Understanding state-of-the-art material classification through deep visualization

By

Jordan T. Donovan

A Thesis
Submitted to the Faculty of
Mississippi State University
in Partial Fulfillment of the Requirements
for the Degree of Master of Science
in Computer Science
in the Department of Computer Science and Engineering

Mississippi State, Mississippi

December 2019

Understanding state-of-the-art material classification through deep visualization

By

Jordan T. Donovan

Approved:

Shahram Rahimi
(Major Professor)

Stefano Iannucci
(Committee Member)

Christopher Archibald
(Committee Member / Thesis Director)

T. J. Jankun-Kelly
(Graduate Coordinator)

Jason M. Keith
Dean
Bagley College of Engineering

Name: Jordan T. Donovan

Date of Degree: December 13, 2019

Institution: Mississippi State University

Major Field: Computer Science

Major Professor: Shahram Rahimi

Director of Thesis: Christopher Archibald

Title of Study: Understanding state-of-the-art material classification through deep visualization

Pages of Study: 85

Candidate for Degree of Master of Science

Neural networks (NNs) excel at solving several complex, non-linear problems in the area of supervised learning. A prominent application of these networks is image classification. Numerous improvements over the last few decades have improved the capability of these image classifiers. However, neural networks are still a black-box for solving image classification and other sophisticated tasks. A number of experiments conducted look into exactly how neural networks solve these complex problems. This paper dismantles the neural network solution, incorporating convolution layers, of a specific material classifier. Several techniques are utilized to investigate the solution to this problem. These techniques look at specifically which pixels contribute to the decision made by the NN as well as a look at each neuron's contribution to the decision. The purpose of this investigation is to understand the decision-making process of the NN and to use this knowledge to suggest improvements to the material classification algorithm.

ACKNOWLEDGEMENTS

I would like to thank the members of the Robotic Integrated Engineer Operations research team at the Construction Engineering Research Laboratory for supporting this endeavor both financially as well as having patience with the delivery of this research. The findings and opinions presented in this article do not necessarily represent these persons and are solely those of the author.

I would also like to thank my committee for their comments on this thesis, and I thank my thesis director, Christopher Archibald for directing this research as well as providing information regarding processes and feedback that aided my research immeasurably.

TABLE OF CONTENTS

CHAPTER

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Neural networks have been shown to excel at solving several complex, non-linear tasks. With this being the case, the interworking of these networks are still very much a mystery. Visualization tools have been applied to aid in the discovery of how these networks operate. Material classification is a recently growing area of research in which neural networks can be effectively applied. Material classification is crucial in its own right to several experimental tasks. Both of these topics are further introduced in the following sub-sections.

1.1 Neural Networks and Visualization

Over the past several years, the training and performance of powerful deep neural networks has produced results surpassing even human abilities on numerous machine learning tasks [42], [33], [16]. One of these areas involves training deep convolutional neural networks (CNNs) with supervised learning techniques to classify objects in natural images. This area has seen several improvements due to faster computing, for example, using GPUs [26], [44]; better training techniques, such as dropout [18]; better activation units, for example, rectified linear units (relu) [14]; and larger labeled datasets, like ImageNet [23].

Although the above improvements have been discovered over the last several years, the understanding of how these deep neural networks operate has not progressed at the same rate. Neural networks have been seen as ”black boxes”, as it is not well-known how any particular, trained neural network functions. This, in large part, is due to the large number of interconnected, non-linear units. As these deep neural networks (DNNs) grow in their architectural size, they become even more difficult to study. Understanding exactly what is learned by these neural networks is imperative, as it provides valuable insight into improving their performance. Without this, the process of improving these models is reduced to trial-and-error. One recent approach to visualizing the learned information of a DNN is a technique called Deconvolution [48] which visualizes the features learned by the hidden units of a specific neural network to evaluate which input stimuli excite individual feature maps at each layer in the model. This thesis will utilize recently developed deep visualization tools [47], which employ this Deconvolution approach. The first tool provides forward activation values and backward diffs computed by back-propagation, or Deconvolution, starting from arbitrary units. This provides meaningful information about the activations produced at each layer of the trained DNN. The second tool allows for more interpretable visualization of learned features computed by individual units at each layer of a DNN through preferred stimuli, top images for each unit from the training set, and a utilization of the Deconvolution technique to highlight the top images. This leads to observations of the evolution of features during training that can ultimately help diagnose potential deficiencies within the model.

1.2 Material Classification

Material recognition holds a critical role in the understanding of, and interactions with, the world. One can use it to determine the capability to traverse specific terrain, or if a specific object is movable. Automated material recognition is useful in a variety of applications, including robotics [10] and image editing [21]. Along these same lines, recognizing materials in real-world images is a very challenging task. Materials vary in appearance due to several factors including lighting, shape, and viewing angle. Large-scale databases, such as ImageNet [32], in combination with CNNs have provided breakthroughs recently in object recognition, as mentioned above. Material recognition is now starting to see similar advancements through large-scale data and learning. Several moderate-sized datasets like the Flickr Material Database (FMD) [37] have been compiled and utilized for benchmark material classification tasks. Recently, the Materials in Context Database (MINC) was also compiled [3] which includes 3 million material samples which supplies a more diverse and complete dataset for material classification. This research will utilize the MINC database and a technique from recent state-of-the-art research [3] as it has been shown to outperform other material classification techniques with its large database of materials and its use of CNNs for patch material classification as well as full-scene material classification and segmentation.

1.3 Contribution

This work utilizes and augments the deep visualization tools described briefly above [47] to evaluate state-of-the-art material classification [3]. This shows feature activations projected back to pixel space which is ideal since the material classification technique conducts a per-pixel classification and segmentation. This allows for a complete analysis of how the feature activations map to the input space resulting in information on how the trained neural network is making its decisions. It also allows for visualizing features at each layer of the classifier via regularized optimization in image space. Once this analysis has been completed, this work speculates on possible methods of improving the accuracy of the material classification technique. In addition to performing this analysis and speculation of the material classifier, this effort provides an assessment of the various techniques of the deep visualization tools utilized to perform said analysis.

CHAPTER 2

LITERATURE REVIEW

This chapter will discuss research that has been conducted previously pertaining to two separate areas of interest for this work: visualization of deep neural networks and material classification techniques. It will discuss the evolution of discovery and innovation for these areas as well as providing a valid basis for the experiments to come.

2.1 Neural Network Visualization

Visualizing features to gain intuition about the network's functionality is common practice when discussing the first layer of the network. When it comes to higher layers, there are only a few methods that offer this ability, but of these techniques, deconvolutional networks and the tools from recent work by Yosinski et al. [47] offer advantages over other techniques. These visualization tools produce projections from feature activations at each layer back to pixel space as well as displaying preferred inputs for these layers and the neural networks within.

