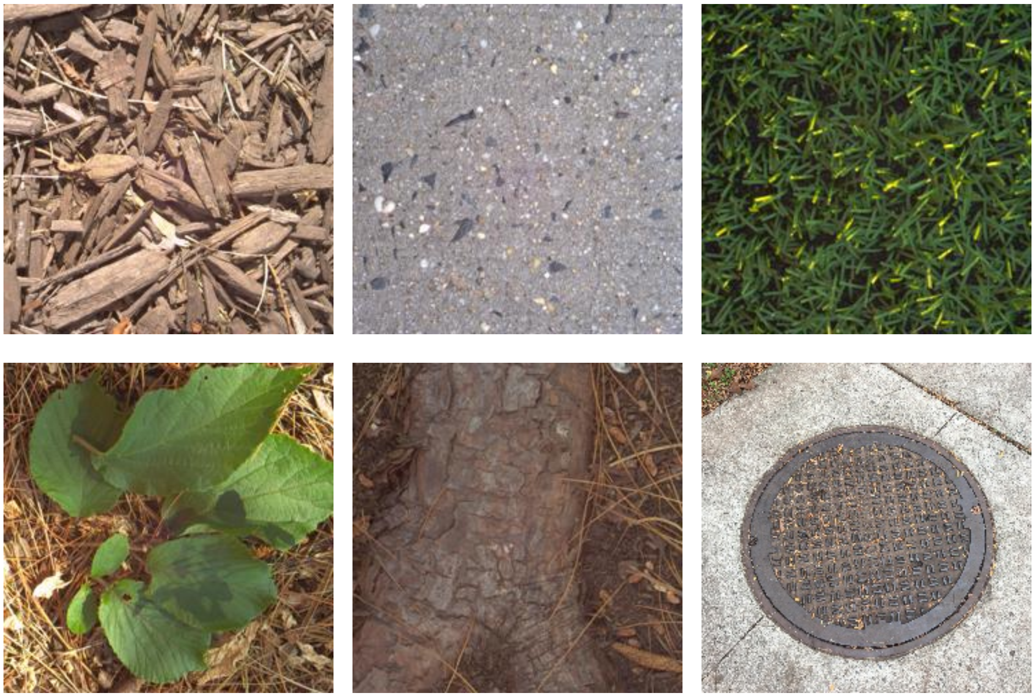


Figure 3.1: Homogeneous textures (upper row) compared to more common real-world instances with local spatial structure that provides an important cue for recognition (lower row).

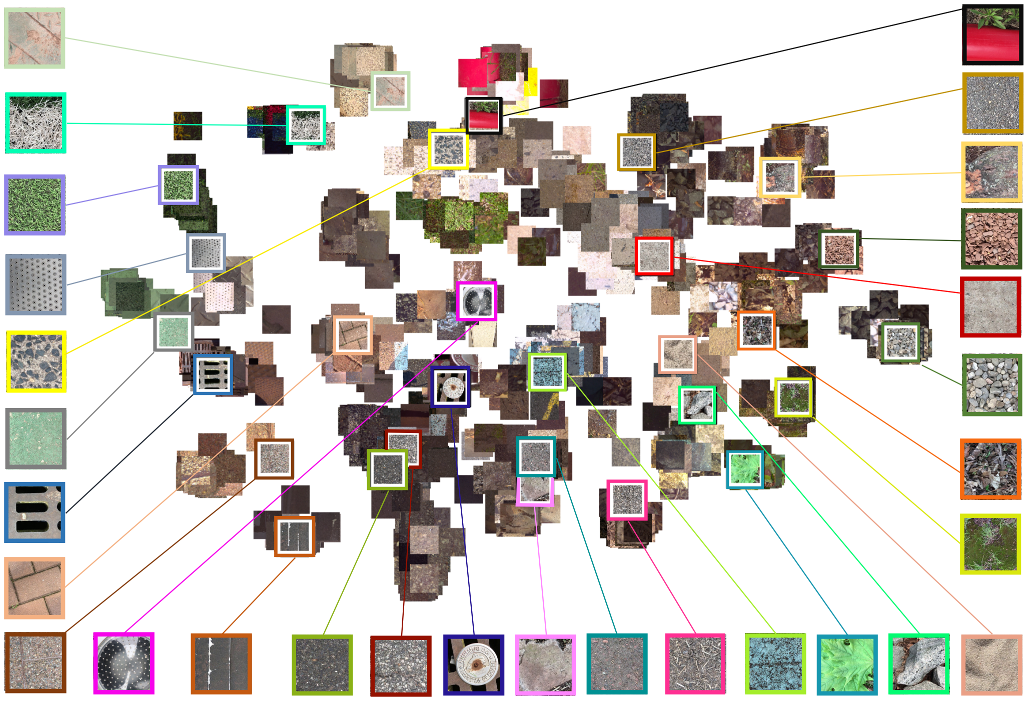


Figure 3.2: The result of texture manifold by DEP-manifold. Images with color frames are images in test set. The material classes are (from upper left to counter-clockwise): plastic cover, painted turf, turf, steel, stone-cement, painted cover, metal cover, brick, stone-brick, glass, sandpaper, asphalt, stone-asphalt, aluminum, paper, soil, mulch, painted asphalt, leaves, limestone, sand, moss, dry leaves, pebbles, cement, shale, roots, gravel and plastic. Not all classes are shown here for space limitations.

of spatial information (pooling layers). This structure is designed for object recognition, scene understanding, face recognition, and applications where spatial order is critical for classification. However, texture recognition uses an orderless component to provide invariance to spatial layout [4, 61, 23].

In classic approaches for texture modeling, images are filtered with a set of handcrafted filter banks followed by grouping the outputs into texton histograms [15, 16, 17, 18], or bag-of-words [19, 20]. Later, Cimpoi *et al.*[23] introduce FV-CNN that replace the handcrafted filter banks with pre-trained convolutional layers for the feature extractor, and achieve state-of-the-art results. Recently, Zhang *et al.*[4] introduce Deep

Texture Encoding Network (Deep-TEN) that ports the dictionary learning and feature pooling approaches into the CNN pipeline for an end-to-end material/texture recognition network. Recognition algorithms that focus on texture details work well for images containing only a single material. But for "images in the wild", homogeneous surfaces rarely fill the entire field-of-view, and many materials exhibit regular structure.

For texture recognition, since surfaces are not completely orderless, *local spatial order* is an important cue for recognition as illustrated in Figure 3.1. Just as semantic segmentation balances local details and global scene context for pixelwise recognition [62, 63, 64, 65, 66, 67], we design a network to balance both an orderless component and ordered spatial information.

As illustrated in Figure 3.3, we introduce a Deep Encoding Pooling Network (DEP) that leverages an orderless representation and local spatial information for recognition. Outputs from convolutional layers are fed into two feature representation layers jointly; the encoding layer [4] and the global average pooling layer. The encoding layer is employed to capture texture appearance details and the global average pooling layer accumulates spatial information. Features from the encoding layer and the global average pooling layer are processed with bilinear models [68]. We apply DEP to the problem of ground terrain recognition using an extended GTOS dataset [69]. The resultant network shows excellent performance not only for GTOS, but also for more general databases (MINC [66] and DTD [13]).

For ground terrain recognition, many class boundaries are ambiguous. For example, "asphalt" class is similar to "stone-asphalt" which is an aggregate mix of stone and asphalt. The class "leaves" is similar to "grass" because most of the example images for "leaves" in the GTOS database have grass in the background. Similarly, the grass images contain a few leaves. Therefore, it is of interest to find not only the class label but also the closest classes, or equivalently, the position in the manifold. We introduce a new texture manifold method, DEP-manifold, to find the relationship between newly captured images and images in dataset.

The t-Distributed Stochastic Neighbor Embedding (t-SNE) [70] provides a 2D embedding and Barnes-Hut t-SNE [6] accelerates the original t-SNE from $\mathcal{O}(n^2)$ to $\mathcal{O}(n \log n)$.

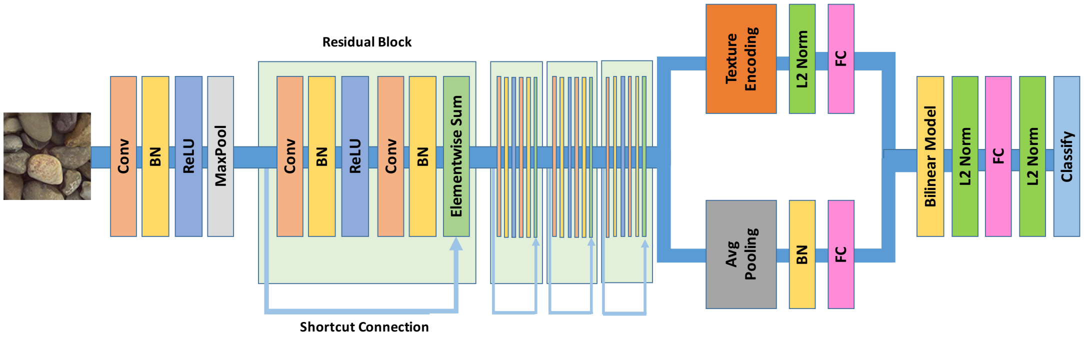


Figure 3.3: A Deep Encoding Pooling Network (DEP) for material recognition. Outputs from convolutional layers are fed into the encoding layer and global average pooling layer jointly and their outputs are processed with bilinear model.

Both t-SNE and and Barnes-Hut t-SNE are non-parametric embedding algorithms, so there is no natural way to perform out-of-sample extension. Parametric t-SNE [71] and supervised t-SNE [72, 73] introduce deep neural networks into data embedding and realize non-linear parametric embedding. Inspired by this work, we introduce a method for texture manifolds that treats the embedded distribution from non-parametric embedding algorithms as an output, and use a deep neural network to predict the manifold coordinates of a texture image directly. This texture manifold uses the features of the DEP network and is referred to as DEP-manifold.

The training set is a ground terrain database (GTOS) [69] with 31 classes of ground terrain images (over 30,000 images in the dataset). Instead of using images from the GTOS dataset for testing, we collect GTOS-mobile, 81 ground terrains videos of similar terrain classes captured with a hand-held mobile phone and with arbitrary lighting/viewpoint. Our motivation is as follows: The training set (GTOS) is obtained in a comprehensive manner (known distance and viewpoints, high-res caliabrated camera) and is used to obtain knowledge of the scene. The test set is obtained under very different and more realistic conditions (a mobile imaging device, handheld video, uncalibrated capture). Training with GTOS and testing with GTOS-mobile enables evaluation of knowledge transfer of the network.

3.2 Related Work

Tenenbaum and Freeman [68] introduce bilinear models to process two independent factors that underly a set of observations. Lin *et al.*[3] introduce the Bilinear CNN models that use outer product of feature maps from convolutional layers of two CNNs and reach state-of-the-art for fine grained visual recognition. However, this method has two drawbacks. First, bilinear models for feature maps from convolutional layers require that pairs of features maps have compatible feature dimensions, i.e. the same height and width. The second drawback is computational complexity; this method computes the outer product at each location of the feature maps. To utilize the advantage of bilinear models and overcome these drawbacks, we employ bilinear models for outputs from fully connected layers. Then, outputs from fully connected layers can be treated as vectors, and there is no dimensionality restriction for the outer product of two vectors.

Material recognition is a fundamental problem in computer vision, the analysis of material recognition has varied from small sets collected in lab settings such as KTH-TIPS [74] and CuRET [33], to large image sets collected in the wild [1, 66, 13, 69]. The size of material datasets have also increased from roughly 100 images in each class [1, 13] to over 1000 images in each class [69, 66]. The Ground Terrain in Outdoor Scenes (GTOS) dataset has been used with angular differential imaging [69] for material recognition based on angular gradients. For our work, single images are used for recognition without variation in viewing direction, so reflectance gradients are not considered.

For many recognition problems, deep learning has achieved great success, such as face recognition [75, 76, 77], action recognition [78, 79] and disease diagnosis [80]. The success of deep learning has also transferred to material recognition. We leverage a recent texture encoding layer [4] that ports dictionary learning and residual encoding into CNNs. We use this texture encoding layer as a component in our network to capture orderless texture details.

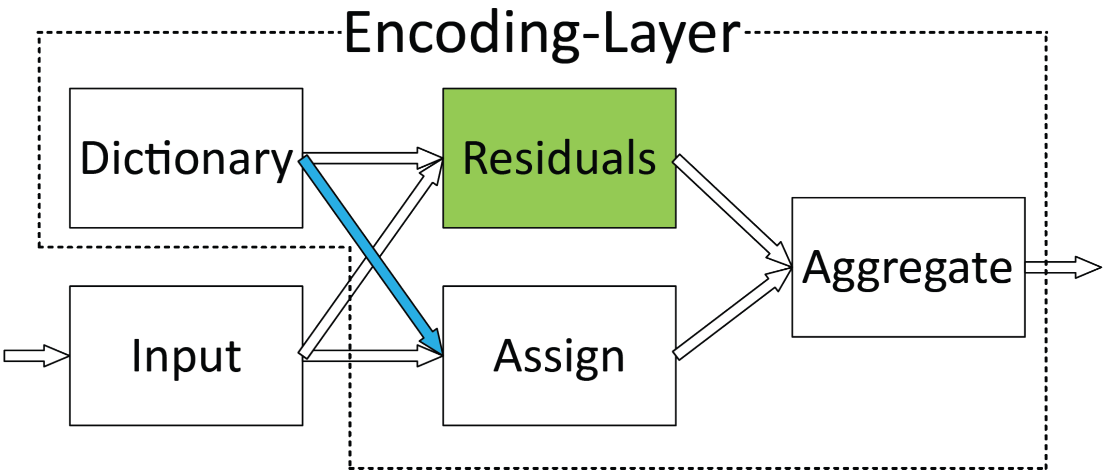


Figure 3.4: The *Encoding Layer* learns an inherent *Dictionary*. The *Residuals* are calculated by pairwise difference between visual descriptors of the input and the codewords of the dictionary. Weights are *assigned* based on pairwise distance between descriptors and codewords. Finally, the residual vectors are *aggregated* with the assigned weights.

3.3 Methods

3.3.1 Encoding Layer

For residual encoding model, given a set of N visual descriptors $X = \{x_1, ..x_N\}$ and a learned codebook $C = \{c_1, ...c_K\}$ containing K codewords that are D-dimensional, each descriptor x_i can be assigned with a weight a_{ik} to each codeword c_k and the corresponding residual vector is denoted by $r_{ik} = x_i - c_k$, where $i = 1, ...N$ and $k = 1, ...K$. Given the assignments and the residual vector, the residual encoding model applies an aggregation operation for every single codeword c_k:

$$e_k = \sum_{i=1}^{N} e_{ik} = \sum_{i=1}^{N} a_{ik} r_{ik}. \tag{3.1}$$

The resulting encoder outputs a fixed length representation $E = \{e_1, ...e_K\}$ (independent of the number of input descriptors N).

As shown in Figure 3.5, the traditional visual recognition approach can be partitioned into feature extraction, dictionary learning, feature pooling (encoding) and classifer learning. The texture encoding layer [4] ports the dictionary learning and

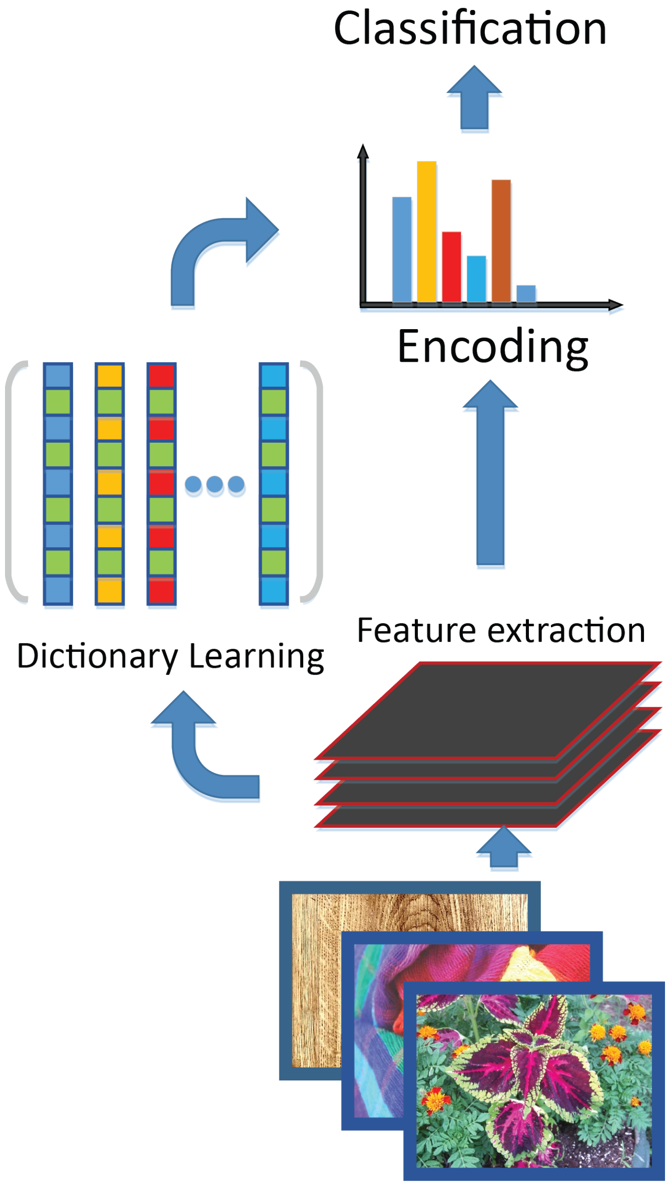


Figure 3.5: Pipelines of classic computer vision approaches. Given in put images, the local visual appearance is extracted using hang-engineered features (SIFT or filter bank responses). A dictionary is then learned off-line using unsupervised grouping such as K-means. An encoder (such as BoWs or Fisher Vector) is built on top which describes the distribution of the features and output a fixed-length representations for classification.

residual encoding into a single layer of CNNs, which we refer to as the *Encoding Layer*. The Encoding Layer simultaneously learns the encoding parameters along with with an inherent dictionary in a fully supervised manner. The inherent dictionary is learned from the distribution of the descriptors by passing the gradient through assignment weights. During the training process, the updating of extracted convolutionalfeatures can also benefit from the encoding representations.

Consider the assigning weights for assigning the descriptors to the codewords. Hard-assignment provides a single non-zero assigning weight for each descriptor x_i, which corresponds to the nearest codeword. The k-th element of the assigning vector is given by $a_{ik} = \mathbb{1}(\|r_{ik}\|^2 = min\{\|r_{i1}\|^2, ...\|r_{iK}\|^2\})$ where $\mathbb{1}$ is the indicator function (outputs 0 or 1). Hard-assignment doesn't consider the codeword ambiguity and also makes the model non-differentiable. Soft-weight assignment addresses this issue by assigning a descriptor to each codeword [81]. The assigning weight is given by

$$a_{ik} = \frac{\exp(-\beta\|r_{ik}\|^2)}{\sum_{j=1}^{K}\exp(-\beta\|r_{ij}\|^2)}, \tag{3.2}$$

where β is the smoothing factor for the assignment.