An increasingly popular method is to consider each layer in a DNN to understand which computations are performed there. One of the approaches of achieving this information is to study each layer as a group and investigate the type of computation performed by the set of neurons at that layer as a whole [28] [46]. In a study by Mahedran et al. [28], an inverted representation is used at each layer to visualize the information retained by that layer. A seperate investigation experiments with transferring features between neural networks to observe which layers store general versus specific information regarding the task and domain [46]. This is completed using transfer learning of progressively more layers of a specific neural network trained on a source dataset and task and then applied to a target task and dataset. These techniques are helpful due to the interaction between layers. Each layer passes information up through to the higher layers, so each individual unit's contribution to the neural network's overall function depends on that neuron's context in the layer.

Another approach is to interpret the computation performed at each individual neuron. There seem to be two separate methods of doing this. One of these methods involves visualizing the feature activations on the images used for training and/or testing the neural network. The deconvolution technique [48] is an example of one of these methods. It highlights the portions of a particular image that contributes to the activation of particular neural units. Additionally, one of the tools produced in research by Yosinksi et al. [47] utilizes this deconvolutional technique along with a combination of other visualization mechanisms to produce a visualization of every neuron in a trained CNN. Figure 2.1 shows an example of one of the types of visualizations that is produced by this tool. The other method for this approach is slightly different and involves investigating the neural network directly without the use of any data. During the study of this method, images were assembled that caused high activations for specific neurons [9]. An initial input is examined for activation caused at a particular unit, and a systematic process is followed

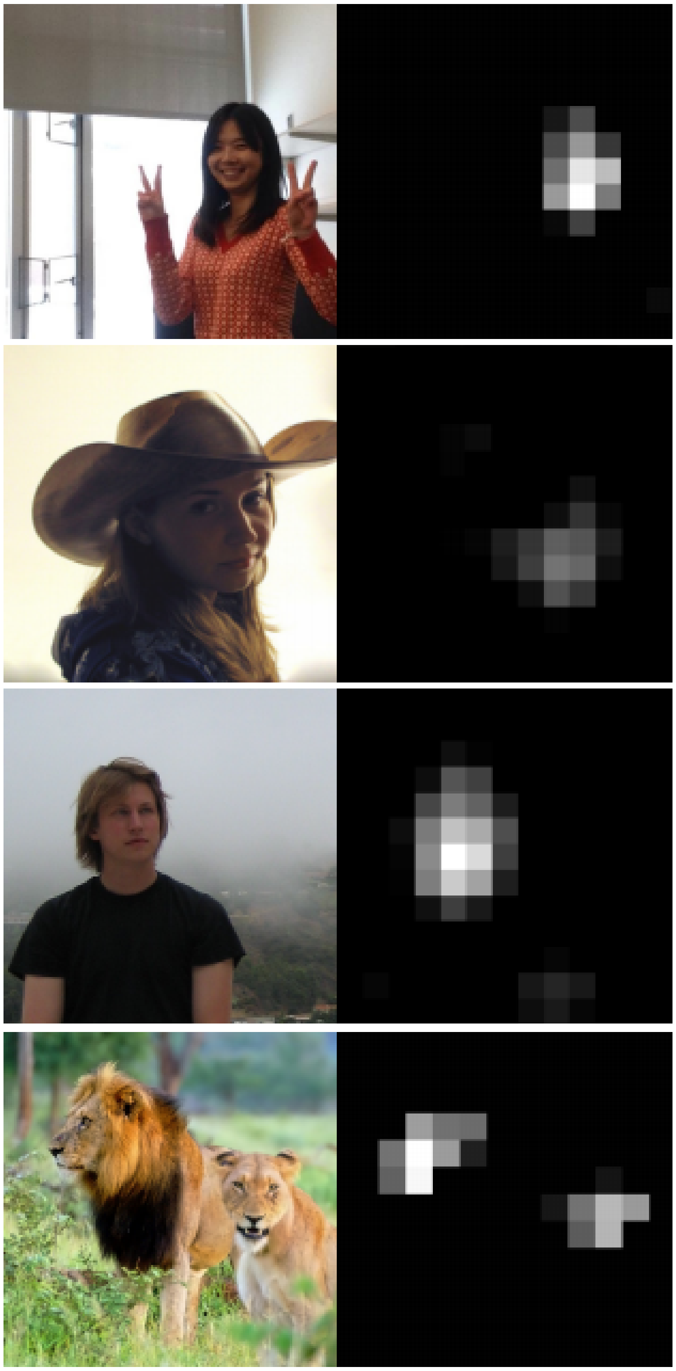


Figure 2.1

Visualizations of a specific layer of a deep neural network trained on the ImageNet dataset. Specifically, this is an example of the first tool from [47] that plots the activation values for the neurons in a specific layer of the CNN. It shows that the classifier recognizes faces.

to create inputs that cause progressively higher activations of this unit until a threshold of activation is reached. This results in an input image that can be displayed for human interpretation. Along these same lines, another experiment generates an image that maximizes the class score for a specific class which results in a visualization of the class as it is represented by the CNN [38]. Nguyen et al. also conduct an experiment evaluating DNNs in a similar fashion [29]. Their study uses two evolutionary algorithms (EAs) and a gradient ascent approach to produce images that the DNN will classify with high accuracy as a particular class but that are not human-recognizable images. These ”fooling images” show differences in how humans and computer vision algorithms perceive information. Another technique that produces so-called adversarial examples results in imperceptible changes which are applied to a test image to maximize the prediction error [41]. This results in a misclassification by the neural network.

Numerous articles make use of gradient-based approaches to find higher [9], [38], [29] or lower [41] activations for output units. These approaches are attractive due to their simplicity, but they tend to produce inputs that do not greatly resemble natural images. Instead they are computed using a collection of adversarial optimizations that happen to cause these high or low activations in the form of extreme pixel values, structured high frequency patterns, and copies of common motifs without global structure [9], [38], [29], [41], [15]. Several studies have researched why activations may be so affected by such optimizations. As mentioned above, Szegedy et al. noted that imperceptible changes can be made to correctly classified images to cause a misclassification [41]. Evolutionary algorithms

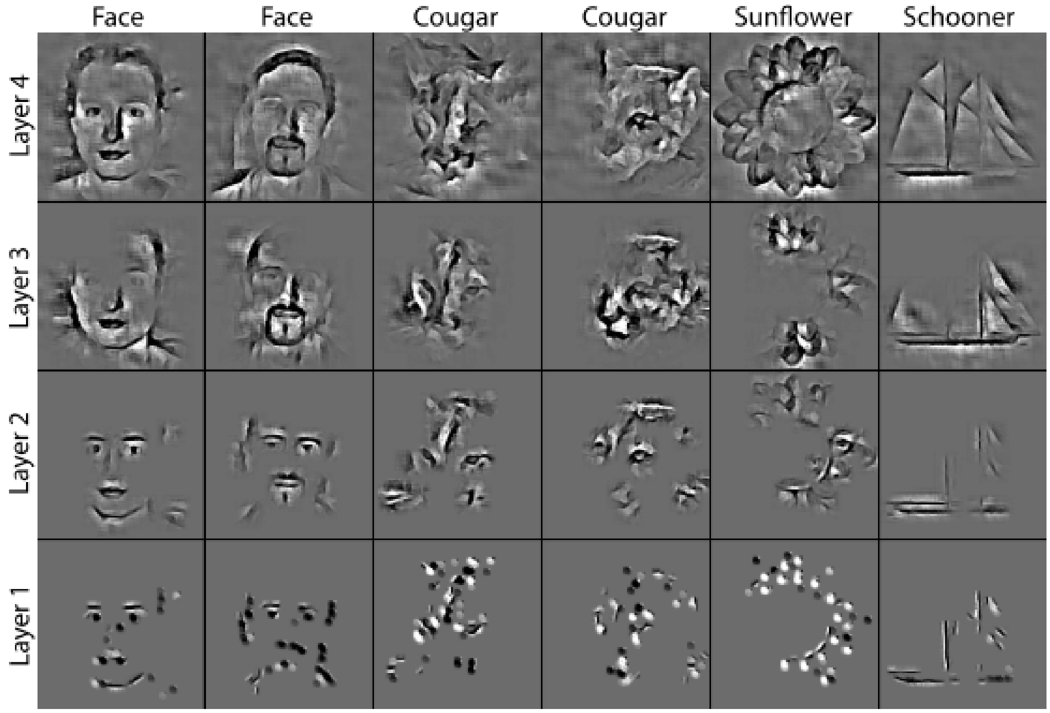


Figure 2.2

An example of visualization of image decomposition using the deconvolutional framework from [48].

can also produce unrecognizable adversarial inputs [29], and adversarial perturbations can be explained by the linear nature of neural networks [15].

Although these approaches produce unrecognizable images after optimization for high activations, there has been some research into producing discernable images using similar optimization techniques. One of these, [38] produces somewhat logically recognizable images in the final layers of a CNN by regularizing the optimization. This work also established that gradient-based visualization techniques generalize the Deconvolutional network reconstruction approach. Additionally, incorporating image priors, recovering the low-level image statistics removed during training, produced more interpretable information when looking at the images simulating a layer's activation for a specific input [28]. These regularizations are expanded upon with 4 new regularization techniques (L_2 decay, Gaussian blur, clipping pixels with small norm, and clipping pixels with small contribution) to bias images found via optimization to be more visually interpretable [47].

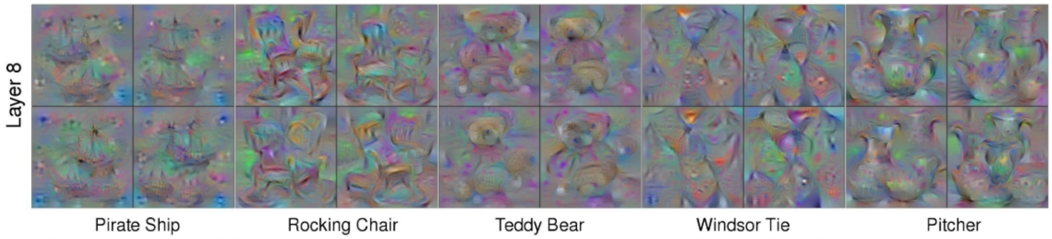


Figure 2.3

Visualization of features from the last layer of a deep, convolutional neural network. This is taken from 4 random gradient descent runs on 5 objects. This is an example visualization output from the second tool from [47] that applies the regularization parameters that produce preferred activation patterns in input space. It is one of the most interpretable visualizations of this type of data.

2.2 Material Classification

Material classification has been a growing area of research over the past decade and, as such, has encountered several datasets, techniques, and improvements. Specifically, this review will now look at the recent databases selected for material classification research, the material classification techniques as a whole, and the research revolving around CNN improvement as it is a common algorithm utilized for material classification.

2.2.1 Material Databases

Early work in the area of material classification and recognition focused on classifying specific instances of textures and material samples. The CUReT database [8] was a common choice during this time as it contains 61 material samples with 205 different lighting and viewing conditions for each. This focus led to research on the challenges of learning features that were invariant to pose and illumination. For example, Leung et al. [24] presents a framework for representing 3D textures to build a texton vocabulary that can describe generic local features and be used to predict the appearance of materials under novel conditions. Similarly, textons have also been created from frequency histograms of filter response cluster centers [45]. This method utilizes rotationally invariant, low-dimension, maximum response filter banks to aid in the classification of single, uncalibrated images using the CUReT database for all experiments performed. Also noted in this work is the need for a database that overcomes some of the limitations of the CUReT database such as the lack of multiple instances of the same texture that makes intra-class variation unobservable. As research continued in this field, databases with more diverse examples for each material class started to emerge. KTH-TIPS [4], [17] was an example of one of these and led to research of generalizations between examples of a class of material. On this same note, real-world textures and their attributes have also been explored [5].

Databases began to emerge with categorized materials such as FMD which contains 100 samples for each of the 10 material categories it contains. The images were taken from Flickr and represent a broad range of appearances for each of the categories while also incorporating real-world context into the samples. Although, FMD has been used in several material classification research papers, [27], [19], [31], and [36], improvements for classifing real-world materials was still needed. The OpenSurfaces dataset [2] addressed a few of the issues with FMD by introducing a larger amount of material samples with both object and material labels. However, with this introduction of samples, OpenSurfaces had some material categories that were undersampled compared to others. Recent work builds upon the OpenSurfaces database with millions of material samples in a new database MINC [3]. MINC follows a similar crowdsourcing pipeline as OpenSurfaces, but contains labeled images pulled from both Flickr images as well as Houzz images (professional photographs of staged interiors). OpenSurfaces and MINC both contain real-world context whereas CUReT, KTH-TIPS, and many of the experiments conducted with FMD cropped images to material samples only. This effort will utilize the MINC dataset as it contains real-world context and a much more vast number of samples including object and material labels. MINC is available online at http://minc.cs.cornell.edu/. Details of its construction are also available [1].