Soft-assignment assumes that different clusters have equal scales. Inspired by guassian mixture models (GMM), we further allow the smoothing factor s_k for each cluster center c_k to be learnable:

$$a_{ik} = \frac{\exp(-s_k\|r_{ik}\|^2)}{\sum_{j=1}^{K}\exp(-s_j\|r_{ij}\|^2)}, \tag{3.3}$$

which provides a finer modeling of the descriptor distributions. The Encoding Layer concatenates the aggregated residual vectors with assigning weights (as in Equation 3.1). As is typical in prior work [82], the resulting vectors are normalized using the $L2$-norm.

The Encoding Layer is a directed acyclic graph as shown in Figure 3.4, and all the components are differentiable $w.r.t$ the input X and the parameters (codewords $C = \{c_1, ...c_K\}$ and smoothing factors $s = \{s_1, ...s_k\}$). Therefore, the Encoding Layer can be trained end-to-end by standard SGD (stochastic gradient descent) with back-propagation.

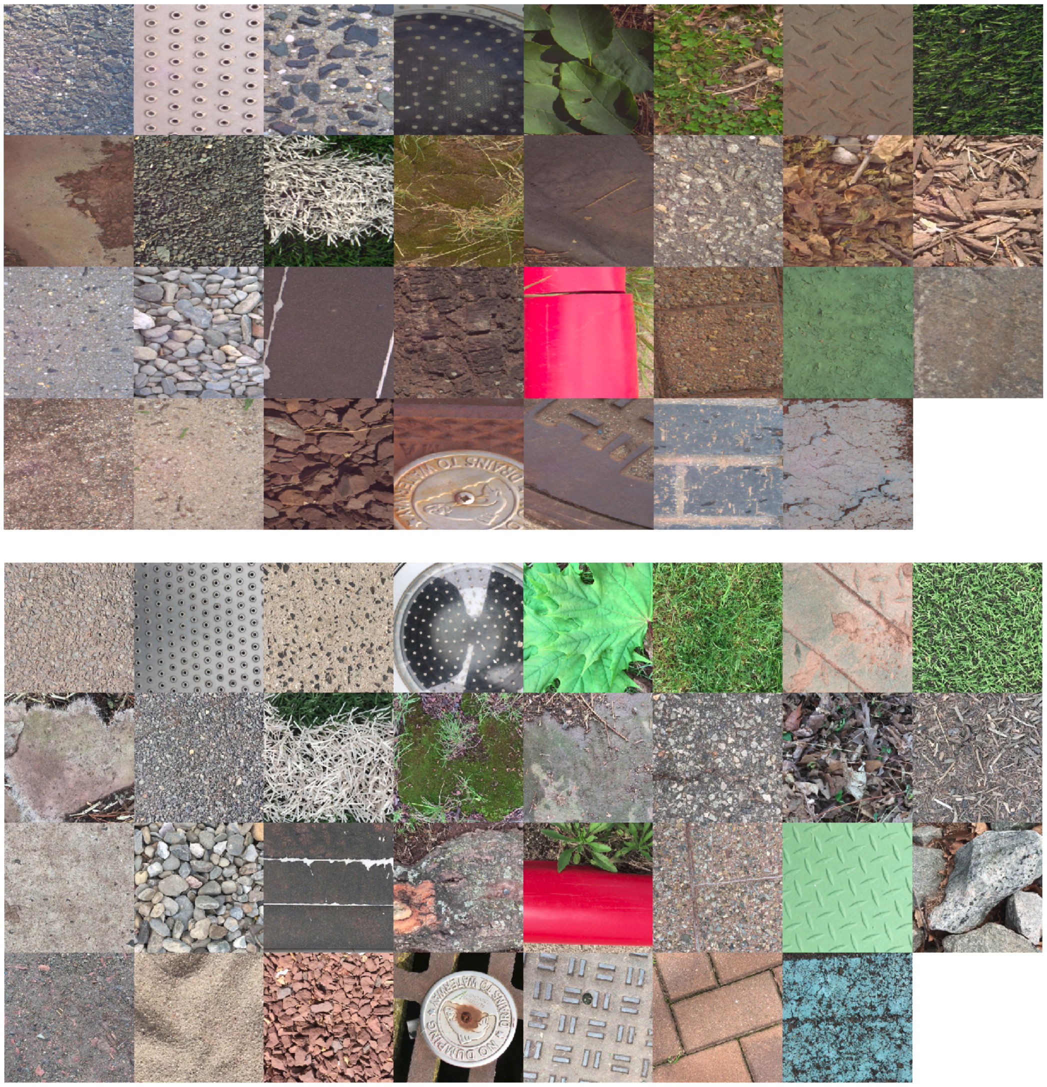


Figure 3.6: Comparison of images from the GTOS dataset (top) and GTOS-mobile (down) video frames. The training set is the ground terrain database (GTOS) with 31 classes of ground terrain images (over 30,000 images in the dataset). GTOS is collected with calibrated viewpoints. GTOS-mobile, consists of 81 videos of similar terrain classes captured with a handheld mobile phone and with arbitrary lighting/viewpoint. A total of 6066 frames are extracted from the videos with a temporal sampling of approximately 1/10th seconds. The figure shows individual frames of 31 ground terrain classes.

3.3.2 Bilinear Models

In many vision problems, a set of observations is often influenced by two or more independent factors, we want to infer more than one hidden factors wich interact to produce the observations. Bilinear models are two-factor models such that their outputs are linear in one factor if the other factor is constant [83]. The factors in bilinear models balance the contributions of the two components. Let a^t and b^s represent the material texture information and spatial information with vectors of parameters and with dimensionality I and J. The bilinear function Y^{ts} is given by

$$Y^{ts} = \sum_{i=1}^{I} \sum_{j=1}^{J} w_{ij} a_i^t b_j^s, \tag{3.4}$$

where w_{ij} is a learnable weight to balance the interaction between material texture and spatial information. The outer product representation captures a pairwise correlation between the material texture encodings and spatial observation structures.

Bilinear model has been used in many vision tasks. For example, Pirsiavash *et al.*[84] demonstrate bilinear SVMs on difficult programs of people detection in video sequences and action classification of video sequences, achieving state-of-the-art results in both. Recently, Lin *et al.*[3] integrate the bilinear model into Convolutional Neural Networks by introducing the Bilinear CNN models that consists of two CNN feature extractors whose outputs are multiplied using outer product at each location of the image and pooled to obtain an image descriptor, they reach state-of-the-art for fine grained visual recognition.

3.3.3 Deep Encoding Pooling Network (DEP)

Our Deep Encoding Pooling Network (DEP) is shown in Figure 3.3. As in prior transfer learning algorithms [3, 4], we employ convolutional layers with non-linear layers from ImageNet [56] pre-trained CNNs as feature extractors. Outputs from convolutional layers are fed into the texture encoding layer and the global average pooling layer jointly. Outputs from the texture encoding layer preserve texture details, while outputs from the global average pooling layer preserve local spatial information. The dimension of outputs from the texture encoding layer is determined by the codewords N and

layer name	output size	encoding-pooling
conv1	112×112×64	7×7, stride 2
conv2_x	56×56×64	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$
conv3_x	28×28×128	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$
conv4_x	14×14×256	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 2$
conv5_x	7×7×512	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 2$
encoding / pooling	8 x 512 / 512	8 codewords / ave pool
fc1_1 / fc1_2	64 / 64	4096×64 / 512×64
bilinear mapping	4096	-
fc2	128	4096×128
classification	n classes	128×n

Table 3.1: The architecture of Deep Encoding Pooling Network based on 18-layer ResNet [2]. The input image size is 224×224.

the feature maps channel C (N×C). The dimension of outputs from the global average pooling layer is determined by the feature maps channel C. For computational efficiency and to robustly combine feature maps with bilinear models, we reduce feature maps dimension with fully connected layers for both branches. Feature maps from the texture encoding layer and the global average pooling layer are processed with a bilinear model and followed by a fully connected layer and a classification layer with non-linearities for classification. Table 3.1 is an instantiation of DEP based on 18-layer ResNet [2]. We set 8 codewords for the texture encoding layer. The size of input images are 224×224. Outputs from CNNs are fed into the texture encoding layer and the global average pooling layer jointly. The dimension of outputs from the texture encoding layer is $8 \times 512 = 4096$ and the dimension of outputs from global average pooling layer is 512. We reduce the dimension of feature maps from the deep encoding layer and the global average pooling layer to 64 via fully connected layers. The dimension of outputs from bilinear model is $64 \times 64 = 4096$. Following prior works [4, 85], resulting vectors from the texture encoding layer and bilinear model are normalized with L2 normalization.

The texture encoding layer and bilinear models are both differentiable. The overall architecture is a directed acyclic graph and all the parameters can be trained by back propagation. Therefore, the Deep Encoding Pooling Network is trained end-to-end using stochastic gradient descent with back-propagation.

3.3.4 Texture Encoded Angular Network (TEAN)

Adapting the DEP to the RGB image branch into the Differential Angular Imaging Network (DAIN) introduced in Section 2, we introduce the Texture Encoded Angular Network (TEAN). The detailed network is shown in Figure 3.7. We develop a two-stream convolutional neural network, one branch input is the differential angular image, representing the material reflectance information. The other branch input is the RGB image, representing the orderless texture details and ordered spatial information. For the color image branch, we utilize the Deep Encoding Pooling Network (DEP) to balance the orderless texture component and ordered spatial information. As in DAIN, we combine feature maps at both intermediate layer and final prediction layer. With the

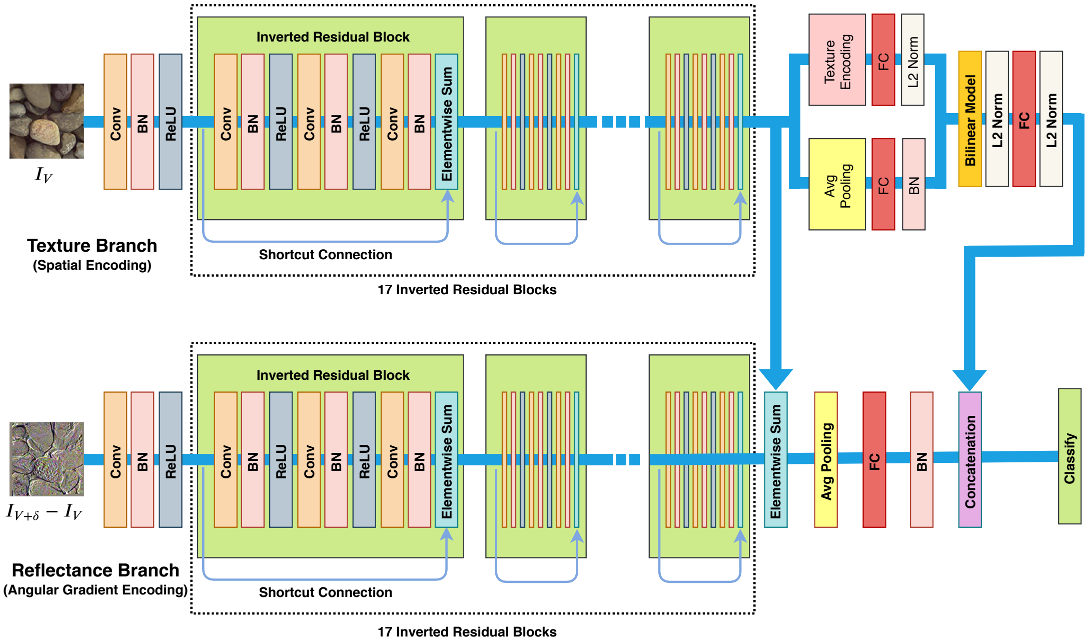


Figure 3.7: A Texture Encoded Angular Network (TEAN) for material recognition. The input to the reflectance branch is the differential angular image, which captures material reflectance information via angular gradients. The input to the texture branch is the RGB color image, to provide the ordered and orderless spatial information. For the texture branch, we utilize DEP to balance the orderless texture component and ordered spatial information. The overall architecture of TEAN enables material classification using angular reflectance information, orderless texture and ordered spatial structure.

proposed Texture Encoded Angular Network (TEAN), we take advantage of material reflectance information, orderless texture details and ordered spatial information for ground terrain material recognition. This combination of angular cues, orderless spatial cues and ordered spatial cues leads to improved recognition results.

For quick inference in real applications, we employ ImageNet[56] pre-trained MobileNet V2[5] as backbone, which is specially designed for real-time inference. As shown in Figure 3.7, we employ ImageNet[56] pre-trained MobileNet V2[5] as the initial prediction unit. As in single view DAIN (Sum), we combine feature maps from the bottlenect8_x with element-wise sum as intermediate layer combination. Feature maps from color images are fed into the texture encoding layer and the global average pooling layer jointly, followed by bilinear model and fully connected layer, the output from fully connected layer is a 128-D vector. The elemen-wise summed feature maps are fed into a fully connected layer for dimension reduction, the output is also a 128-D vector. These two 128-D vectors are concatenated and fed into classify layer for material classification.

3.4 DEP Experiments

3.4.1 Baseline Network

We compare the DEP network with the following three methods based on ImageNet [86] pre-trained 18-layer ResNet [2]: (1) CNN with ResNet, (2) CNN with Deep-Ten and(3) CNN with bilinear models. All three methods support end-to-end training. For equal comparison, we use an identical training and evaluation procedure for each experiment.

CNN with global average pooling (ResNet)

We follow the standard procedure to fine-tune pre-trained ResNet, by replacing the last 1000-way fully connected layer with the output dimension of 31 (the number of material classes). The global average pooling works as feature pooling that encodes the 7×7×512 dimensional features from the 18-layer pre-trained ResNet into a 512 dimensional vector.

CNN with texture encoding (Deep-TEN)

The Deep Texture Encoding Network (Deep-TEN) [4] integrates Encoding Layer on top of convolutional layers, which ports the entire dictionary learning and encoding pipeline into a single model. Deep-TEN provides an end-to-end learning framework, where the inherent visual vocabularies are learned directly from the loss function. The features, dictionaries, encoding representation and the classifier are all learned simultaneously. Deep-TEN shows superior performance as compared to state-of-the-art methods using gold-standard databases such as MINC-2500, Flickr Material Database, KTH-TIPS-2b

	ResNet [2]	Bilinear CNN[3]	Deep-TEN[4]	DEP (ours)
Single scale	70.82	72.03	74.22	**76.07**
Multi scale	73.16	75.43	76.12	**82.18**

Table 3.2: Comparison our Deep Encoding Pooling Network (DEP) with ResNet (left) [2], Bilinear CNN (mid) [3] and Deep-TEN (right) [4] on GTOS-mobile dataset with single scale and multi scale training. For ResNet, we replace the 1000-way classification layer with a new classification layer, the output dimension of new classification layer is 31.

and GTOS. In original paper, we embed the texture encoding layer on top of the 50-layer pre-trained ResNet [2]. For quick inference and to make an equal comparison, we replace the 50-layer ResNet with 18-layer ResNet. Same as [4], we reduce the number of CNN streams outputs channels from 512 to 128 with a 1×1 convolutional layer. We replace the global average pooling layer in the 18-layer ResNet with texture encoding layer, set the number of codewords to 32 for experiments. Outputs from the texture encoding layer are normalized with L2 normalization. A fully connected layer with soft max loss follows the texture encoding layer for classification.

CNN with bilinear models (Bilinear-CNN)

Bilinear-CNN [3] employs bilinear models with feature maps from convolutional layers. Outputs from convolutional layers of two CNN streams are multiplied using outer product at each location and pooled for recognition. To make an equal comparison, we employ the 18-layer pre-trained ResNet as CNN streams for feature extractor. Feature maps from the last convolutional layer are pooled with bilinear models. The dimension of feature maps for bilinear models is 7×7×512 and the pooled bilinear feature is of size 512×512. The pooled bilinear feature is fed into classification layer for classification.

3.4.2 Dataset and Evaluation

Dataset

Extending the GTOS database [69], we collect *GTOS-mobile* consisting of 81 videos obtained with a mobile phone (Iphone SE) and extract 6066 frames as a test set. To simulate real world ground terrain collection, we walk through similar ground terrain regions in random order to collect the videos. Scale is changed arbitrarily by moving far or close and changes in viewing direction are obtained by motions in a small arc. The resolution of the videos is 1920×1080, and we resize the short edge to 256 while keeping the aspect ratio for experiments. As a result, the resolution of the resized images are 455×256. Some materials in GTOS were not accessible due to weather, therefore we removed the following classes: dry grass, ice mud, mud-puddle, black ice and snow from the GTOS dataset. Additionally, we merged very similar classes of asphalt and metal. The original GTOS set is 40 classes, as shown in Figure 3.6, there are 31 material classes in the modified dataset. The class names are (in the order of top-left to bottom-right): asphalt, steel, stone-cement, glass, leaves, grass, plastic cover, turf, paper, gravel, painted turf, moss, cloth, stone-asphalt, dry leaves, mulch, cement, pebbles, sandpaper, roots, plastic, stone-brick, painted cover, limestone, soil, sand, shale, aluminum, metal cover, brick, painted asphalt.

Multi-scale Training

Images in the GTOS dataset were captured from a fixed distance between the camera and ground terrain, however the distance between the camera and ground terrain can be arbitrary in real world applications. We infer that extracting different resolution patches with different aspect ratio from images in GTOS simulate observing materials at different distance and viewing angle will be helpful for recognition. So for image pre-processing, instead of directly resizing the full resolution images into 256×256 as [69], we resize the full resolution images into different scales, and extract 256×256 center patches for experiment. Through empirical validation, we find that resizing the full resolution images into 256×256, 384×384 and 512×512 works best.