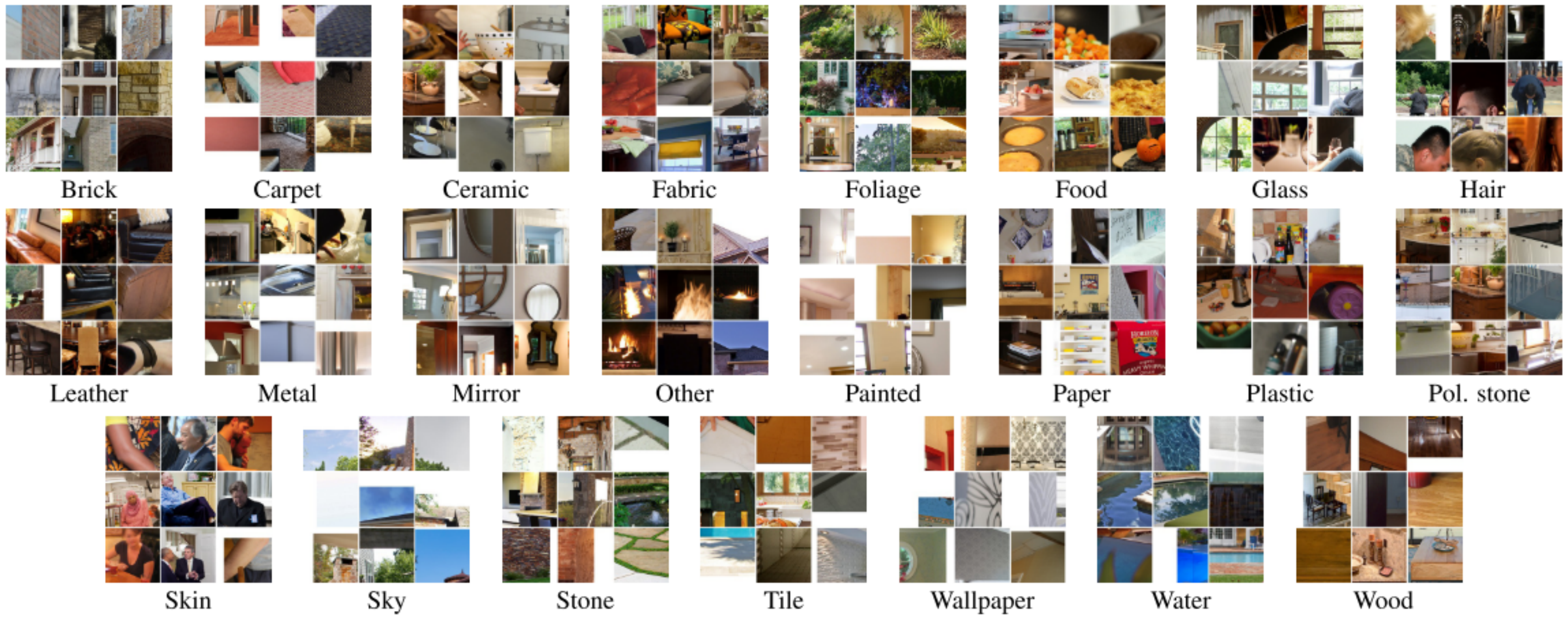


Figure 2.4

Image patches from the MINC database showing real-world context as well as a broad range of categories and variation within each individual category [3].

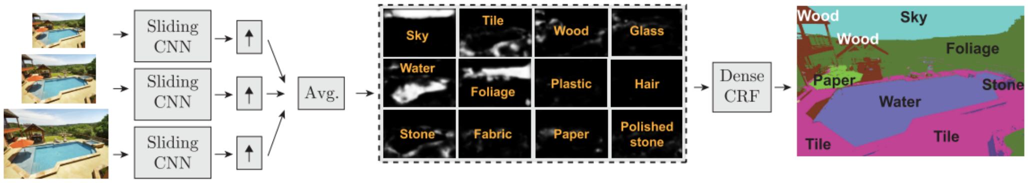
2.2.2 Material Recognition

There have been numerous studies focusing on material classification. Specifically, taking a material sample and classifying it into a material category or categories. These often used hand-picked features to classify these material samples. A set of rich, low and mid-level features have been used in combination with augmented Latent Dirichlet Allocation (aLDA) to combine the features under a Bayesian framework yielding an optimal combination of features which includes reflectance-based edge features, color, and SIFT [27]. Another set of features based on oriented gradients and variances of these gradients has also been used for experimentation [19]. A feature was introduced and utilized called pairwise local binary pattern (LBP) which is a rotation invariant feature [31]. A classifier utilizing LBP and dense SIFT features and a KTH-TIPS2-based database called Virtual texture under varying Illumination, Pose and Scales (MPI-VIPS) has been experimented with [25]. Material shaders, which are now contained in large libraries, are used to create this MPI-VIPS database by selecting shaders that supply specific pieces of information. A training-free texture classification technique has been shown that uses Oriented Basic Image Features (oBIF) for pixel level description and standard spatial pyramid matching (SPM) scheme for representing textures [43]. Material traits as a representation of per-pixel object-independent material information were introduced and shown to generalize well to novel datasets [34]. These material traits utilize features learned using the unsupervised learning technique of a convolutional auto-encoder (CAE) model. State-of-the-art performance was achieved on the KTH-TIPS2 and FMD datasets using Deep Convolutional-network Activation Features (DeCAF) and improved Fisher Vector (IFV) classifier [5]. Additionally, joint object and material classification has been shown to improve performance [19], [49]. Object recognition outputs have been incorporated into the kernel descriptors to improve material recognition [19], and attribute and object class labels have been predicted simultaneously for pixels as well as regions in an image showing that this can aid in image segmentation for both object classes and attributes [49]. As stated in the previous section, much of this work was conducted on material samples without real-world context.

The technique utilized in this thesis and introduced in recent research [3] is chosen as a baseline in this paper due to its state-of-the-art performance on the vast MINC datast containing millions of samples with this real-world context. It uses a CNN combined with a fully connected conditional random field (CRF) to predict the materials at every pixel in the image. This effectively produces full-scene material recognition and segmentation. This method improves upon the best prior method [5] on material classification on FMD by incorporating AlexNet features instead of DeCAF. It then finds that a fine-tuned AlexNet CNN is better on MINC than the linear SVM used to achieve these state-of-the-art results on FMD. Figure 2.5 portrays the full-scene material classification and segmentation process [3].

2.2.3 Convolutional Neural Networks (CNNs)

CNNs were introduced decades ago but have recently seen vast improvements in several fields including object classification. Much of this success has been driven by large datasets, competitions for optimizing these algorithms, and a substantial uptick in research



(a) Multiscale input (b) Probability map (12 of 23 shown) (c) Predicted labels

Figure 2.5

Pipeline used for full scene material classification in [3]. This shows various scales of images with a sliding window CNN that predicts a probability map of materials. From here a dense, fully-connected CRF ([22]) is applied to produce full-scene material classification and segmentation.

in recent years. Specifically, the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) has driven much of this success [32]. Numerous CNN architectures have resulted from the research that has ensued [39], [40], [50], [35]. The work by Krizhevsky et al. [23] showed a high level of success with the AlexNet network that encouraged further research producing the GoogleNet architecture [40]. While AlexNet and GoogleNet were both applied to the task of image classification, CNNs have also been successfully applied to the tasks of object detection and localization as well as per-pixel material segmentation. Girshirk et al. [13] uses Regions with CNN features (R-CNN) which uses region proposals that it then extracts feature vectors from using CNNs. These vectors are then used in combination with linear SVMs for object class detection. Another example is OverFeat [35] which applies an integrated framework for convolutional networks to classify and localize objects. It learns to predict boundaries in the images by accumulating bounding boxes for objects. It was the winner of the ILSVRC2013 localization task. Farbet et al. [12] utilizes a mutli-scale convolutional network approach to predict the material at every pixel in an image. This method extracts dense feature vectors and alleviates the need for engineered features. Oquab et al. [30] employs a sliding window CNN trained on a large dataset, ImageNet in this case, and reuses layers to compute mid-level image representations for patch classification on PASCAL VOC [11] dataset. The technique utilized for the research of this thesis [3] builds upon this research by using a sliding window CNN at multiple scales to compute a probability map of material categories. It then uses a fully-connected CRF to produce a full-scene material classification and segmentation (Figure 2.5).

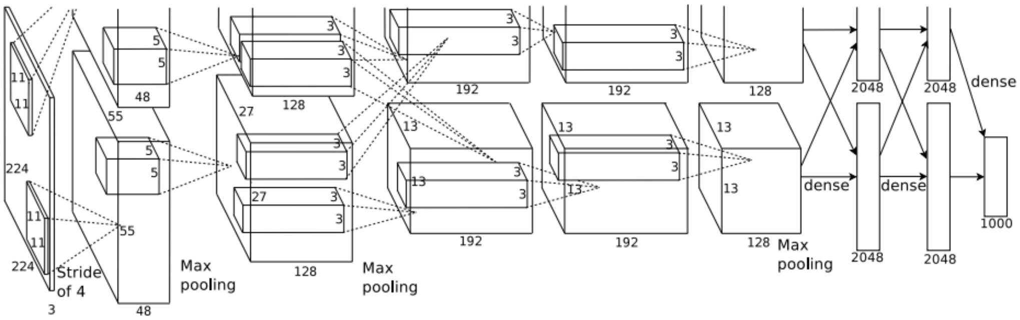


Figure 2.6

AlexNet CNN architecture showing repetitive convolution and pooling as well as the flattening of features going into a fully connected network that outputs a classification. This particular image shows the separation of responsibilites between two GPUs ([23]).

CHAPTER 3

METHODOLOGY

The following subsections describe the methodology that was used to conduct the experiment and research. This paper begins by examining an expectation of results based on the literature review and intuitions about the research itself. Next it outlines the contributions of the research outlining the objectives of the experiments. Lastly, an elaboration of the specific approach to the experimentation process is presented.

3.1 Hypothesis

Completing an analysis of the trained CNN utilized for material classification in a specific successful material classification algorithm will yield insight into the functionality of the CNN and algorithm as a whole. This insight will lead to discussion of how this compares to previous methods and will ultimately point to areas of improvement for the material classification algorithm.

3.2 Contribution

The following are the contributions of this research:

- An in-depth analysis and more comprehensive understanding of a state-of-the-art material classification algorithm [3] using deep visualization tools [47].
- An analysis of the deep visualization tools through utilization.

- As a result of the analysis of the material classifier, comparisons are made to previous methods of material classification for intuitions corresponding to modifications to the classifier that show potential of leading to improved classification performance.

3.3 Methods

This research begins by replicating the material classifier introduced in recent work [3]. This replication provides for numerous purposes. It not only serves as a reinforcement for the research conducted by Bell et al. [3], but also supports several objectives of this thesis. This includes both serving as a baseline for the comparison to previous material classifiers, and, in the same manner, providing a baseline to which supplemental modifications can be suggested.

Once this research [3] has been repeated to a sufficient degree of similarity, deep visualization tools [47] are utilized. This involves an integration of the material classifier and these tools as well as an augmentation of the tools available in the visualization toolbox. An in-depth analysis of the material classifier ensues which highlights the explicit functionality of the classifier at each layer (and neuron). Additionally, this step in the research also results in a thorough review of the visualization tools.

Subsequently, conjectures are created for the accuracy improvement of the material classification algorithm. These predictions reference previous methods heavily to portray historical evidence for their validity. These take an in-depth look at the material classifier's features and utilize the knowledge gained from the analysis to make these comparisons. The expectations of these suggested modifications are elaborated to portray well-detailed instruction for further research.

The results of the above research are presented in accordance with the contributions of this work producing analyses of both the material classification technique and deep visualization tools as well as augmenting both and presenting recommendations for improvements to the material classifier.

CHAPTER 4

EXPERIMENT

In the next few subsections, the design of the experiment is presented in detail. This includes three steps: 1) Replication and implementation of the material classifier [3] including code additions, 2) Utilization and augmentation of the deep visualization tools [47], and 3) In-depth Comparison of material classifier to previous methods providing recommendations for improvements.

4.1 Material Classifier Replication

The state-of-the-art material classifier is taken from recent work by Bell et al. [3], and the trained Caffe model is available for download along with the MINC dataset as mentioned in **??**.

To replicate the results of Bell et al. [3], a copy of this model, the trained weights, and dataset are downloaded from http://minc.cs.cornell.edu/ [3]. The Caffe deep learning framework version 1.0.0 is then installed within a Ubuntu 18.04 operating system environment. Installation instructions can be found at `https://Caffe.berkeleyvision.org/andcodeathttps://github.com/BVLC/Caffe/` [20].

To begin, the test.py script included with the material classifier was utilized along with the MINC dataset to compare the accuracy to the stated accuracy [3]. The model utilized for this test and comparison was the GoogLeNet Caffe model provided in the MINC download.

The next step to further this replication was to follow the implementation procedure for this CNN [3] to produce a dense grid of predictions across an input image. The GoogLeNet CNN was converted to a fully-convolutional classifier to produce a dense grid. Accomplishing this required the last fully-connected layer of the CNN to be converted to a convolutional layer. This process is well represented in the net surgery example in the Caffe github link above. Additionally, the pool5 layer of the GoogLeNet CNN was changed to have a kernel of size 1. These changes allowed for the classifier to output a material prediction every 32 pixels. While Bell et al. [3] shifted the input image by half-strides to produce a prediction every 16 pixels, this thesis does not since it only provides minor improvement in mean class accuracy (.2%) across segments and zero improvement across clicks. Its computational overhead was also a factor in this decision.

Up to this point, this thesis has only discussed experimenting with the CNN used for patch material classification as it is the main focus of this research. However, to complete a further analysis of the research and classifier, an analysis of the material classification technique up to the point of CRF implementation is observed.

A sliding-window approach was implemented using the fully-convolutional CNN from above that produced a classification output map at every patch across the full-scene image. Then, multiple scales are implemented. The input image is resized so that a patch maps to a 256x256 square. This, for example would resize an image's smaller dimension *d = 256/s* where *s* is the trained patch scale (23.3% in this case). This is repeated for *d/sqrt(2)* and *d*sqrt(2)*. Padding is then added to align the output probability map to the input image. After calculating the predictions in a sliding window manner across these 3 scaled input images, the output probability maps are upscaled to have a smaller image dimension of 550. The average of these 3 maps for each material is taken as the final output probability map for that material [3]. This final output probability map is viewed as a full-scene material classifier.