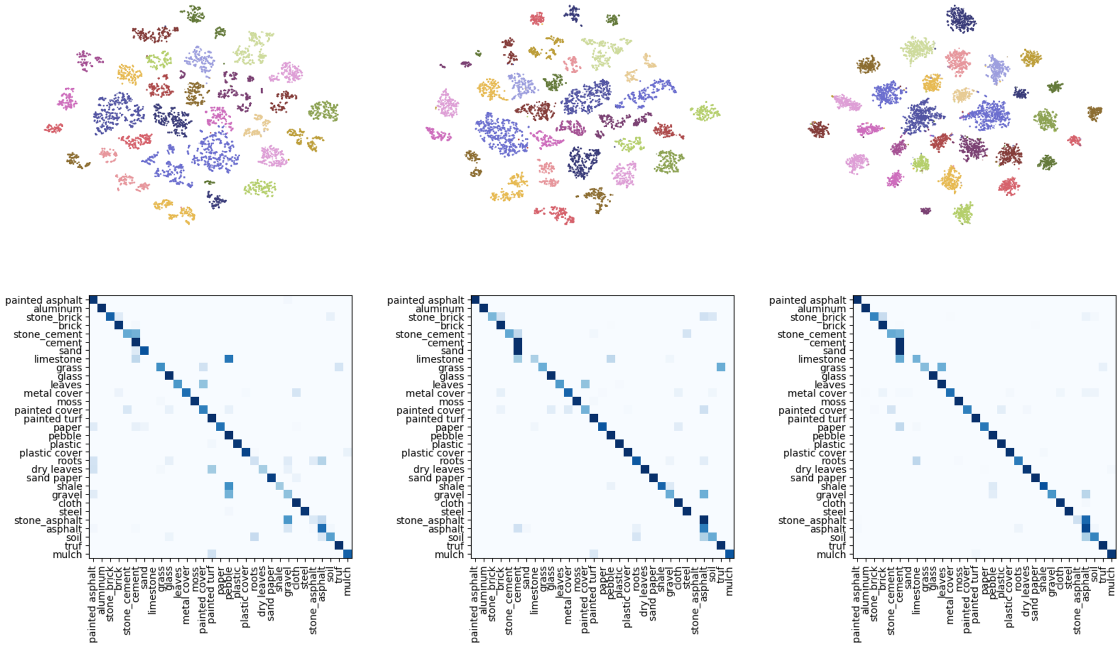
Training procedure

We employ an identical data augmentation and training procedure for experiments. For single scale training experiment, we resize the full resolution images into 384×384 and extract 256×256 center patches as training set. For multi scale training experiment, we resize the full resolution images into 256×256, 384×384 and 512×512, and extract 256×256 center patches as training set. For the training section data augmentation, following prior work [87, 4], we crop a random size (0.8 to 1.0) of the original size and a random aspect ratio (3/4 to 4/3) of the original aspect ratio, resize the cropped patches to 224×224 for experiment. All images are pre-processed by subtracting a per color channel mean value and normalized to unit variance with a 50% chance horizontal flip. The learning rate of newly added layers is 10 times of the pre-trained layers. The experiment starts with learning rate at 0.01, momentum 0.9, batch size 128; the learning rate decays by factor of 0.1 for every 10 epochs, and is finished after 30 epochs.

3.4.3 Recognition Results

Evaluation on GTOS-mobile

Table 3.2 is the classification accuracy of fine-tuning ResNet [2], bilinear CNN [3], Deep-TEN [4] and the proposed DEP on the GTOS-mobile dataset. When comparing the performance of single-scale and multi-scale training, multi-scale training outperforms single-scale training for all approaches. It proves our inference that extracting different resolution patches with different aspect ratio from images in GTOS to simulate observing materials at different distance and viewing angle will be helpful for recognition. The multi-scale training accuracy for combined spatial information and texture details (DEP) is 82.18%. That's 9.02% better than only focusing on spatial information (ResNet) and 6% better than only focusing on texture details (Deep-TEN). To gain insight into why DEP outperforms ResNet and Deep-TEN for material recognition, we visualize the features before classification layers of ResNet, Deep-TEN and DEP with Barnes-Hut t-SNE [6] . We randomly choose 10000 images from training set for the experiment. The result is shown in Figure 3.8. Notice that DEP separates classes farther



(a) ResNet

(b) Deep-TEN

(c) DEP (ours)

Figure 3.8: The Barnes-Hut t-SNE [6] and confusion matrix of three material recognition models: ResNet (left), Deep-TEN (mid) and DEP (right). For Barnes-Hut t-SNE, we randomly choose 10000 images from training set and extract features before classification layers of three models for experiment. We see that DEP separates and clusters the classes better. Some classes are misclassified, however, they are typically recognized as a nearby class. (Dark blue represents higher values and light blue represents lower values in the confusion matrix.)

apart and each class is clustered more compactly.

Evaluation on MINC and DTD Dataset

To show the generality of DEP for material recognition, we experiment on two other material/texture recognition datasets: Describable Textures Database (DTD) [13] and Materials in Context Database (MINC) [66]. For an equal comparison, we build DEP based on a 50-layer ResNet [2], the feature maps channels from CNN streams are reduced from 2048 to 512 with a 1×1 convolutional layer. The result is shown in Table 3.3, DEP outperforms the state-of-the-art on both datasets. Note that we only

Method	DTD[13]	Minc-2500[66]
FV-CNN [23]	72.3%	63.1%
Deep-TEN [4]	69.6%	80.4%
DEP (ours)	**73.2**%	**82.0**%

Table 3.3: Comparison with state-of-the-art algorithms on Describable Textures Dataset (DTD) and Materials in Context Database (MINC).

experiment with single scale training. As mentioned in [3], multi-scale training is likely to improve results for all methods.

3.5 Texture Manifold

Inspired by Parametric t-SNE [71] and supervised t-SNE [72, 73], we introduce a parametric texture manifold approach that learns to approximate the embedded distribution of non-parametric embedding algorithms [70, 6] using a deep neural network to directly predict the 2D manifold coordinates for the texture images. We refer to this manifold learning method using DEP feature embedding as DEP-manifold. Following prior work [72, 71], the deep neural network structure is depicted in Figure 3.9. Input features are the feature maps before the classification layer of DEP, which means each image is represented by a 128 dimensional vector. Unlike the experiment in [72, 71], we add non-linear functions (Batch Normalization and ReLU) before fully connected layers, and we do not pre-train the network with a stack of Restricted Boltzmann Machines (RBMs) [88]. We train the embedding network from scratch instead of the three-stage training procedure (pre-training, construction and fine-tuning) in parametric t-SNE and supervised t-SNE. We randomly choose 60000 images from the multi-scale GTOS dataset for the experiment. We experiment with DEP-parametric t-SNE, and DEP-manifold based on outputs from the last fully connected layer of DEP.

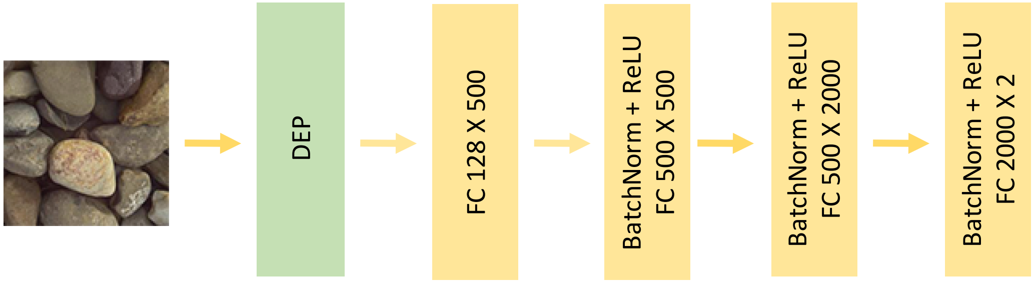


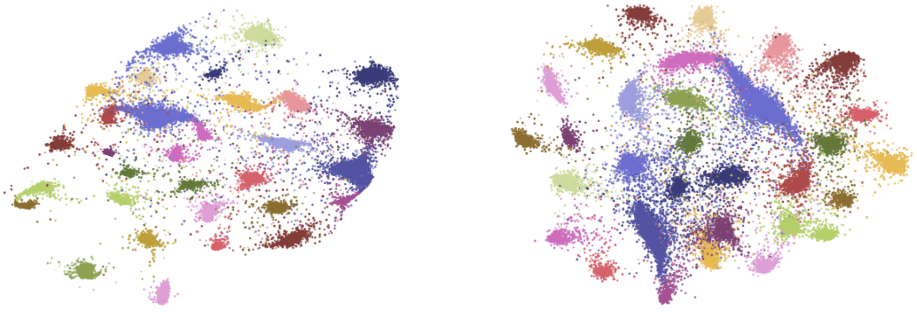
Figure 3.9: The deep network for texture manifold, we employ DEP as feature extractor, outputs from the last fully connected layer of DEP works as input for texture embedding.

3.5.1 Implementation

For the DEP-manifold, we employ Barnes-Hut t-SNE [6] as a non-parametric embedding to build the embedded distribution. Following prior setting [6], we set perplexity to 30 and the output dimension of PCA to 50 for the experiment. For training the deep embedding network, we experiment with batch size 2048 and the parameters of the fully connected layers are initialized with the Xavier distribution [89]. We employ L2 loss as the objective function for the experiment. The initial learning rate is 0.01, and decays by a factor of 0.1 every 30 epochs. The experiment is finished after 80 epochs. On an NVIDIA Titan X card, the training takes less than 5 minutes.

3.5.2 Texture Manifold

The texture manifold results are shown in Figure 3.10. For the embedded distribution of DEP-Parametric t-SNE, the classes are distributed unevenly with crowding in some areas and sparseness in others. The DEP-manifold has a better distribution of classes within the 2D embedding. We illustrate the texture manifold embedding by randomly choosing 2000 images from training set to get the embedded distribution; then we embed images from test set into the DEP-manifold. Note that the test set images are not used in the computation of the DEP-manifold. The result is shown in Figure 3.2. By



(a) DEP-parametric t-SNE

(b) DEP-manifold

Figure 3.10: Comparison the performance between DEP-parametric t-SNE and DEP-manifold with 60000 images from multi-scale GTOS dataset. For the embedded distribution of DEP-Parametric t-SNE, the classes are distributed unevenly with crowding in some areas and sparseness in others. The DEP-manifold has a better distribution of classes within the 2D embedding.

observing the texture manifold, we find that for some classes, although the recognition accuracy is not perfect, the projected image is within the margin of the correct class, such as cement and stone-cement. Based on the similarity of material classes on the texture manifold, we build the confusion matrix for material recognition algorithms as shown in Figure 3.8. For visualization, the one dimensional ordering of the confusion matrix axes are obtained from a one-dimensional embedding of the 2D manifold so that neighboring classes are close. Observe that for the DEP recognition (Figure 3.8 c), there are very few off-diagonal elements in the confusion matrix. And the off-diagonal elements are often near diagonal indicating find when these images are misclassified, they are recognized as closely-related classes.

3.6 TEAN Experiments

In this section, we compare the performance of DAIN, DEP and TEAN framework for material recognition on the GTOS dataset. We evaluate the performance of TEAN and verify the performance of multi-scale training. To gain insight into the performance, we construct the confusion matrix and visualize the features before classification layers with BarnesHut t-SNE [6] for MobileNet, DEP, DAIN and TEAN.

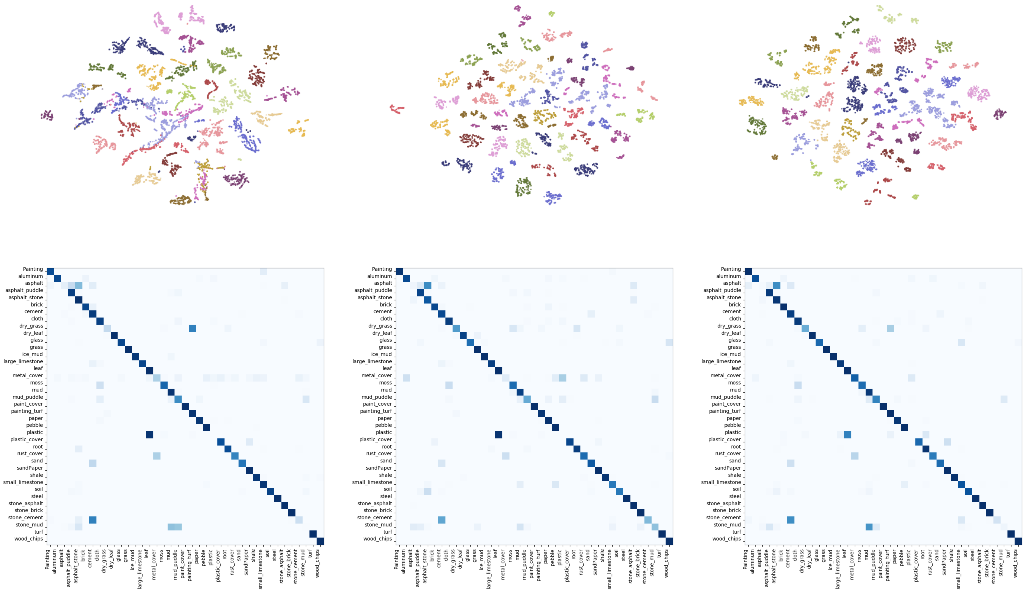
Training procedure We use the same 5 training and testing splits as Section 2. Each input image from our GTOS database is resized into 240 × 240. Since the snow class only has 2 samples in the dataset, we omit this class from experiments.

The number of parameters for recent mobile platform designed MobileNet V2 [5] is much less comparing with VGG-M [90]. So for MobileNet V2 [5] based DAIN, we train the network end-to-end on GTOS dataset directly. We employ the augmentation method that horizontally and vertically stretch training images within ±10%, with an optional 50% horizontal and vertial mirror flips. The images are randomly cropped into 224 × 224 material patches. All images are pre-processed by subtracting a per color channel mean and normalizing for unit variance.

Following prior works [69, 60], for the two-branch MobileNet V2 model, we experiment with batch size 64 and learning rate starting from 0.01 which is reduced by a factor of 0.1 when the training accuracy saturates. We augment training data with randomly stretch training images by ±25% horizontally and vertically, and also horizontal and vertical mirror flips with 50% chance. The images are randomly cropped into 224 × 224 material patches. We first backpropagate only to feature maps combination layer for 3 epochs, then fine tunes all layers. We employ the same augmentation method for the multiview images of each material surface. We randomly select the first viewpoint image, then subsequent $N = 4$ view point images are selected for experiments.

3.6.1 TEAN Recognition Results

Table 3.4 is the mean classification accuracy comparison of MobileNet V2 based single view or multiview CNN fine-tune, DEP, DAIN and TEAN. As in DAIN, we experiment with voting and pooling to combine the multiview image set. From the result we can see that multiview TEAN (Sum/pooling) performs best, the recognition accuracy is 87.6%, which is 5.1% better than multiview CNN, voting baseline. Also the recognition performance for TEAN outperforms DAIN in both single view and multiview.



(a) DEP (b) DAIN (c) TEAN

Figure 3.11: The Barnes-Hut t-SNE [6] and confusion matrix of four material recognition models based on GTOS : DEP (left), DAIN (mid) and TEAN (right). For Barnes-Hut t-SNE [6], we employ images from validation set and extract features before classification layers of four models for experiment. We see that TEAN separates and clusters the classes better. (Dark blue represents higher values and light blue represents lower values in the confusion matrix.)

3.6.2 Multi-scale Training

Multi-scale training is a common image augmentation trick to simulate observing materials at different distances [55, 4, 60]. We also experiment this with our GTOS dataset. We resize images into different resolutions, and randomly crop 224×224 patches for training. Following prior works [4, 60], We experiment TEAN with two groups of resolution settings: (256×256, 384×384, 512×512) and (224×224, 246×246, 268×268). For training/testing split 1, the recognition accuracy is 81.93% and 82.03% respectively, it is lower than the single view TEAN, in which the accuracy is 82.87%. Although the result is contrary with prior works [55, 4, 60], that simulating observing materials at different distances with multi-scale training is helpful for performance, we think the

Method	Accuracy
single view CNN	$80.4_{\pm 3.2}$
multiview CNN, voting	$82.5_{\pm 2.8}$
single view DEP	$83.3_{\pm 2.1}$
multiview DEP, voting	$85.8_{\pm 1.9}$
single view DAIN (Sum)	$82.5_{\pm 2.3}$
multiview DAIN (Sum/voting)	$85.8_{\pm 2.6}$
multiview DAIN (Sum/pooling)	$86.2_{\pm 2.5}$
single view TEAN (Sum)	$84.7_{\pm 2.1}$
multiview TEAN (Sum/voting)	$87.4_{\pm 2.3}$
multiview TEAN (Sum/pooling)	$87.6_{\pm 2.0}$

Table 3.4: Results comparing performance of CNN fine-tune, DEP, DAIN and TEAN based on MobileNet V2 [5].

result is meaningful for GTOS. Since images in the GTOS dataset are captured with a fixed distance between the camera and ground terrain, the observing distance is constant for all the images. We conclude the multi-scale training is not helpful for our GTOS dataset.

3.6.3 Confusion Matrix and Feature Visualization

To gain insight into why TEAN performs best for material recognition, based on training/testing split 1, we compute the confusion matrix of DEP, DAIN and TEAN and visualize features before classification layers with Barnes-Hut t-SNE [6]. For features visualization, we employ images from validation set and extract features before classification layers of four models for experiment. The result is shown in Figure 3.11. Notice that TEAN separates and clusters the classes better.

3.7 Summary and Conclusion

We have developed methods for recognition of ground terrain for potential applications in robotics and automated vehicles. We make three significant contributions in this paper: 1) introduction of Deep Encoding Pooling network (DEP) that leverages an orderless representation and local spatial information for recognition. When integrate DEP into DAIN, we use differential angular imaging, texture details and spatial information for material recognition, showing superior results for differential inputs as compared to original images; 2) Introduction of DEP-manifold that integrates DEP network on top of a deep neural network to predict the manifold coordinates of a texture directly; 3) Collection of the GTOS-mobile database comprised of 81 ground terrains videos of similar terrain classes as GTOS, captured with a handheld mobile phone to evaluate knowledge-transfer between different image capture methods but within the the same domain.

Chapter 4

Angular Luminance Networks for Material Segmentation

Moving cameras provide multiple intensity measurements per pixel, yet often semantic segmentation, material recognition and object recognition do not utilize this information. With basic alignment over several frames of a moving camera sequence, a distribution of intensities over multiple angles is obtained. It is well known from prior work that luminance histograms and the statistics of natural images provide a strong material recognition cue. We utilize per-pixel *angular luminance distributions* as a key feature in discriminating the material of the surface. The angle-space sampling in a multiview image sequence is an unstructured sampling of the underlying reflectance function of the material. For real-world materials there is significant intra-class variation that can be managed by building a angular luminance network (AngLNet). This network combines new angular reflectance cues from multiple images with more traditional spatial cues as in fully convolutional networks for semantic segmentation. We demonstrate the increased performance of AngLNet over prior state-of-the-art in material segmentation from drone video sequences and satellite imagery.

4.1 Background

Material recognition and segmentation is an important area of research in computer vision for providing more complete scene understanding. Materials play a fundamental role in numerous applications including asphalt for automated driving, tree-cover in fire risk assessment, path material (grass vs. concrete) for robot navigation, and landcover albedo analysis for climate studies. Material segmentation assigns a material id to every pixel in an image. Material segmentation differs from semantic segmentation, which assigns an object class label to every image pixel based on the object's shape, color

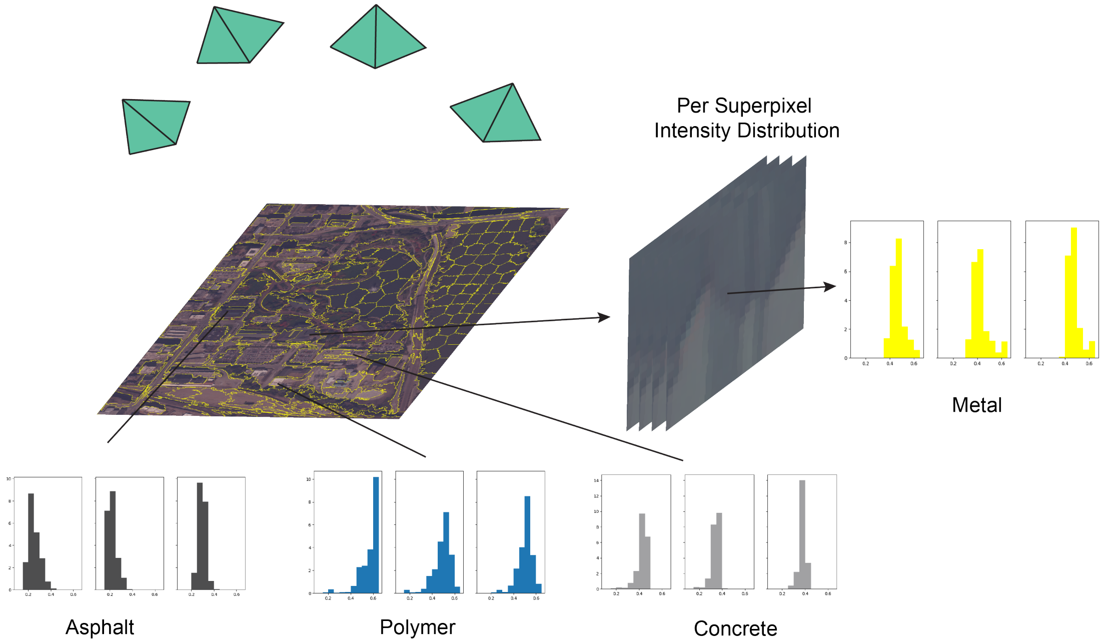


Figure 4.1: Distribution of intensities from multiview images provides a direct material cue. Multiview image sequences may be acquired from cameras mounted on vehicles, drones and satellites.

and position cues as well as global context information. Material segmentation focuses on texture and reflectance cues in an image. To underscore the difference between material and object-based semantic segmentation, consider that same semantic object can be made of different materials. For example, a building rooftop may be made of wood, metal, concrete, polymer or asphalt. Materials can often be distinguished using the bidirectional reflectance distribution function (BRDF). Traditional methods use a full BRDF [33, 91, 92] or dense partial samplings of the BRDF [21, 93, 94] to characterize a material. Capturing a full BRDF is rarely practical in applications. Multiview observations of a scene provide an opportunity to use reflectance for recognition, since aligned imagery can provide a vector of reflectance samples per pixel.

Early studies in material recognition from reflectance characteristics largely concentrated on per-image recognition, which predicts one material class for the entire image or region. But pixel-wise prediction is required for segmentation, and characterizing material appearance with a BRDF for each pixel is an insurmountable task. Also, in

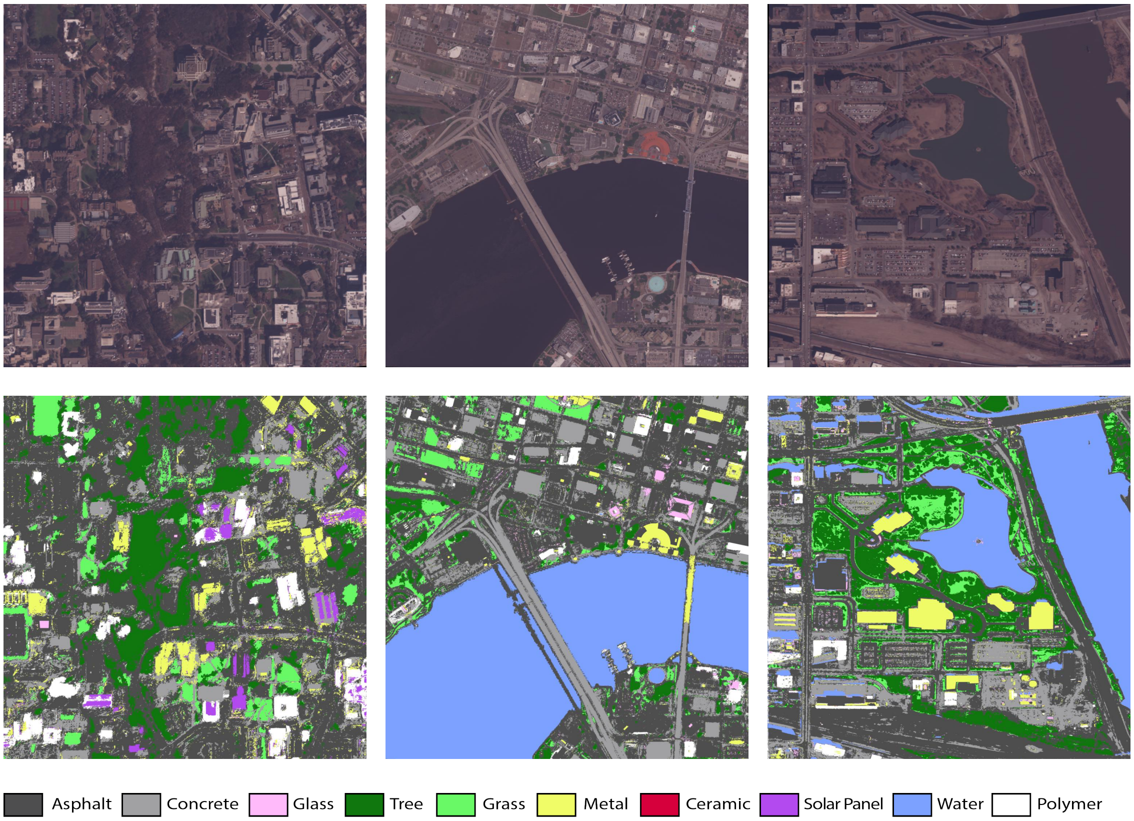


Figure 4.2: Overview of the satellite images and our labeled ground truth. We labeled the material classes of five different regions from San Diego, Jacksonville and Omaha, with 10 material classes: Asphalt, Concrete, Glass, Tree, Grass, Metal, Ceramic, Solar Panel, Water and Polymer. The height/width of labeled regions varies from 3000×3000 to 8000×8000.

many cases, we lack sufficient training data to formulate a probability distribution over the entire space of realizable BRDFs that fully capture the intraclass variation. The approach by Dror *et al.*[26] asserts that given a finite but arbitrary set of candidate reflectance functions, we can identify which one most closely represents an observed surface by a low-cost intensity distribution. These luminance histograms computed over a spatial region are powerful for material discrimination [27]. Inspired by this, we make use of a per-pixel *angular luminance histogram* representing the distribution of intensities observed per-pixel from all viewing angles in the multiview sequence. Taken by itself, the angular histogram cue is weak, but integrated in a meaningful way within a large deep network, this cue consistently provides a signficant performance boost for material-based segmentation. Moreover, in applications where multiview images are collected, the angular histogram is readily available with little additional cost.

MINC [66] is a pioneering large scale scene material segmentation dataset, that provides segment annotations for 23 material categories. However, the dataset provides single view images only. A common method to get multiview images is to extract multiview image sequences from videos. For example, Ma *et al.*capture multiview image sequences by using DVO-SLAM [95] with NYUDv2 RGB-D dataset [96] for semantic segmentation [97]. To our knowledge, there is no publicly available multiview material segmentation dataset. We provide a new dataset of ground truth labels for multiview images to enable quantitative evaluation of various material segmentation methods. The images are derivatives of the SpaceNet Challenge dataset [98]. Figure 4.2 shows part of our labeled regions, we labeled the material classes of five different regions from San Diego, Jacksonville and Omaha, with 10 material classes: Asphalt, Concrete, Glass, Tree, Grass, Metal, Ceramic, Solar Panel, Water and Polymer. The number of annotated pixels for each class is shown in Figure 4.3. For the other labeled regions, please see Section 4.6.

We present three major contributions in this work: 1) the angular luminance histogram as an important cue for material segmentation; 2) AngLNet to integrate the angular luminance histograms with state-of-the-art deep learning architectures for material-based semantic segmentation; 3) A training dataset for multiview imagery with per-pixel material labels.

4.2 Related Work

Models based on Fully Convolutional Neural Network (FCN) [99] have demonstrated superior performance on several segmentation benchmarks [100, 101, 102]. When adapting the CNNs [2, 43, 103] deployed for the task of image classification into segmentation, a coarse-to-fine deconvolution network is learned for upsampling [104, 105, 106, 107].

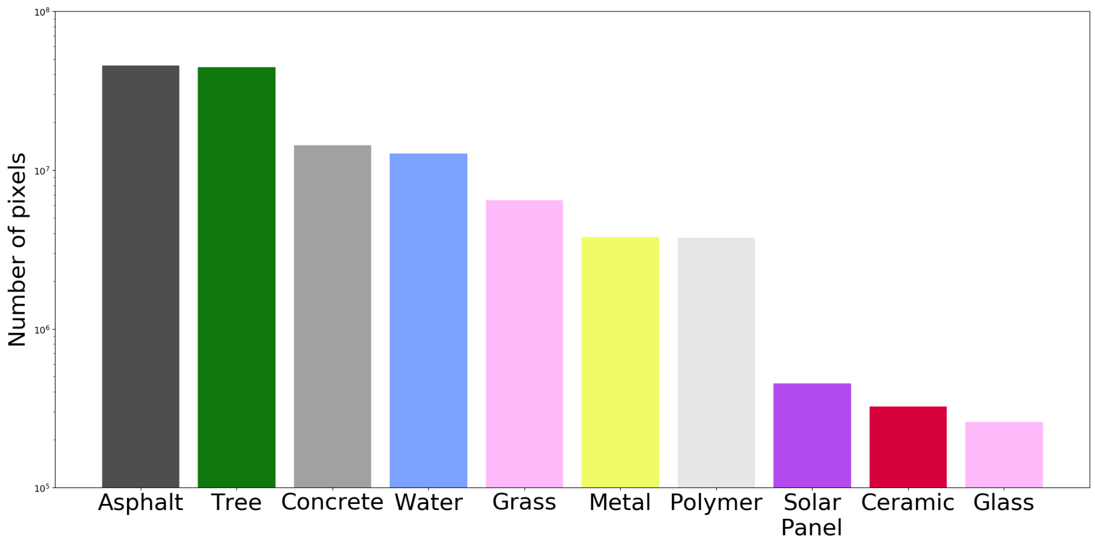


Figure 4.3: Number of annotated pixels (y-aixs) per class and their associated categories (x-aixs) for the labeled satellite material segmentation dataset.

For example, DeepLabv3+ [106] extends DeepLabv3 [107] by adding a decoder module to recover the object boundaries. To enforce prediction consistency during the coarse-to-fine deconvolution, a hierarchical supervision is introduced to help optimize the learning process [108, 109]. Inspired by this work, we use a similar hierarchical supervision module that combines angular histograms with image features at different upsampling scales during the coarse-to-fine deconvolution. Recently, PSPNet [110] and EncNet [111] leverage global context information and boost segmentation performance dramatically. However, for material segmentation, the global context information often provides limited help in cases where the same semantic object can be composed of different materials. Indeed, our comparison results in Section 4.4 indicate that the context information of PSPNet and EncNet do not boost performance over FCN for the task of material segmentation. For multiview segmentation, Ma *et al.*enforce the prediction consistency by warping CNN feature maps from multiple views into a common reference [97]. For this work, the segmentation is constrained to be consistent over all views. Our approach has an important distinction in its use of multiview images. Instead of consistency, we consider the per-pixel vector of intensities as a sampling of the underlying BRDF from unknown viewing and illumination angles. This unstructured sampling of

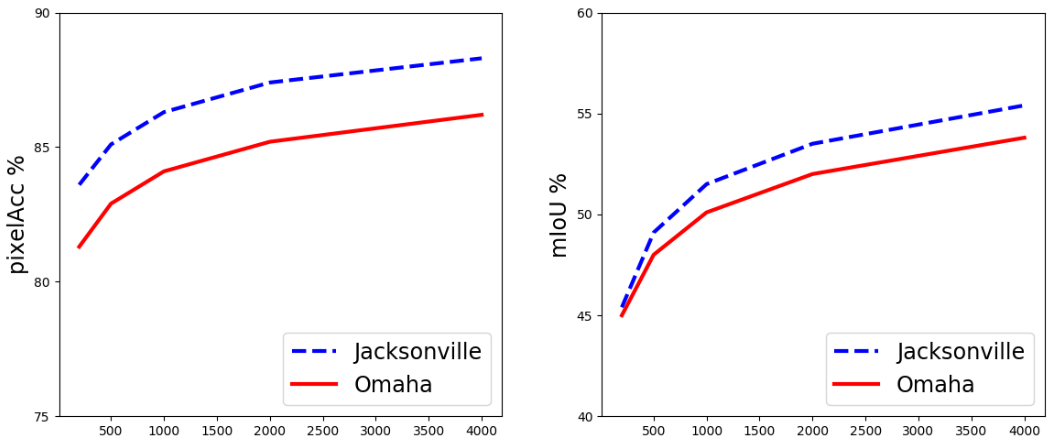


Figure 4.4: Oracle superpixel quantization performance for the test regions (Jacksonville and Omaha). The average achievable segmentation pixel accuracy (pixelAcc, Left) and mean Intersection of Union (mIoU, Right) with the varying number of superpixels for the satellite dataset.

reflectance information is a physically meaningful cue for material id.

Material radiometric properties are captured by the bidirectional reflectance distribution function (BRDF) [37] and the many variants including the bidirectional texture function (BTF) [33], svBRDF[112], BSSRDF[113]. Using intensity distributions or *luminance histograms* to represent reflectance variations has been explored in several previous works [26, 114, 115, 116, 27]. Motoyoshi *et al.*showed that the skewness of the luminance histogram and the skewness of sub-band filter outputs are correlated with surface gloss and inversely correlated with surface albedo (diffuse reflectance) [27]. Dror *et al.*employs histogram statistics to estimate surface reflectance from a single image [26]. Recently, Wang *et al.*propose a learnable histogram layer and learn histogram features within deep neural networks in end-to-end training [117]. Inspired by both classic and recent work, we employ the angular histogram to represent material reflectance in material segmentation networks. While prior work computes image statistics over a fixed spatial region, our approach computes statistics over multiple viewing angles for each pixel (or for each superpixel region).

4.3 Methods

We propose a new framework called Angular Luminance Network (AngLNet), which aims to leverage material reflectance variance information and recover image detailed information from downsampled feature maps. In the network, we combine the constructed angular histogram with pre-trained CNNs in a coarse-to-fine manner based on step by step upsampling. In the following, we will first introduce the angular histogram and the way to integrate angular histogram with CNNs, then we will introduce our network architecture and the relationship with previous works.

4.3.1 Angular Histogram

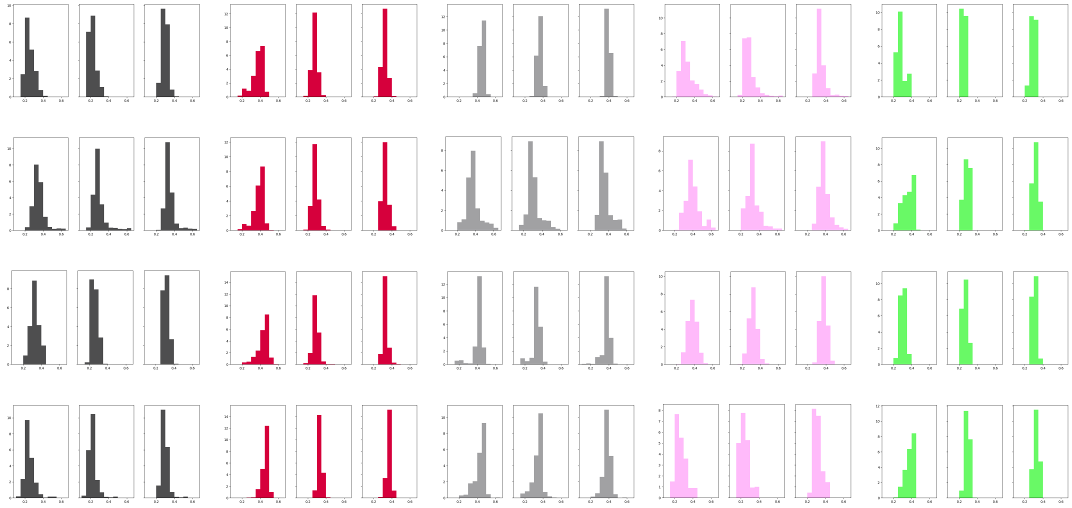
The bidirectional reflectance distribution function (BRDF) specifies how much of the light incident is reflected by an opaque surface patch in each possible view direction. BRDF is a function of four continuous angular variables. Let l be the light direction, v represents the view direction, the BRDF function is expressed as $f(\theta_i, \varphi_i; \theta_r, \varphi_r)$, where θ_i, θ_r and φ_i, φ_r are the polar and azimuth angles of I and v respectively. If $I(\theta_i, \varphi_i)$ represents the illumination incident, the total reflected radiance $I(\theta_r, \varphi_r)$ of the surface patch to the view direction is

$$\int_{\varphi_i=0}^{2\pi} \int_{\theta_i=0}^{\pi/2} f(\theta_i, \varphi_i; \theta_r, \varphi_r) I(\theta_i, \varphi_i) \cos\theta_i \sin\theta_i \, d\theta_i d\varphi_i. \tag{4.1}$$

Replacing θ_i, φ_i and θ_r, φ_r with l and v, the BRDF function becomes $f_r(l, v)$, and the relationship between the reflected radiance $I_o(v)$ and the illumination incident $I_i(l)$ is

$$I_o(v) = I_i(l) f_r(l, v) \tag{4.2}$$

For different material classes $c = 1, 2 \ldots k$, the BRDF function is different as $f_{cr}(l, v)$. With the same input light $I_i(l)$, the observed output light is determined by the material BRDF function $f_{cr}(l, v)$. For uncontrolled lighting condition, if input light follows similar distribution for different materials, since different materials own different BRDF functions, the captured output light has different distributions for different materials. Although we lack enough information to formulate the BRDF functions, we can identify which one most closely represents an observed surface with a finite set of candidate



(a) Asphalt (b) Ceramic (c) Concrete (d) Glass (e) Grass

Figure 4.5: The angular luminance histogram for different classes (asphalt, ceramic, concrete, glass and grass) of the satellite images. The angular histograms are computed over each local superpixel and over 14 viewing angles. Each histogram is constructed based on one superpixel. For the angular luminance histogram of other classes, please see Section 4.6.

reflectance functions. In our dataset, we have 10 material classes, which corresponds to 10 reflectance functions.