A brief analysis of this full-scene classifier is presented in the results section along with a comparison to the full-scene material classifier accuracy of Bell et al. [3].

It should be noted that this paper does not train and implement the CRF [22]. The reason for this is in regards to introducing more variability via training this CRF, thus establishing multiple independent variables and adding uncertainty to the cause of the results. This also gives this paper a metric for the improvement offered by the CRF implemented in Bell et al. [3]. It should also be noted that Bell et al. [3] experimented with several patch scales, CNNs, and database sizes. For the purpose of simplicity and due to the architectures and code provided by the referenced research, this thesis will only observe the GoogLeNet CNN with the above specified implementation. Additionally, the code for the converted CNN for full-scene material classification as well as the the code for classifying a full-scene image using this classifier is added and can be seen at `https://github.com/jdonovanCS/material_classification/`.

4.2 Neural Network Visualization

The main visualization tools utilized within the research of this paper, is the deep visualization toolbox and included scripts from recent research by Yosinski et al. [47]. The scripts were obtained from the corresponding github repository for this research (`https://github.com/yosinski/deep-visualization-toolbox`) [47]. Steps are included for installation in the README.md file of this repository and include installing and configuring Caffe, installing the modified version of Caffe for the toolbox, installing dependencies, and installing the deep visualization toolbox. The settings file also needed to be modified to point to the GoogLeNet Caffe model and weights corresponding to the material classifier discussed above.

Without any further modifications, the toolbox allowed for the user to view some information about the material classfication CNN. The activations of each node, the deconvolution of these activations, and an input image (or input from camera) are now viewable. Additional scrips are included to select the images or crops of images that cause the highest activations for each node as well as to optimize synthetic images to cause high activations for each node. These scripts were utilized to visualize exactly what patterns the CNN was looking for at each node and images that contained the most closely matching patterns to these. The optimized synthetic images mentioned above were generated using the parameters from Table 4.1. The most visually informative of these sets of parameters was chosen to generate the optimized images for every node at each convolutional layer (second row of parameters from Table 4.1). They were then cropped and resized as were the image patches that caused high activations. The purpose of this was for easibility of viewing.

These images were populated into the toolbox as was intended so that the visualization tools were as comprehensive and understandable as possible. The top 9 images that cause the highest activation as well as a single optimized image are generated for each node at each convolutional layer. Once all of these visualizations were in place in the deep visualization toolbox, the toolbox was used to interpret the CNN to understand exactly what it had learned and glean information about how to improve it based on this knowledge.

Table 4.1

Table 1 in [47] represents the paramaers used to generate the optimized images. The second set was used in the toolbox itself in this thesis.

decay	blur width	blur every	norm pct clip	contribution pct clip	iterations
0	0.5	4	50	0	500
0.3	0	0	20	0	750
0.0001	1.0	4	0	0	1000
0	0.5	4	0	90	1000

The specific augmentations produced by this paper to the visualization tools are: 1) automated layer-based and network-based scripts for optimizing synthetic images 2) script for cropping images to layer-based filter size and 3) scripts for combining images into single file for toolbox. The automated layer-based and network-based scripts allow for the parameters to be specified in the exact same manner as the optimization script in the toolbox. The layer-based scripts allow for manual parallelization of scripts to maximize utilization of computational power and were hand-designed for the GoogLeNet architecture's convolutional layers (as was the network-based script). The script for cropping images allows for specification of crop sizes. These sizes can be computed using the technique present in the script to find max images and assumes the optimizations are centered in the image. Lastly, the script for combining top 9 images can be edited to point to the file locations of these images and assumes there are 9 files in the folder with a naming convention. The script for combining optimized, cropped images works very similary. These modifications can be found at `https://github.com/jdonovanCS/dvtb/`.

4.3 In-depth Comparison of Material Classification Methods and Recommendations for Improvement

An in-depth comparison to previous methods from literature is performed to point to areas of potential weakness, strength, and difference within the CNN investigated. This involves, observing the learned features of the CNN through the visualization toolbox, but it also involves computing similarities between features from past works where applicable. Visualizations of features and the benefit of these features and corresponding techniques are observed in detail. Each of the sources from related **??** subsections are elaborated descriptively and compared to the features learned and the performance of the CNN investigated in this thesis. This comparison is split similarly to the **??** section for comprehensiveness and readability. Features from past works are visually compared to those of the CNN as are the performance metrics and overall techniques.

Once this comparison has been made, recommendations are stated for improving the material classifier observed here. These recommendations stem directly from the visual and theoretical comparisons of the datasets, features, and techniques for material classification compared.

CHAPTER 5

RESULTS

This chapter will present the findings and results of the experiments. This will include an analysis of the material classifier via employment of the visualization toolbox, an analysis of the visualization toolbox and its scripts in the attempts to understand the material classifier, and the results of the in-depth comparison made between the material classifier and previous techniques with recommendations for modifications for the purpose of improvement. The analysis of the visualization toolbox also includes the results of the augmentation scripts contributed by this research, and the material classification analysis section includes the results for the code additions in that area.

5.1 Material Classifier Analysis

One of the questions at the forefront of this research regards what this material classifier actually learned during training. The deep visualization toolbox is able to portray a better picture of this than previously known. Before employing the visualization toolbox, one of the only methods for observing the learned features of the material classifier was by observing its accuracy and final output when classifying an image.

The first step to analyzing the material classifier explored in this paper is to compare the results of using it on a test set to the result of the original work [3]. This was completed by taking the provided architecture of GoogLeNet trained on materials and applying it to the 2500 material samples also provided by the work from Bell et al. [3] to verify the accuracy. The accuracy tables can be seen here (Table 5.1 and Table 5.2). The overall accuracy was actually slightly higher. Some categories went up or down by a few percentiles. This is most likely due to the different numbers of samples in the training and test set in the original research [3] whereas in the 2500 patch images provided, each category had the same amount of samples.

Table 5.1

Accuracy percentages of the GoogLeNet architecture recorded on the test images from [3].

Sky 97.3	Food 90.4	Wallpaper 83.4	Glass 78.5
Hair 95.3	Leather 88.2	Tile 82.7	Fabric 77.8
Foliage 95.1	Other 87.9	Ceramic 82.7	Metal 77.7
Skin 93.9	Pol. stone 85.8	Stone 82.7	Mirror 72.0
Water 93.6	Brick 85.1	Paper 81.8	Plastic 70.9
Carpet 91.6	Painted 84.2	Wood 81.3	**Total: 85.2**

The next step is to analyze what the CNN has actually learned to perform this classification. The deep visualization toolbox shows that in the first few layers of the CNN, the classifier begins filtering based on very simplistic color and edge based patterns (Figure 5.1). This is logically understandable since convolutions usually begin with these simple patters for filtering. There are multiple convolutions for each layer of the neural network. In this

Table 5.2

Accuracy of the provided weights and architecture of GoogLeNet from [3] on the 2500 image patches provided. This was used for verification purposes.