Histogram distribution for material BRDF representation has been explored in many previous works [26, 115, 118, 119]. But many focus on material recognition or 3D rendering, where the single image only contains single material. For material segmentation, the single image contains multiple different materials, so a per-pixel material id is needed. For efficient computation, we build the angular histogram per-superpixel instead of per-pixel. Superpixel for image segmentation is a well-studied area in computer vision [120, 121, 122, 123]. Figure 4.4 shows the average achievable segmentation performance with different number of superpixels for Omaha and Jacksonville, which are the test set in our experiment. In experiment, we employ SLIC [124] as the superpixel algorithm and set the number of superpixels be 2000 for each image. As shown in Figure 4.1, for each superpixel segmented material patch, we compute the histogram distribution from a stack of multiview material patches, which are at the same spatial location of the multiview images. Observing that most of the pixel intensities are concentrated on a relatively small region, for the setting of histogram bins, we use a multi-scale bin setting. We set a relatively large bin for the whole range of pixel values, and set another relatively small bin for the narrow concentrated pixel value range. Finally we concatenate the multi-scale histogram intensities as our angular histogram. To integrate the constructed angular histogram with deep network, we replace each pixel in the segmented material patch with the constructed angular histogram. In this way, we get a histogram image that has the same height and width as the input color image. To combine the constructed angular histogram with RGB color image, as shown in Figure 4.6, through a sequence of 1×1 convolutional and non-linear mapping layers, we project the angular histogram into same space as feature maps extracted from CNN, and the concatenated features form the material feature representation.

4.3.2 Angular Luminance Network

We develop a new architecture, the Angular Luminance Network (AngLNet), incorporating the angular luminance histogram and a novel configuration of network elements that encompasses recent developments in deep learning. The angular histograms provide a $h \times w \times b$ feature (where b is the bin size) that is integrated with pre-trained CNNs in a coarse-to-fine manner as illustrated in Figure 4.6. Following prior work [125, 110, 111], we use a pre-trained ResNet [2] model with dilated network strategy [125, 126] to extract feature maps. By downsampling to the same height/width as the extracted feature maps and going through a projection layer, which is composed by a sequence of 1×1 convolutional layers and non-linear mapping layers, the constructed angular histogram features (bin-counts) are projected into the same space as the features maps from ResNet. Based on [127], features from shallow layers are helpful to capture textures, so we use skip connections to combine low level features for material segmentation. Features from shallow layers (labelled SF1 in Figure 4.6), features from last residual block, and projected angular histograms (PAH1) are concatenated as group features (GF). The group features go through a newly designed residual block

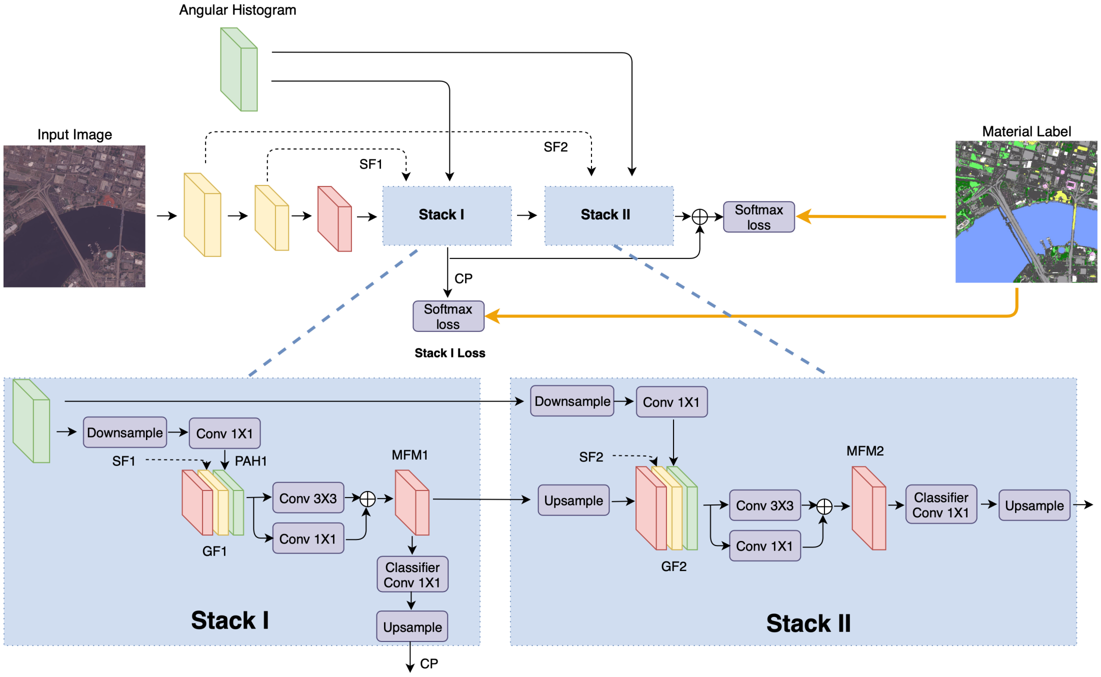


Figure 4.6: Overall architecture of the proposed AngLNet. Given an input image, we first use a pre-trained CNN to extract feature maps. Through a projection layer which is composed by a sequence of 1×1 convolutional layers and non-linear mapping layers, the constructed angular luminance features are concatenated with CNN extracted feature maps to form the material feature representation. The material representation is fed into a residual block and a convolutional layer to make per-pixel prediction. AngLNet makes prediction with a coarse-to-fine stacking network, where stack I learns the coarse prediction, and stack II learns the residual for refinement. (Notation: ⊕ means element-wise addition. CP means coarse prediction, GF means group features, SF1 and SF2 are the shallow layer features, MFM1 and MFM2 are the Material Feature Maps.)

and generate the Material Feature Map (MFM1). Generating this MFM1 is the process labelled Stack I in Figure 4.6. The MFM1 is fed in two directions: in one direction the MFM1 is upsampled by 2 and fed into the Stack II process where it is combined with the projected angular histogram (PAH2) and the shallow layer feature maps (SFM2). The other direction generates a coarse prediction (CP). Based on the observation in [107], that upsampling the prediction for loss computation gives better performance, we keep the ground truth at the original resolution and upsample the prediction to the same size of ground truth for the coarse prediction loss computation. To optimize the learning process and fully utilize features learned from Stack I stage (MFM1), we use a coarse-to-fine residual learning that predicts the fine segmentation by the addition of the coarse prediction (CP) and prediction from Stack II. In this way, Stack II prediction is only responsible for residual refinement learning and the overall network is learned in a coarse-to-fine manner. In network training, the computed loss is a combination of the fine prediction loss and the coarse prediction loss multiplied by a loss weight. In network testing, the accuracy and mIoU are calculated based on the fine prediction only.

Stacking Networks

Stacking networks has been applied in many different areas [128, 129, 108, 130]. FlowNet2 improves the optical flow prediction results by stacking two FlowNetS for flow refinement [128]. StackGAN generates small size images first and stack the same network for refinement to generate high-resolution images with photo-realistic details [129]. The advantage of stacking networks is to mitigate the single network burden for complex tasks. In this manner, the first network in the stack can generate a coarse prediction, and the second network is responsible for refinement. Inspired by this, we use a two stages coarse-to-fine network for material segmentation, as shown in Figure 4.6, we combine the angular histogram with features extracted from pre-trained CNN in two network stages. Prediction in Stack I comes from small size height/width feature maps; it is responsible for coarse prediction. Prediction in Stack II comes from upsampled feature maps, and it is responsible for refinement. To optimize the learning process and fully leverage the residual learning strategy, the final prediction is the addition of prediction from Stack I and prediction from Stack II. Our comparison results in Section 4.4 indicate the importance of coarse-to-fine residual learning.

4.3.3 Dataset Processing

The SpaceNet Challenge dataset contains both WordView2 and WorldView3 multispectral and panchromatic satellite images from several regions taken over multiple years. The dataset has been used for challenges involving off-nadir building detection, road network extraction, and building footprint extraction. The regions of interest in the dataset are medium sized cities and suburbs from the United States. In this work, only multispectral WorldView3 images are used. The multispectral images contain eight wavelengths ranging from coastal blue to near infrared red. Images with snow or too much cloud cover are manually removed from the dataset. The dataset is non-uniformly sampled in regard to both the times and the angles the images were taken.

The original images in the SpaceNet Challenge are unwieldy due to their large size. Thus, all images belonging to the same region are first cropped at specified latitude and longitude coordinates. A sparse ground truth material mask is manually created by labeling high confidence regions with material labels. The ground truth material masks are labeled in a space directly nadir to the ground. In order to correctly assign material labels to off-angle images, a mapping between image space and nadir orientation is required. Images are orthorectified given the image and an elevation model provided by P3D, a module of the Danesfield [131]. Images are further aligned using the Lucas-Kanade pixel-wise alignment method.

WorldView3 images are originally relatively radiometrically calibrated to remove streaks and banding artifacts. The values of each pixel are a function of how much spectral radiance enters the telescope, which is unique to the WorldView3 satellite images. Each channel of the image is converted to top-of-atmospheric spectral radiance separately by:

$$L = GAIN \cdot DN \cdot \frac{abscalfactor}{effectivebandwidth} + OFFSET \tag{4.3}$$

where the DN corresponds to the raw pixel value, the $GAIN$ and $OFFSET$ are absolute radiometric calibration values, and the $abscalfactor$ is the radiometric calibration factor. The images are further normalized for solar irradiance and sensor radiance by conversation to top-of-atmospheric reflectance by:

$$R_\lambda = \frac{L_\lambda \cdot d^2 \cdot \pi}{E_\lambda \cdot \cos\theta_S} \tag{4.4}$$

Where L_λ is the sensor radiance, found in Equation 4.3, d is the Earth-Sun distance, E_λ is the solar irradiance, and θ_S is the solar zenith angle. With the images in reflectance units, pixel values can be directly compared to reflectance values measured in material BRDF libraries.

For ground truth labeling, labeling every rooftop in a tile can require thousands of manually generated outlines as well as expert knowledge in material identification from satellite images. This process is difficult to scale and is infeasible for full image material annotation. Instead we develop a semi-automated process that reduces the tedious aspects of manually labeling to generate fully annotated material masks for each of the tiles in reasonable time frames. A pixel-wise multi-angle convolutional neural network (CNN) is trained on all manually labeled ground truth data. The trained network evaluates each pixel in the new tile to generate a dense material mask. Generating annotations of dynamic scenes in a shared space inherently leads to label ambiguity. Specific challenges of labeling materials in satellite images from the SpaceNet dataset are seasonal changes, moving objects (e.g. cars), buildings construction, and general outdoor wear and tear of rooftops (e.g. rust or dirt). The resultant dense ground truth material masks are noisy but generally accurate. We employ several noise reducing techniques to improve the ground truth masks used to train our algorithms. Individual image annotation masks are aggregated to produce smoother dense annotations. Third party building outlines from U.S. Cities or OpenStreetMap for the tile are gathered

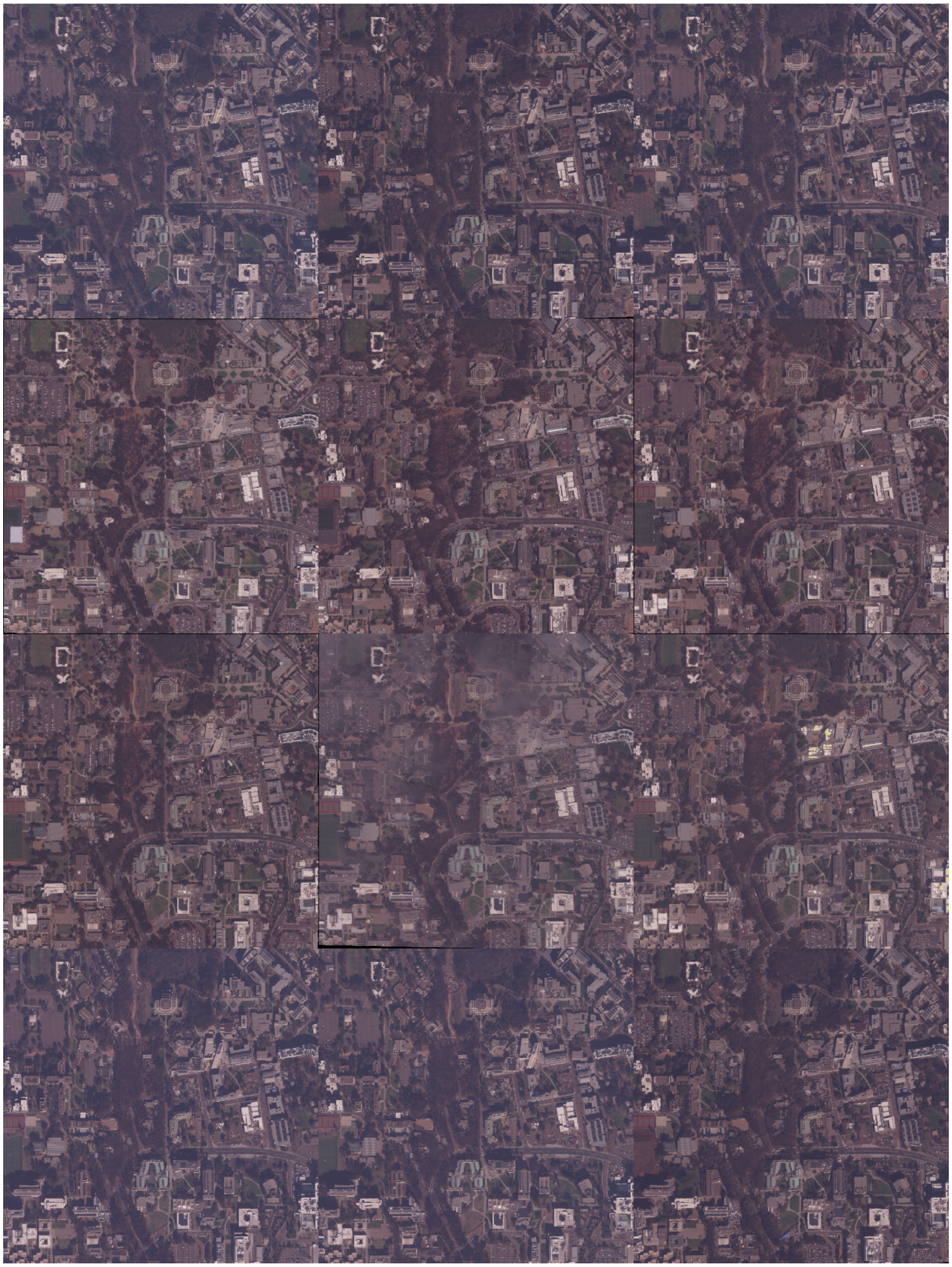


Figure 4.7: Part of the multiview images for the San Diego region.

according to the coordinates of the tile and projected into image space. For each building outline, an initial material classification is given based on the prediction from the dense mask. An annotator cycles through the building outlines updating any erroneous material classifications and/or adjusting any building outline errors. The time required to label new tiles is significantly reduced through this method. The densely labeled material masks can then be used to train semantic segmentation algorithms.

With aforementioned dataset processing, we provide a new dataset of per-pixel ground truth labels for images from the SpaceNet Challenge. As shown in Figure 4.2, we labeled the material classes of five different regions from San Diego, Jacksonville and Omaha. The height/width of the labeled regions varies from 3000×3000 to 8000×8000, the number of observations in each region varies from 12 to 19. Part of the multiview images for the San Diego region is shown in Figure 4.7, for the Multiview images example of Jacksonville, please see Section 4.6. For the material segmentation task, the labeled satellite images are cropped into 500×500 patches resulting in 7421 patches with 10 material classes: Asphalt, Concrete, Glass, Tree, Grass, Metal, Ceramic, Solar Panel, Water and Polymer. The number of annotated pixels for each class is shown in Figure 4.3.

4.4 Experiments

In this section, we evaluate the AngLNet framework for material segmentation and compare the results with several state-of-the-art single view or multiview segmentation algorithms. We introduce the satellite dataset and provide implementation details for AngLNet and the baseline approach. We conduct a complete ablation study of the network structure. Additionally, we demonstrate transferring AngLNet to aerial imagery using the Stanford drone dataset [132] to show the generalization of AngLNet.