Sky 96.8	Food 95.1	Wallpaper 89.4	Glass 83.3
Hair 95.6	Leather 84.5	Tile 82.8	Fabric 78.2
Foliage 94.9	Other 90.4	Ceramic 85.1	Metal 75.6
Skin 94.8	Pol. stone 85.9	Stone 88.8	Mirror 69.0
Water 92.4	Brick 84.4	Paper 89.9	Plastic 77.4
Carpet 91.8	Painted 82.8	Wood 78.2	**Total: 86.4**

manner, the CNN employed here takes a brute-force by generated filter approach to detecting differing lighting, pose, and orientation conditions. Additionally the training set was augmented to learn various orientations and stretches in [3] on which this work builds.

An exploration deeper into the neural network in question reveals slightly more complex edges, such as curls and curves, being filtered upon as well as simplistic shapes, such as circles as seen in Figure 5.2.

After the pool3 layer, the features learned by the CNN become increasingly more complex combining colors and edges (Figure 5.4). Simple patterns start to build to form larger more complex patterns, and the filters themselves begin to look at larger portions of the image. These pattern filters start to recognize texture and other subtleties within the data.

As it gets deeper it begins to filter certain patterns that are 2-dimensional in the image, but correspond to 3-dimensional objects, such as the corners of furniture or the concavity of human eyes (Figure 5.5) and recessed upholstery buttons. Context becomes a feature of growing importance in these later layers of the CNN. We can also begin to see the

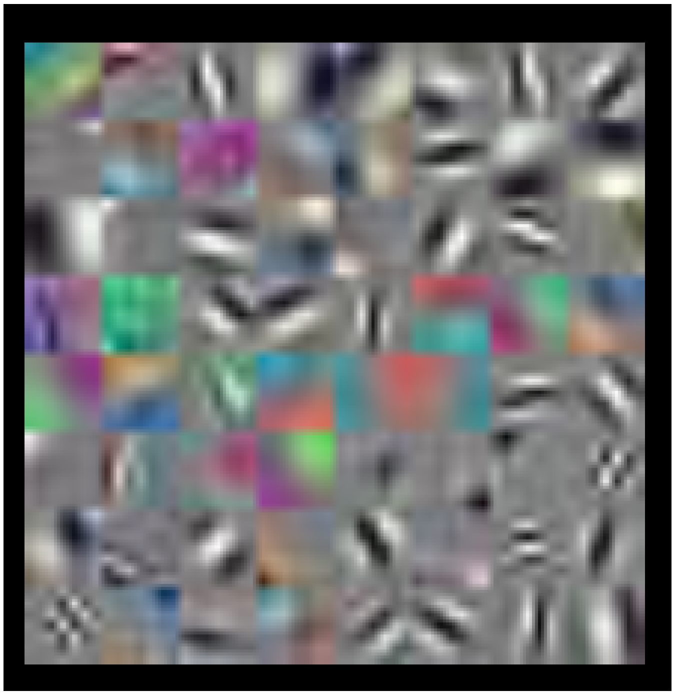


Figure 5.1

The convolutional kernels used in the first layer of GooogLeNet for material classification.

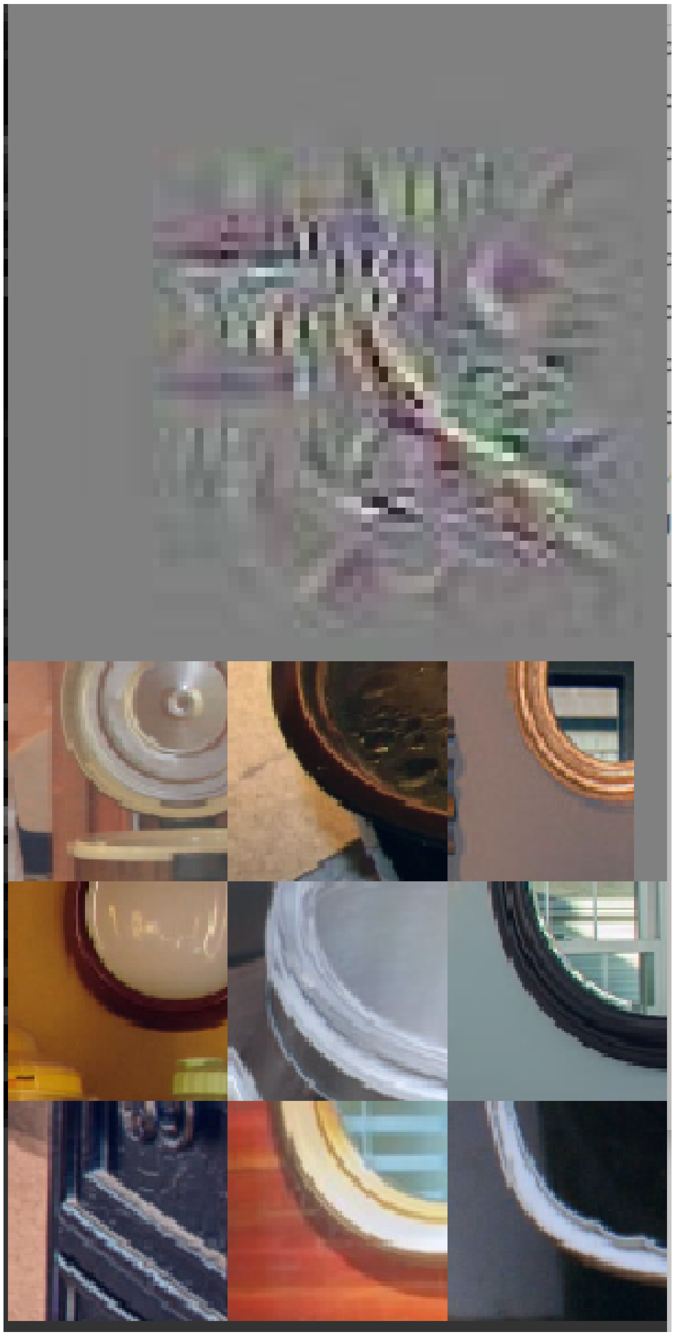


Figure 5.2

Optimized input for node 11 of the inception 3a 5x5 convolutional layer of the CNN used in this thesis.