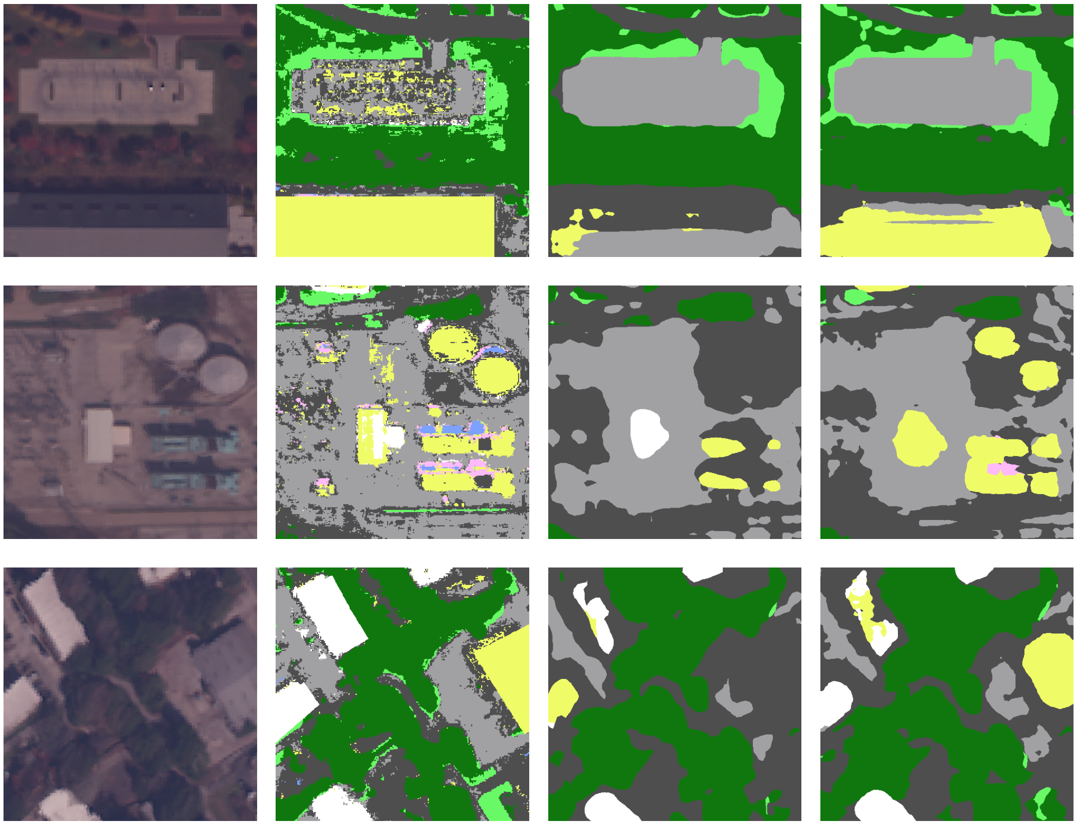
4.4.1 Training procedure

In experiments, we use cross validation that set one region as the test set and the rest of the regions as the training set. This dataset of ground truth labels is made

Method	BaseNet	Angular Histogram	Stacking Network	Voting	Jacksonville		Omaha	
					pixACC	mIoU	pixACC	mIoU
FCN	ResNet18				72.6	26.6	66.9	30.8
AngLNet	ResNet18	✓			73.9	28.5	68.3	31.7
AngLNet	ResNet18	✓	✓		74.7	28.9	68.9	31.9
AngLNet	ResNet50	✓	✓		75.2	29.9	68.5	30.6
AngLNet	ResNet18	✓	✓	✓	77.1	33.5	74.4	36.0

Table 4.1: Results comparing performance with different components of AngLNet for material segmentation. AngLNet + Angular histogram means concatenating angular luminance histogram with features from FCN baseline for segmentation, i.e. without the Stack II process in Figure 4.6. Stacking Network means prediction from both Stack I and Stack II, the final prediction is the addition of coarse prediction from Stack I with refinement prediction from Stack II. Voting means multiview majority voting.

publicly available. For training, we separately choose one region from Jacksonville and Omaha as test set, and set the other four regions as training set for experiments. We incorporate pre-trained ResNet [2] based on ImageNet [56]. Following prior works [110, 111, 126, 125], we apply dilation strategy to stage 3 and 4 of the pre-trained networks with the output height/width 1/8 of input image, we use the poly learning rate scheduling that $lr = baselr*(1-\frac{iter}{total_iter})^{power}$ for experiment, the base learning rate is set to 0.01 and the power is set to 0.9. Momentum and weight decay are set to 0.9 and 0.0001 respectively. The networks are trained for 50 epochs. For data augmentation, we adopt randomly horizontal and vertical mirror flips with 50% chance respectively. We randomly resize all training images between 0.5 and 2, we randomly rotate the image between -10° to 10° and finally crop the images into 480×480 for training. We use the mini-batch size of 16 with synchronized Batch Normalization [110, 111, 107] during the training. For ResNet implementation, we use ResNet18. For comparison with our network, we use dilated ResNet FCN as the baseline approaches. Pixel accuracy (pixAcc) and mean Intersection of Union (mIoU) are used as as evaluation metrics.



(a) Image (b) Ground Truth (c) FCN (d) AngNet

Figure 4.8: Qualitative material segmentation results of AngLNet and dilated FCN baseline on the satellite dataset. AngLNet improves the performance on both material prediction correctness and material prediction completeness. In the first three columns, FCN predicts some material classes incorrectly, and in the last three columns, FCN prediction is incomplete.

Number of	Jacksonville		Omaha	
Networks	pixACC	mIoU	pixACC	mIoU
1	73.9	28.5	68.3	31.7
2	**74.7**	28.9	**68.9**	**31.9**
3	74.4	**29.1**	67.5	30.9

Table 4.2: Results comparing performance with different number of stacked networks N. $N = 1$ means without the Stack II process in Figure 4.6, i.e. prediction from Stack I only. $N = 2$ stacked network is the AngLNet as depicted in Figure 4.6. $N = 3$ stacked network upsamples the feature maps and combine angular histograms for three times to compute the final prediction.

4.4.2 AngLNet segmentation performance

We evaluate AngLNet material segmentation performance of two regions from our labeled satellite dataset. Additionally, for an ablation study, we conduct experiments with different network structure settings as shown in Table 4.1. Compared to the dilated FCN baseline, using the angular histogram improves the performance in Jacksonville from 72.6/26.6 to 73.9/28.5 in terms of pixel accuracy and mean IoU (%). With stacking networks, the result further exceeds by 0.8/0.4 and reaches 74.7/28.9. We also experiment with a deeper network that use ResNet50 as base net, but it does not consistently improve the segmentation performance.

Majority voting is a common way to improve for multiview recognition, by observing the same object with different viewing points and making prediction based on the most probable assumption. We adopt a similar method for material segmentation: for each pixel, we assign the most likely material class based on material id prediction frequency. With majority voting, the final segmentation performance is 77.1/33.5 for Jacksonville and 74.4/36.0 for Omaha. Table 4.5 shows the importance of majority voting for multiview material segmentation. Majority voting boosts performance for both dilated FCN and AngLNet. Dense-CRF is a ubiquitous adopted post processing

Loss Weight α	Jacksonville		Omaha	
	pixACC	mIoU	pixACC	mIoU
$\alpha = 0.0$	74.3	28.7	67.7	31.1
$\alpha = 0.1$	**74.8**	**28.9**	68.1	31.6
$\alpha = 0.2$	74.7	**28.9**	**68.9**	**31.9**
$\alpha = 0.5$	74.4	28.4	68.2	**31.9**
$\alpha = 0.8$	74.4	28.3	68.3	31.8

Table 4.3: The performance with different loss weight α for Stack I loss. $\alpha = 0.0$ denotes without stack I loss. Empirically, $\alpha = 0.2$ yields the best performance.

in segmentation [125, 117], but as shown in Table 4.5, we find that it is not effective for our task. To study the effect of Stack I loss, we test different weight of Stack I loss $\alpha = \{0.0, 0.1, 0.2, 0.5, 0.8\}$ for two test sets, as shown in Table 4.3, we find $\alpha = 0.2$ yields the best performance. To study the effect of number of stacked networks, we test the performance with different number of stacked networks N, as shown in Table 4.2. $N = 1$ means without the Stack II process in Figure 4.6, i.e. prediction from Stack I only. $N = 2$ stacked network is the AngLNet as depicted in Figure 4.6. $N = 3$ stacked network upsamples the feature maps and combine angular histograms for three times to compute the final prediction. Similar to [128], we do not find consistent improvement by stacking more networks. In Table 4.4, we compare the network performance with and without coarse-to-fine residual learning. The residual learning improves by 0.7/1.0 on Jacksonville and 1.4/2.8 on Omaha. The results indicate the benefit of computing the final prediction as the addition of coarse prediction from Stack I with refinement prediction from Stack II.

The qualitative segmentation comparison of AngLNet and FCN is shown in Figure 4.8. AngLNet improves the performance on both material prediction correctness and material prediction completeness. For example, in the first column, FCN predicts the metal building as half asphalt and half concrete, but AngLNet classifies it correctly.

Coarse-to-fine	Jacksonville		Omaha	
Residual	pixACC	mIoU	pixACC	mIoU
N	74.0	27.9	67.5	29.1
Y	**74.7**	**28.9**	**68.9**	**31.9**

Table 4.4: A comparison of the performance with and without coarse-to-fine residual learning.

In the 4^{th} column, both FCN and AngLNet predict the metal building and concrete road correctly, but FCN predicts part of them as asphalt, the prediction of AngLNet is complete.

Table 4.6 shows the segmentation result of AngLNet outperforms three other single view (PSPNet [110], EncNet [111]) or multiview (MVCNet [97]) segmentation algorithms. For equal comparison, no majority voting is used, all reported performances are based on pre-trained ResNet18, and we use the same image augmentation for all the algorithms. We set auxiliary weight 0.2 for PSPNet, we set the weight of SE-loss 0.2, and the number of codewords 32 for EncNet. For MVCNet, since we don't have depth images in our dataset, in our experiment, we do not use depth view branch fusion. The results show that the performance of PSPNet and EncNet is similar to the dilated FCN baseline, so we conclude that the global context information provides limited help for our task. For example, the material of buildings and parking lots is independent of the global context information.

4.4.3 AngLNet on Stanford Drone Dataset

We experiment on the Stanford drone dataset [132] to show the generalization of AngLNet trained with the labelled satellite database. Figure 4.9 shows example images from the Stanford Drone Dataset and the Spacenet Challenge dataset. Since the Stanford dataset does not have material segmentation ground truth, we train the AngLNet from the satellite dataset, and test on the Stanford drone dataset. For the Stanford drone dataset, we extract a sequence of frames from each video and resize them into

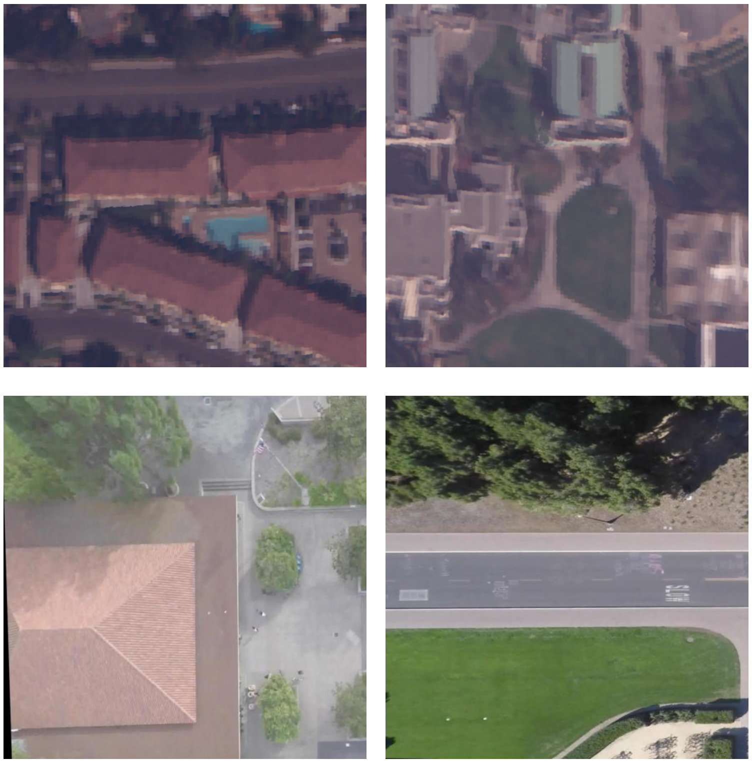


Figure 4.9: Comparing the difference between satellite (top) and drone (down) images. Notice that the building roofs in 1^{st} column are all ceramic.

Method	Jacksonville		Omaha	
	pixACC	mIoU	pixACC	mIoU
FCN	72.6	26.6	66.9	30.8
FCN + CRF	68.8	20.4	42.5	15.5
FCN + Voting	75.3	31.2	72.9	34.4
AngLNet	74.7	28.9	68.9	31.9
AngLNet + CRF	69.3	20.9	44.1	14.4
AngLNet + Voting	**77.1**	**33.5**	**74.4**	**36.0**

Table 4.5: A comparison of the performance with and without CRF or majority voting. CRF does not work in our task, majority voting boosts the performance of both dilated FCN baseline and AngLNet.

520 × 520 for experiment. To reduce the difference between the two datasets, we normalize the image mean and standard derivation of the images from the Stanford dataset to be the same as the Spacenet dataset using

$$I_o = \frac{I_i - m_d}{std_d} std_s + m_s, \tag{4.5}$$

where I_i is the resized original image the Stanford dataset, m_d, std_d and m_s, std_s are the mean and standard derivation of Stanford dataset and the Spacenet dataset respectively. To further overcome the gap between two datasets, we use histogram matching to transform the satellite images into the same histogram distribution as the stanford drone dataset. Considering the effect of atmospheric condition for prediction, we retrain our network on the San Diego region only for this experiment. The results are shown in Figure 4.10. Considering the inherent difference between satellite images and drone images, our network provide good generalization for ground terrain material segmentation.

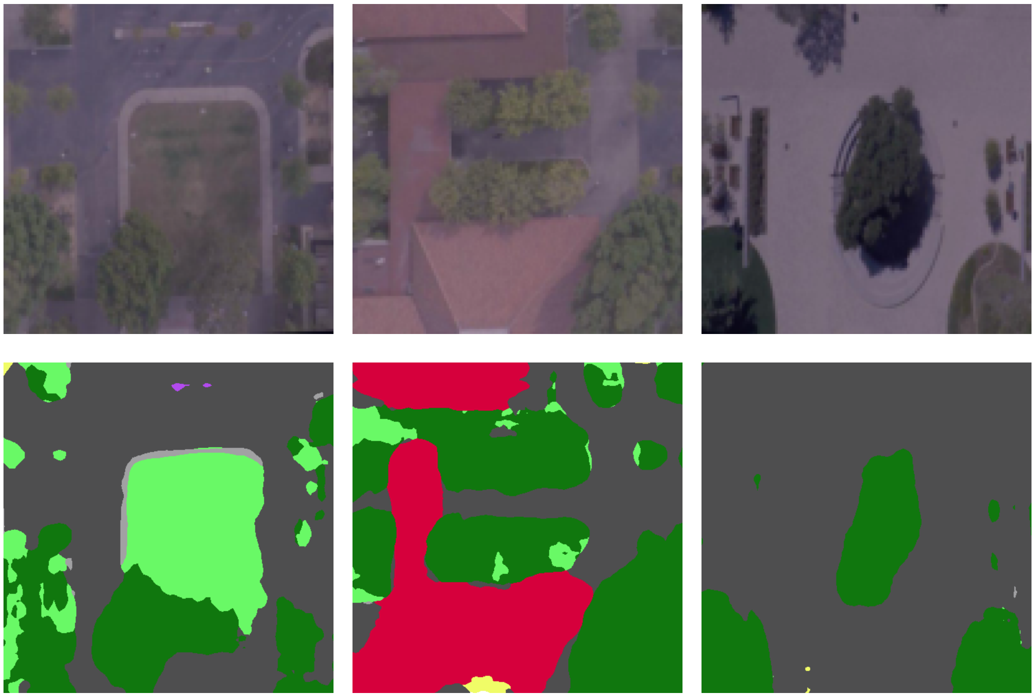


Figure 4.10: Experiment results on Stanford drone dataset. To reduce the difference between two datasets, we transfer the database mean and standard derivation of stanford drone dataset to same as the satellite dataset, and use histogram matching to match the satellite images into the same histogram distribution for experiment.

Method	Jacksonville		Omaha	
	pixACC	mIoU	pixACC	mIoU
PSPNet [110]	73.5	28.0	67.1	29.9
EncNet [111]	73.7	28.1	67.2	29.2
MVCNet [97]	74.2	27.8	66.8	29.6
AngLNet (ours)	**74.7**	**28.9**	**68.9**	**31.9**

Table 4.6: A comparison with state-of-the-art single view or multi-view segmentation algorithms. For equal comparison, we do not use majority voting for all the algorithms. Notice that our method, AngLNet, achieves the best material segmentation.

4.5 Summary and Conclusion

Luminance histograms provide the statistics of natural images and can be used as a strong material recognition cue. We revisit these classic concepts for multiview image sequences that are commonly captured in modern applications including, but not limited to, drone and satellite imagery. The variation of local intensity with viewing angle is used to compute an angular luminance histogram. We show that utilizing this feature boost performance in modern deep learning architectures for material-based semantic segmentation. Our contribution are the angular luminance histogram integrated with a novel architecture and a ground truth multiview material training dataset.

4.6 Appendix

Figure 4.11 is the overview of the rest satellite images and our labeled ground truth. Figure 4.12 is part of the multiview images for the Jacksonville region. Figure 4.13 is the angular luminance histogram for different classes (metal, polymer, solar panel, tree and water) of the satellite images. The angular histograms are computed over each local superpixel and over 14 viewing angles. Figure 4.14 is some other qualitative material segmentation results of AngLNet and dilated FCN baseline on the satellite dataset

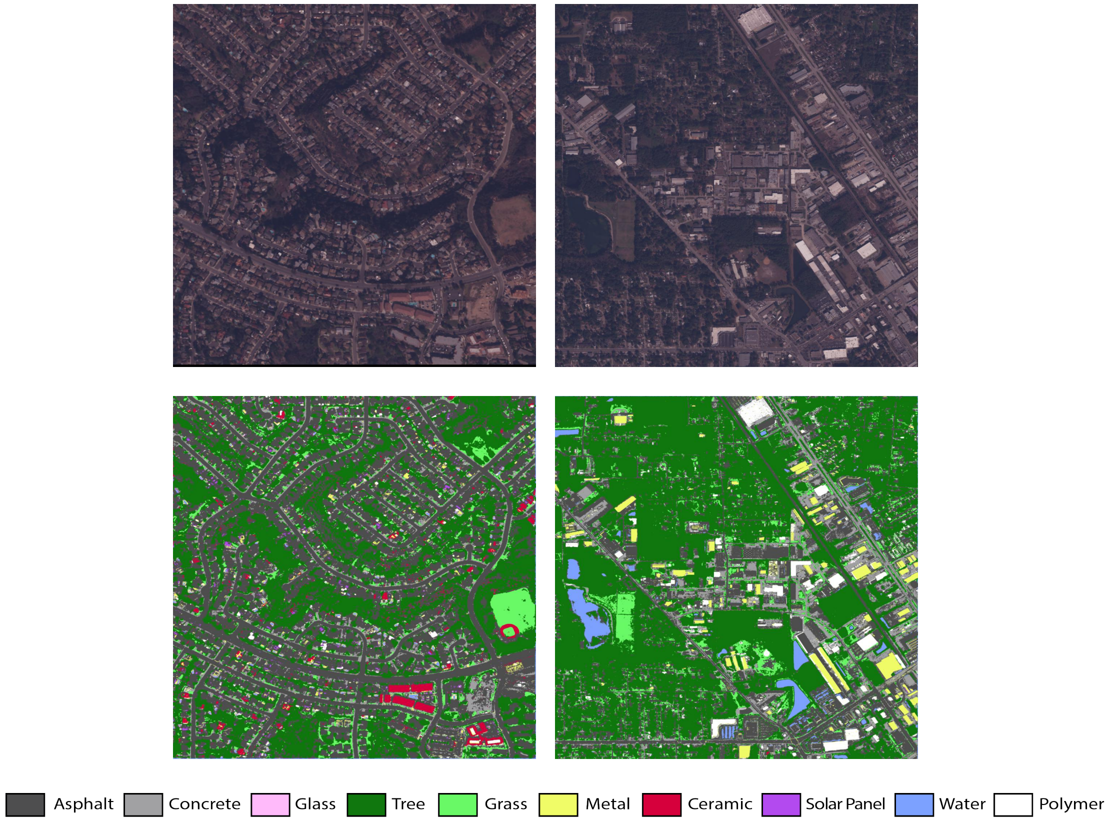


Figure 4.11: Overview of the satellite images and our labeled ground truth. We labeled the material classes of five different regions from San Diego, Jacksonville and Omaha, with 10 material classes: Asphalt, Concrete, Glass, Tree, Grass, Metal, Ceramic, Solar Panel, Water and Polymer. The height/width of labeled regions varies from 3000×3000 to 8000×8000.

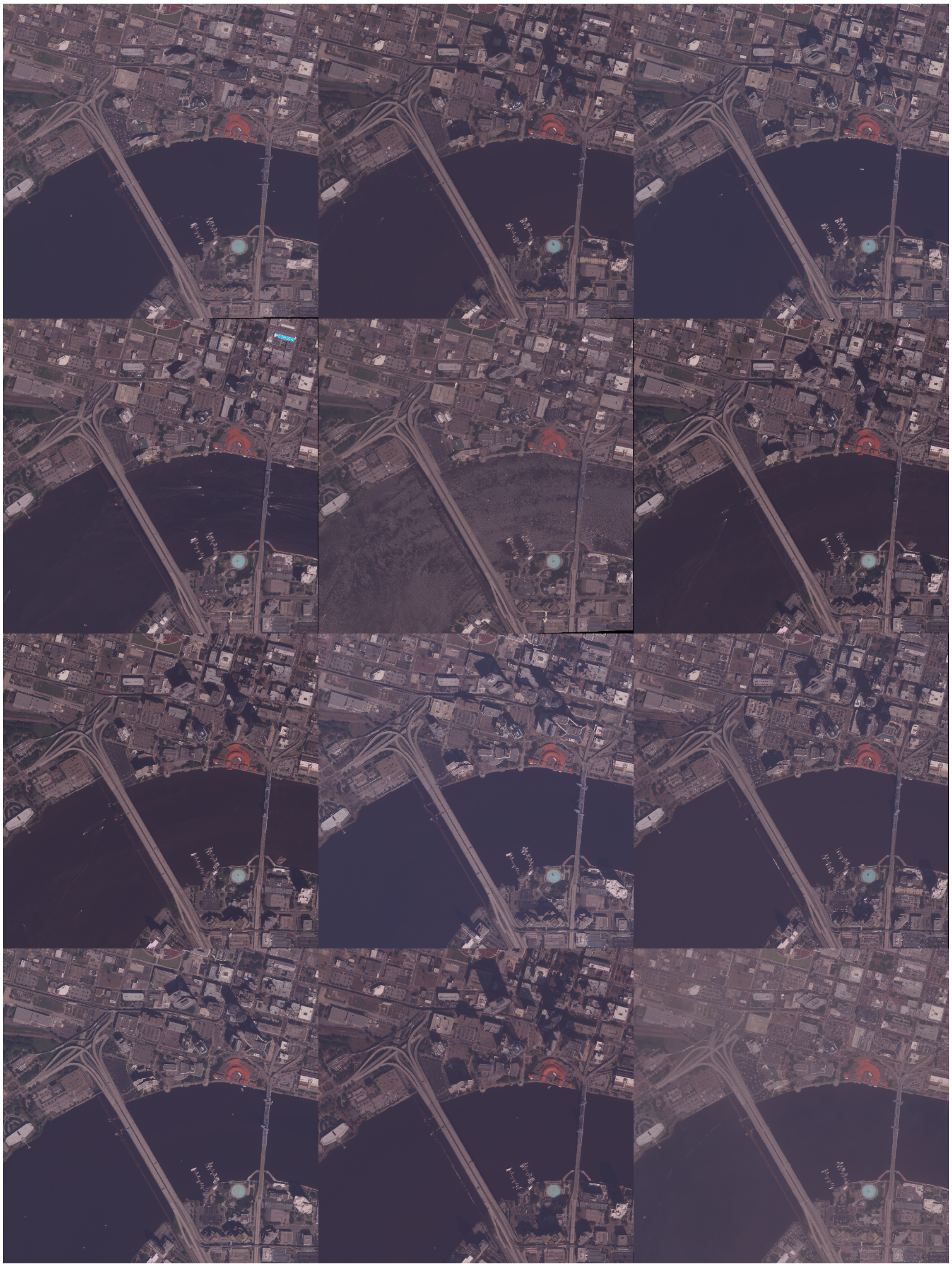
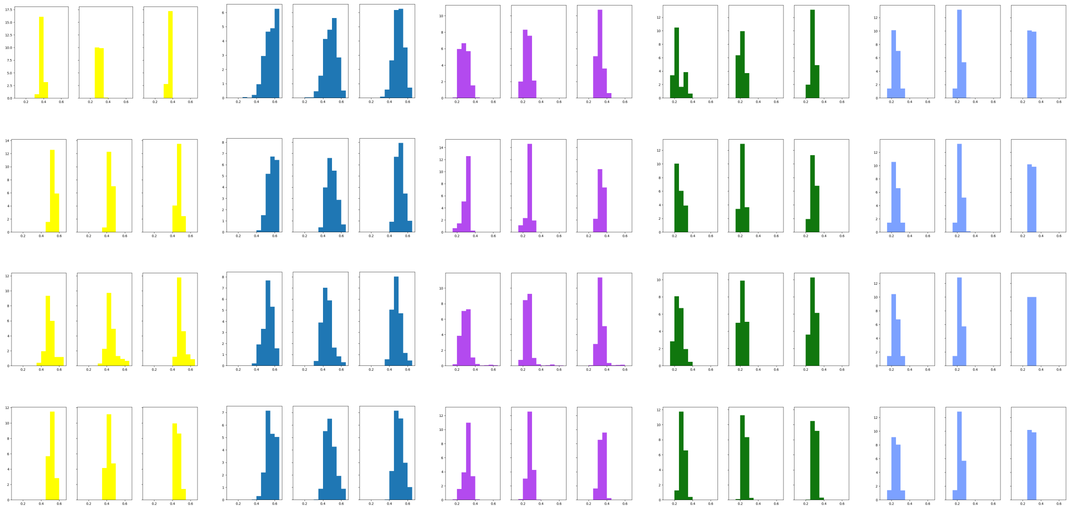
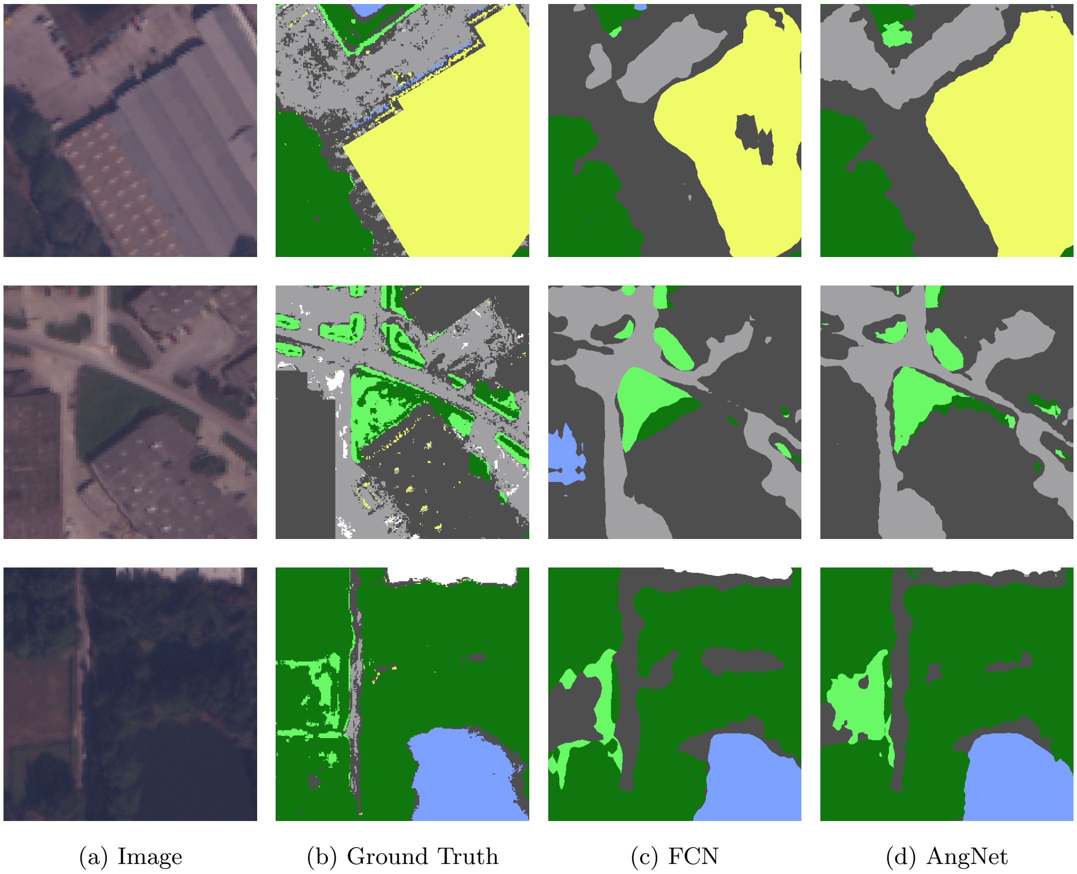


Figure 4.12: The multiview images for the Jacksonville region.



(a) Metal (b) Polymer (c) SolarPanel (d) Tree (e) Water

Figure 4.13: The angular luminance histogram for different classes (metal, polymer, solar panel, tree and water) of the satellite images. The angular histograms are computed over each local superpixel and over 14 viewing angles. Each histogram is constructed based on one superpixel.



(a) Image (b) Ground Truth (c) FCN (d) AngNet

Figure 4.14: Qualitative material segmentation results of AngLNet and dilated FCN baseline on the satellite dataset. AngLNet improves the performance on both material prediction correctness and material prediction completeness. In the first three columns, FCN predicts some material classes incorrectly, and in the last three columns, FCN prediction is incomplete.

Chapter 5

Conclusion and Future Work

5.1 Conclusion

This thesis is dedicated to develop representations that capture the intrinsic invariant properties of material surfaces, which enables fine-grained material recognition and segmentation. I have focused on: 1) a middle-ground approach for material recognition that takes advantage of both rich radiometric cues and flexible image capture. 2) deep texture networks to capture material reflectance information, orderless texture details and ordered spatial information for robust ground terrain material recognition and segmentation. 3) a new texture manifold method, DEP-manifold, to find the relationship between newly captured images and images in dataset.

I have demonstrated the effectiveness and the efficiency of the techniques theoretically and practically. Specifically, I have developed the following methods:

Differential Angular Imaging. I have introduced the Differential Angular Imaging for a sparse representation of the spatial distribution of angular gradients that provides key cues for material recognition. I collect the GTOS Dataset with ground terrain imaged by systematic in-scene measurement of partial reflectance instead of in-lab reflectance measurements. The database contains 34,243 images with 40 surface classes, 18 viewing directions, 4 illumination conditions, 3 exposure settings per sample and several instances/samples per class. I develop and evaluate an architecture for using differential angular imaging, showing superior results for differential inputs as compared to original images. My work in measuring and modeling outdoor surfaces has important implications for applications such as robot navigation (determining control parameters based on current ground terrain) and automatic driving (determining road conditions by partial real time reflectance measurements).

Material Recognition Networks I develop and evaluate novel approaches for material recognition, including Differential Angular Imaging Network (DAIN), Deep Encoding Pooling Network (DEP) and the Texture Encoded Angular Network (TEAN). The material recognition networks employ differential angular imaging, texture details and spatial information for material recognition, showing superior results for differential inputs as compared to original images. With the novel network architectures, I extract characteristics of materials encoded in the angular and spatial gradients of their appearance. The results show that the introduced methods achieve recognition performance that surpasses existing single view performance and standard (non-differential/large-angle sampling) multiview performance. These methods will provides a foundation for additional in-depth studies of material recognition in the wild.

Deep Texture Manifold. I have developed methods for recognition of ground terrain for potential applications in robotics and automated vehicles. I have introduced the Deep Encoding Pooling network (DEP) that leverages an orderless representation and local spatial information for recognition. I also Introduce the DEP-manifold that integrates DEP network on top of a deep neural network to predict the manifold coordinates of a texture directly. I collection the GTOS-mobile database comprised of 81 ground terrains videos of similar terrain classes as GTOS, captured with a handheld mobile phone to evaluate knowledge-transfer between different image capture methods but within the the same domain.

Angular Luminance Networks. Luminance histograms provide the statistics of natural images and can be used as a strong material recognition cue. I revisit these classic concepts for multiview image sequences that are commonly captured in modern applications including, but not limited to, drone and satellite imagery. The variation of local intensity with viewing angle is used to compute an angular luminance histogram. I show that utilizing this feature boost performance in modern deep learning architectures for material-based semantic segmentation. Our contribution are the angular luminance histogram integrated with a novel architecture and a ground truth multiview material training dataset.

Till now, my material recognition and segmentation works have been broadly adapted into other works. For example, Fan *et al.* [133] integrated the image-level features and encoding layer into a single module called the Global Encoding Module (GEModule) for the task of semantic segmentation. Features extracted from convolutional layers are fed into the texture encoding layer and the global average pooling layer jointly, outputs are concatenated for semantic prediction. Song *et al.* [134] combine features from texture detail layer and global average pooling layer for ground terrain recognition, they achieve superior performance on the GTOS dataset. My satellite material segmentation work has been adapted into the Danesfield 3D modeling system [131, 135]. I hope my approaches inspire others finding methods for recognition and segmentation tasks. Most techniques proposed in this dissertation are general and ready to be applied to other vision tasks.

5.2 Future Directions

There are several important topics in the material and texture modeling field that need further investigation. I will discuss interesting and promising topics that I will pursue in the future work in the following:

Differential Angular Imaging. The motivation for this differential change in viewpoint is improving computation of the angular gradient of intensity $\partial I_v/\partial v$. Differential angular imaging provides key information about material reflectance properties while maintaining the flexibility of convenient in-scene appearance capture. Although the differential angular images have several advantage characteristics: 1) the differential image reveals the gradients in BRDF/BTF at the particular viewpoint. 2) relief texture is also observable in the differential image due to non-planar surface structure. 3) the differential images are sparse. This sparsity can provide a computational advantage within the network. Traditional convolution neural network can not utilize the computational efficiency of sparse images. With convolutional operation, the network still needs to go through differential angular image pixel by pixel. I can design a deep learning network architecture for sparse images, which employ the sparsity property of differential angular images, boost the computational burden and improve the recognition performance. Another potential direction is that the domain difference between the differential angular images and object recognition pre-trained models. With model fine-tuning to transfer the knowledge learned from pre-trained models to new domains, I can employ advanced fine-tuning methods such as transfer module [136] to improve model fine-tuning accuracy.

Angular Luminance for Segmentation. Luminance histograms provide the statistics of natural images and can be used as a strong material recognition cue. For material segmentation, the single image contains multiple different materials, so a per-pixel material id is needed. For efficient computation, I build the angular histogram per-superpixel instead of per-pixel. Superpixel for image segmentation is a well-studied area in computer vision, but most superpixel algorithms are based on image clustering and work as offline pre-processing. To construct luminance histograms in real time and improve the inference efficiency of AngLNet, I can implement a deep learning network for image superpixel prediction. So that I can integrate the superpixel prediction into material segmentation and realize real time inference.

References

[1] T.-C. Wang, J.-Y. Zhu, E. Hiroaki, M. Chandraker, A. A. Efros, and R. Ramamoorthi, “A 4d light-field dataset and cnn architectures for material recognition,” in *European Conference on Computer Vision.* Springer, 2016, pp. 121–138.

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

[3] T.-Y. Lin, A. RoyChowdhury, and S. Maji, “Bilinear cnn models for fine-grained visual recognition,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1449–1457.

[4] H. Zhang, J. Xue, and K. Dana, “Deep ten: Texture encoding network,” *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[5] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, “Mobilenetv2: Inverted residuals and linear bottlenecks,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4510–4520.

[6] L. Van Der Maaten, “Accelerating t-sne using tree-based algorithms.” *Journal of machine learning research*, vol. 15, no. 1, pp. 3221–3245, 2014.

[7] K. J. Dana, “Computational texture and patterns: From textons to deep learning,” *Synthesis Lectures on Computer Vision*, vol. 8, no. 3, pp. 1–113, 2018.

[8] ——, “Capturing computational appearance: More than meets the eye,” *IEEE Signal Processing Magazine*, vol. 33, no. 5, pp. 70–80, 2016.

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

[10] C. Liu and J. Gu, “Discriminative Illumination: Per-Pixel Classification of Raw Materials Based on Optimal Projections of Spectral BRDF,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 1, pp. 86–98, January 2014.

[11] S. Bell, P. Upchurch, N. Snavely, and K. Bala, “Material recognition in the wild with the materials in context database,” *Computer Vision and Pattern Recognition (CVPR)*, 2015.

[12] D. Hu, L. Bo, and X. Ren, “Toward Robust Material Recognition for Everyday Objects,” in *BMVC*, 2011, pp. 48.1–48.11.

[13] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, and A. Vedaldi, "Describing textures in the wild," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 3606–3613.

[14] C. Liu, L. Sharan, E. H. Adelson, and R. Rosenholtz, "Exploring Features in a Bayesian Framework for Material Recognition," in *CVPR*, 2010, pp. 239–246.

[15] J. Malik, S. Belongie, T. Leung, and J. Shi, "Contour and texture analysis for image segmentation," *International Journal of Computer Vision*, vol. 43, no. 1, pp. 7–27, Jun 2001. [Online]. Available: https://doi.org/10.1023/A:1011174803800

[16] O. G. Cula and K. J. Dana, "Compact representation of bidirectional texture functions," *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 1041–1067, December 2001.

[17] T. Leung and J. Malik, "Representing and recognizing the visual appearance of materials using three-dimensional textons," *International journal of computer vision*, vol. 43, no. 1, pp. 29–44, 2001.

[18] M. Varma and A. Zisserman, "A statistical approach to texture classification from single images," *International Journal of Computer Vision*, vol. 62, no. 1, pp. 61–81, Apr 2005. [Online]. Available: https://doi.org/10.1023/B:VISI.0000046589.39864.ee

[19] G. Csurka, C. R. Dance, L. Fan, J. Willamowski, and C. Bray, "Visual categorization with bags of keypoints," in *In Workshop on Statistical Learning in Computer Vision, ECCV*, 2004, pp. 1–22.

[20] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, vol. 2, 2006, pp. 2169–2178.

[21] H. Zhang, K. Dana, and K. Nishino, "Reflectance hashing for material recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3071–3080.

[22] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *Advances in Neural Information Processing Systems*, 2014, pp. 568–576.

[23] M. Cimpoi, S. Maji, and A. Vedaldi, "Deep filter banks for texture recognition and segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3828–3836.

[24] 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.

[25] E. Hayman, B. Caputo, M. Fritz, and J.-O. Eklundh, "On the significance of real-world conditions for material classification," in *European conference on computer vision*. Springer, 2004, pp. 253–266.

[26] R. O. Dror, E. H. Adelson, and A. S. Willsky, “Estimating surface reflectance properties from images under unknown illumination,” in *Human Vision and Electronic Imaging VI*, vol. 4299. International Society for Optics and Photonics, 2001, pp. 231–243.

[27] I. Motoyoshi, S. Nishida, L. Sharan, and E. H. Adelson, “Image statistics and the perception of surface qualities,” *Nature*, vol. 447, no. 7141, p. 206, 2007.

[28] J. Xue, H. Zhang, K. J. Dana, and K. Nishino, “Differential angular imaging for material recognition.” in *CVPR*, 2017, pp. 6940–6949.

[29] N. Salamati, C. Fredembach, and S. Süsstrunk, “Material Classification using Color and NIR Images,” in *IS&T/SID Color Imaging Conference*, 2009.

[30] G. Schwartz and K. Nishino, “Visual Material Traits: Recognizing Per-Pixel Material Context,” in *IEEE Color and Photometry in Computer Vision Workshop*, 2013.

[31] ——, “Automatically discovering local visual material attributes,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[32] H. Zhang, K. Dana, and K. Nishino, “Reflectance hashing for material recognition,” *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 371–380, 2015.

[33] 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.

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

[35] G. Choe, S. G. Narasimhan, and I. S. Kweon, “Simultaneous estimation of near ir brdf and fine-scale surface geometry,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[36] O. G. Cula and K. J. Dana, “Recognition methods for 3d textured surfaces,” in *Proceedings of SPIE conference on human vision and electronic imaging VI*, no. 209-220, 2001, p. 3.

[37] F. Nicodemus, J. Richmond, J. Hsia, I. Ginsberg, and T. Limperis, “Geometric Considerations and Nomenclature for Reflectance,” National Bureau of Standards (US), 1977.

[38] G. Ward, “Measuring and modeling anisotropic reflection,” in *ACM SIGGRAPH 92*, 1992, pp. 265–272.

[39] M. Levoy and P. Hanrahan, “Light Field Rendering,” in *Computer Graphics Proceedings, ACM SIGGRAPH 96*, Aug. 1996, pp. 31–42.

[40] P. Debevec, T. Hawkins, C. Tchou, H.-P. Duiker, W. Sarokin, and M. Sagar, "Acquiring the reflectance field of a human face," in *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH '00. New York, NY, USA: ACM Press/Addison-Wesley Publishing Co., 2000, pp. 145–156. [Online]. Available: http://dx.doi.org/10.1145/344779.344855

[41] K. Dana and J. Wang, "Device for convenient measurement of spatially varying bidirectional reflectance," *Journal of the Optical Society of America A*, vol. 21, pp. pp. 1–12, January 2004.

[42] H. Zhang, K. Nishino, and K. Dana, "Friction from Reflectance: Deep Reflectance Codes for Predicting Physical Surface Properties from One-Shot In-Field Reflectance," in *European Conference on Computer Vision*, 2016, pp. 808–824.

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

[44] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman, "Return of the devil in the details: Delving deep into convolutional nets," in *British Machine Vision Conference*, 2014.

[45] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *arXiv preprint arXiv:1512.03385*, 2015.

[46] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587.

[47] A. Karpathy and L. Fei-Fei, "Deep visual-semantic alignments for generating image descriptions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3128–3137.

[48] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.

[49] C. Feichtenhofer, A. Pinz, and A. Zisserman, "Convolutional two-stream network fusion for video action recognition," *arXiv preprint arXiv:1604.06573*, 2016.

[50] K. Soomro, A. R. Zamir, and M. Shah, "Ucf101: A dataset of 101 human actions classes from videos in the wild," *arXiv preprint arXiv:1212.0402*, 2012.

[51] W. Matusik, H. Pfister, M. Brand, and L. McMillan, "A data-driven reflectance model," *ACM Transactions on Graphics*, vol. 22, no. 3, pp. 759–769, Jul. 2003.

[52] J. Filip and R. Vávra, "Template-based sampling of anisotropic brdfs," *Comput. Graph. Forum*, vol. 33, no. 7, pp. 91–99, Oct. 2014. [Online]. Available: http://dx.doi.org/10.1111/cgf.12477

[53] G. Oxholm, P. Bariya, and K. Nishino, "The Scale of Geometric Texture," in *European Conference on Computer Vision*, vol. I, 2012, pp. 58–71.

[54] C. Kampouris, S. Zafeiriou, A. Ghosh, and S. Malassiotis, “Fine-grained material classification using micro-geometry and reflectance,” in *European Conference on Computer Vision.* Springer, 2016, pp. 778–792.

[55] J. DeGol, M. Golparvar-Fard, and D. Hoiem, “Geometry-informed material recognition,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1554–1562.

[56] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “Imagenet: A large-scale hierarchical image database,” in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on.* IEEE, 2009, pp. 248–255.

[57] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, “Learning spatiotemporal features with 3d convolutional networks,” in *2015 IEEE International Conference on Computer Vision (ICCV).* IEEE, 2015, pp. 4489–4497.

[58] H. Su, S. Maji, E. Kalogerakis, and E. Learned-Miller, “Multi-view convolutional neural networks for 3d shape recognition,” in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 945–953.

[59] A. Vedaldi and A. Zisserman, “Efficient additive kernels via explicit feature maps,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 34, no. 3, pp. 480–492, 2012.

[60] J. Xue, H. Zhang, and K. Dana, “Deep texture manifold for ground terrain recognition,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 558–567.

[61] M. Lin, Q. Chen, and S. Yan, “Network in network,” *International Conference on Learning Representations*, 2014.

[62] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. H. S. Torr, “Conditional random fields as recurrent neural networks,” in *The IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[63] R. Mottaghi, X. Chen, X. Liu, N.-G. Cho, S.-W. Lee, S. Fidler, R. Urtasun, and A. Yuille, “The role of context for object detection and semantic segmentation in the wild,” in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.

[64] G. Lin, C. Shen, A. van den Hengel, and I. Reid, “Exploring context with deep structured models for semantic segmentation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PP, no. 99, pp. 1–1, 2017.

[65] G. Schwartz and K. Nishino, “Material recognition from local appearance in global context,” *arXiv preprint arXiv:1611.09394*, 2016.

[66] S. Bell, P. Upchurch, N. Snavely, and K. Bala, “Material recognition in the wild with the materials in context database,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3479–3487.

[67] E. Shelhamer, J. Long, and T. Darrell, "Fully convolutional networks for semantic segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 4, pp. 640–651, April 2017.

[68] J. B. Tenenbaum and W. T. Freeman, "Separating style and content," in *Advances in neural information processing systems*, 1997, pp. 662–668.

[69] J. Xue, H. Zhang, K. Dana, and K. Nishino, "Differential angular imaging for material recognition," *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[70] L. v. d. Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of Machine Learning Research*, vol. 9, no. Nov, pp. 2579–2605, 2008.

[71] L. van der Maaten, "Learning a parametric embedding by preserving local structure," *RBM*, vol. 500, no. 500, p. 26, 2009.

[72] M. R. Min, L. Maaten, Z. Yuan, A. J. Bonner, and Z. Zhang, "Deep supervised t-distributed embedding," in *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, 2010, pp. 791–798.

[73] M. R. Min, H. Guo, and D. Song, "Exemplar-centered supervised shallow parametric data embedding," *arXiv preprint arXiv:1702.06602*, 2017.

[74] B. Caputo, E. Hayman, and P. Mallikarjuna, "Class-specific material categorisation," in *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, vol. 2. IEEE, 2005, pp. 1597–1604.

[75] Z. Meng, P. Liu, J. Cai, S. Han, and Y. Tong, "Identity-aware convolutional neural network for facial expression recognition," in *The Twelfth IEEE International Conference on Automatic Face and Gesture Recognition*, 2017.

[76] J. Cai, Z. Meng, A. S. Khan, Z. Li, and Y. Tong, "Island loss for learning discriminative features in facial expression recognition," *arXiv preprint arXiv:1710.03144*, 2017.

[77] X. Peng, R. S. Feris, X. Wang, and D. N. Metaxas, "A recurrent encoder-decoder network for sequential face alignment," in *European Conference on Computer Vision*. Springer, 2016, pp. 38–56.

[78] Y. Zhu and S. Newsam, "Depth2action: Exploring embedded depth for large-scale action recognition," in *European Conference on Computer Vision*. Springer, 2016, pp. 668–684.

[79] Y. Zhu, Z. Lan, S. Newsam, and A. G. Hauptmann, "Hidden two-stream convolutional networks for action recognition," *arXiv preprint arXiv:1704.00389*, 2017.

[80] Z. Zhang, Y. Xie, F. Xing, M. McGough, and L. Yang, "Mdnet: A semantically and visually interpretable medical image diagnosis network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 6428–6436.

[81] J. C. Van Gemert, J.-M. Geusebroek, C. J. Veenman, and A. W. Smeulders, "Kernel codebooks for scene categorization," in *European conference on computer vision.* Springer, 2008, pp. 696–709.

[82] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, "Netvlad: Cnn architecture for weakly supervised place recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 5297–5307.

[83] W. T. Freeman and J. B. Tenenbaum, "Learning bilinear models for two-factor problems in vision," in *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on.* IEEE, 1997, pp. 554–560.

[84] H. Pirsiavash, D. Ramanan, and C. C. Fowlkes, "Bilinear classifiers for visual recognition," in *Advances in neural information processing systems*, 2009, pp. 1482–1490.

[85] F. Perronnin, J. Sánchez, and T. Mensink, "Improving the fisher kernel for large-scale image classification," *Computer Vision–ECCV 2010*, pp. 143–156, 2010.

[86] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[87] G. Huang, Z. Liu, K. Q. Weinberger, and L. van der Maaten, "Densely connected convolutional networks," *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[88] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[89] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, 2010, pp. 249–256.

[90] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman, "Return of the devil in the details: Delving deep into convolutional nets," *British Machine Vision Conference*, 2014.

[91] T. Weyrich, W. Matusik, H. Pfister, B. Bickel, C. Donner, C. Tu, J. McAndless, J. Lee, A. Ngan, H. W. Jensen *et al.*, "Analysis of human faces using a measurement-based skin reflectance model," in *ACM Transactions on Graphics (TOG)*, vol. 25, no. 3. ACM, 2006, pp. 1013–1024.

[92] S. R. Marschner, S. H. Westin, E. P. Lafortune, K. E. Torrance, and D. P. Greenberg, "Image-based brdf measurement including human skin," in *Rendering Techniques 99.* Springer, 1999, pp. 131–144.

[93] C. Liu and J. Gu, "Discriminative illumination: Per-pixel classification of raw materials based on optimal projections of spectral brdf," *IEEE transactions on pattern analysis and machine intelligence*, vol. 36, no. 1, pp. 86–98, 2014.

[94] 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.

[95] C. Kerl, J. Sturm, and D. Cremers, “Dense visual slam for rgb-d cameras,” in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on.* Citeseer, 2013, pp. 2100–2106.

[96] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, “Indoor segmentation and support inference from rgbd images,” in *European Conference on Computer Vision.* Springer, 2012, pp. 746–760.

[97] L. Ma, J. Stückler, C. Kerl, and D. Cremers, “Multi-view deep learning for consistent semantic mapping with rgb-d cameras,” in *Intelligent Robots and Systems (IROS), 2017 IEEE/RSJ International Conference on.* IEEE, 2017, pp. 598–605.

[98] M. Brown, H. Goldberg, K. Foster, A. Leichtman, S. Wang, S. Hagstrom, M. Bosch, and S. Almes, “Large-scale public lidar and satellite image data set for urban semantic labeling,” in *Laser Radar Technology and Applications XXIII*, vol. 10636. International Society for Optics and Photonics, 2018, p. 106360P.

[99] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.

[100] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba, “Scene parsing through ade20k dataset,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, no. 2. IEEE, 2017, p. 4.

[101] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, “The pascal visual object classes (voc) challenge,” *International journal of computer vision*, vol. 88, no. 2, pp. 303–338, 2010.

[102] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, “The cityscapes dataset for semantic urban scene understanding,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 3213–3223.

[103] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv preprint arXiv:1409.1556*, 2014.

[104] H. Noh, S. Hong, and B. Han, “Learning deconvolution network for semantic segmentation,” in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1520–1528.

[105] V. Badrinarayanan, A. Kendall, and R. Cipolla, “Segnet: A deep convolutional encoder-decoder architecture for image segmentation,” *arXiv preprint arXiv:1511.00561*, 2015.

[106] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, “Encoder-decoder with atrous separable convolution for semantic image segmentation,” *arXiv preprint arXiv:1802.02611*, 2018.

[107] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," *arXiv preprint arXiv:1706.05587*, 2017.

[108] J. Fu, J. Liu, Y. Wang, and H. Lu, "Stacked deconvolutional network for semantic segmentation," *arXiv preprint arXiv:1708.04943*, 2017.

[109] H. Zhao, X. Qi, X. Shen, J. Shi, and J. Jia, "Icnet for real-time semantic segmentation on high-resolution images," *arXiv preprint arXiv:1704.08545*, 2017.

[110] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 2881–2890.

[111] H. Zhang, K. Dana, J. Shi, Z. Zhang, X. Wang, A. Tyagi, and A. Agrawal, "Context encoding for semantic segmentation," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[112] H. P. Lensch, J. Kautz, M. Goesele, W. Heidrich, and H.-P. Seidel, "Image-based reconstruction of spatially varying materials," in *Rendering Techniques 2001*. Springer, 2001, pp. 103–114.

[113] H. W. Jensen, S. R. Marschner, M. Levoy, and P. Hanrahan, "A practical model for subsurface light transport," in *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*. ACM, 2001, pp. 511–518.

[114] E. H. Adelson, "On seeing stuff: the perception of materials by humans and machines," in *Human vision and electronic imaging VI*, vol. 4299. International Society for Optics and Photonics, 2001, pp. 1–13.

[115] J. R. Shell, C. Salvaggio, and J. R. Schott, "A novel brdf measurement technique with spatial resolution-dependent spectral variance," in *Geoscience and Remote Sensing Symposium, 2004. IGARSS'04. Proceedings. 2004 IEEE International*, vol. 7. IEEE, 2004, pp. 4754–4757.

[116] E. P. Simoncelli and B. A. Olshausen, "Natural image statistics and neural representation," *Annual Review of Neuroscience*, vol. 24, no. 1, pp. 1193–1216, 2001, pMID: 11520932.

[117] Z. Wang, H. Li, W. Ouyang, and X. Wang, "Learnable histogram: Statistical context features for deep neural networks," in *European Conference on Computer Vision*. Springer, 2016, pp. 246–262.

[118] H. Zhang, Z. Jiao, Y. Dong, and X. Li, "Evaluation of brdf archetypes for representing surface reflectance anisotropy using modis brdf data," *Remote Sensing*, vol. 7, no. 6, pp. 7826–7845, 2015.

[119] G. A. Atkinson and E. R. Hancock, "Two-dimensional brdf estimation from polarisation," *Computer Vision and Image Understanding*, vol. 111, no. 2, pp. 126–141, 2008.

[120] R. Gadde, V. Jampani, M. Kiefel, D. Kappler, and P. V. Gehler, "Superpixel convolutional networks using bilateral inceptions," in *European Conference on Computer Vision*. Springer, 2016, pp. 597–613.

[121] V. Jampani, D. Sun, M.-Y. Liu, M.-H. Yang, and J. Kautz, "Superpixel sampling networks," *arXiv preprint arXiv:1807.10174*, 2018.

[122] S. Gould, J. Zhao, X. He, and Y. Zhang, "Superpixel graph label transfer with learned distance metric," in *European Conference on Computer Vision.* Springer, 2014, pp. 632–647.

[123] M. Mostajabi, P. Yadollahpour, and G. Shakhnarovich, "Feedforward semantic segmentation with zoom-out features," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3376–3385.

[124] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, S. Süsstrunk *et al.*, "Slic superpixels compared to state-of-the-art superpixel methods," *IEEE transactions on pattern analysis and machine intelligence*, vol. 34, no. 11, pp. 2274–2282, 2012.

[125] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 4, pp. 834–848, 2018.

[126] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," *arXiv preprint arXiv:1511.07122*, 2015.

[127] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *European conference on computer vision.* Springer, 2014, pp. 818–833.

[128] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox, "Flownet 2.0: Evolution of optical flow estimation with deep networks," in *IEEE conference on computer vision and pattern recognition (CVPR)*, vol. 2, 2017, p. 6.

[129] H. Zhang, T. Xu, H. Li, S. Zhang, X. Huang, X. Wang, and D. Metaxas, "Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks," *arXiv preprint*, 2017.

[130] A. Newell, K. Yang, and J. Deng, "Stacked hourglass networks for human pose estimation," in *European Conference on Computer Vision.* Springer, 2016, pp. 483–499.

[131] M. J. Leotta, C. Long, B. Jacquet, M. Zins, D. Lipsa, J. Shan, B. Xu, Z. Li, X. Zhang, S.-F. Chang *et al.*, "Urban semantic 3d reconstruction from multiview satellite imagery," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2019, pp. 0–0.

[132] A. Robicquet, A. Sadeghian, A. Alahi, and S. Savarese, "Learning social etiquette: Human trajectory understanding in crowded scenes," in *European conference on computer vision.* Springer, 2016, pp. 549–565.

[133] L. Fan, H. Kong, W.-C. Wang, and J. Yan, "Semantic segmentation with global encoding and dilated decoder in street scenes," *IEEE Access*, vol. 6, pp. 50 333–50 343, 2018.

[134] P. Song, X. Ma, X. Li, and Y. Li, "Deep residual texture network for terrain recognition," *IEEE Access*, vol. 7, pp. 90 152–90 161, 2019.

[135] M. Purri, J. Xue, K. Dana, M. Leotta, D. Lipsa, Z. Li, B. Xu, and J. Shan, “Material segmentation of multi-view satellite imagery,” *arXiv preprint arXiv:1904.08537*, 2019.

[136] Y. Zhu, J. Xue, and S. Newsam, “Gated transfer network for transfer learning,” in *Asian Conference on Computer Vision.* Springer, 2018, pp. 354–369.