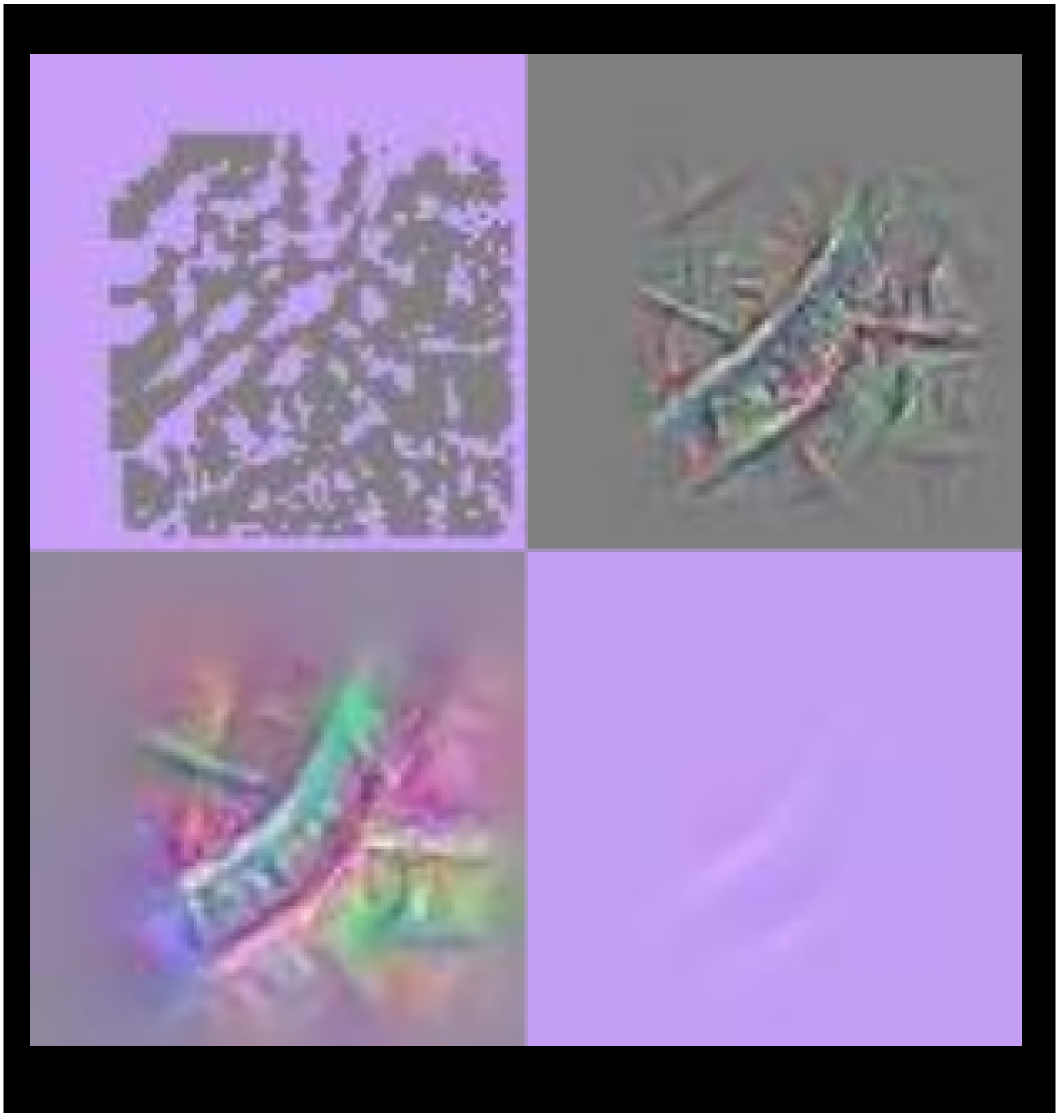
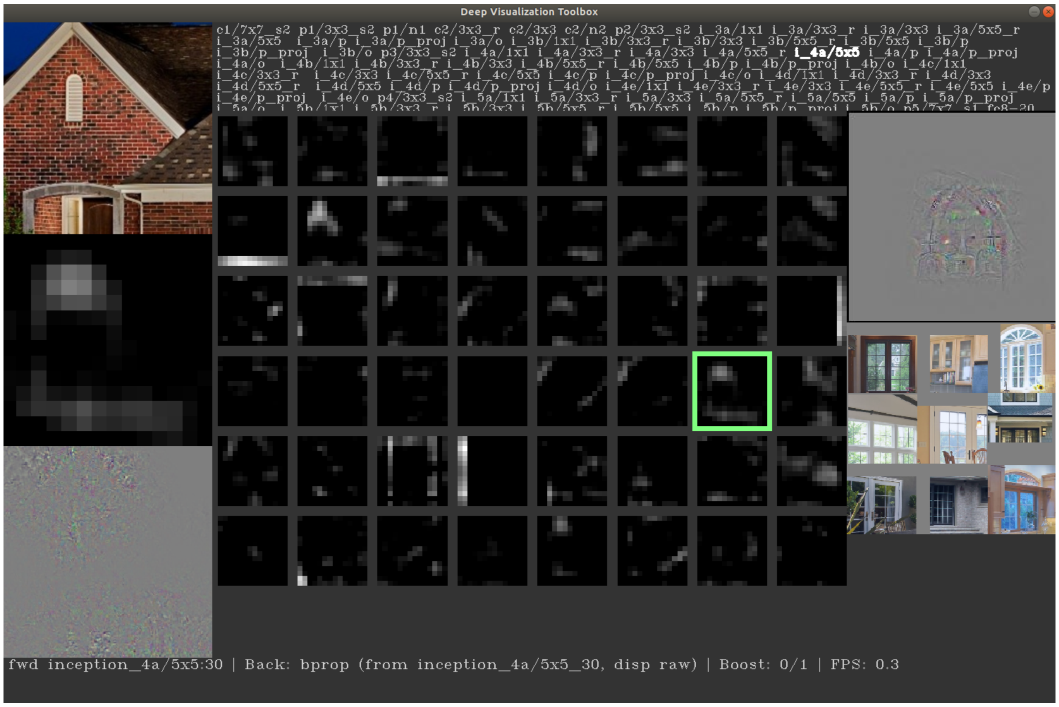


Figure 5.3

Optimized images of a different node at the inceptio 3a 5x5 layer of the network used here.


Deep Visualization Toolbox
c1/7x7_s2 p1/3x3_s2 p1/n1 c2/3x3_r c2/3x3 c2/n2 p2/3x3_s2 i_3a/1x1 i_3a/3x3_r i_3a/3x3 i_3a/5x5_r
i_3a/5x5 i_3a/p i_3a/p_proj i_3a/o i_3b/1x1 i_3b/3x3_r i_3b/3x3 i_3b/5x5_r i_3b/5x5 i_3b/p
i_3b/p_proj i_3b/o p3/3x3_s2 i_4a/1x1 i_4a/3x3_r i_4a/3x3 i_4a/5x5_r i_4a/5x5 i_4a/p i_4a/p_proj
i_4a/o i_4b/1x1 i_4b/3x3_r i_4b/3x3 i_4b/5x5_r i_4b/5x5 i_4b/p i_4b/p_proj i_4b/o i_4c/1x1
i_4c/3x3_r i_4c/3x3 i_4c/5x5_r i_4c/5x5 i_4c/p i_4c/p_proj i_4c/o i_4d/1x1 i_4d/3x3_r i_4d/3x3
i_4d/5x5_r i_4d/5x5 i_4d/p i_4d/p_proj i_4d/o i_4e/1x1 i_4e/3x3_r i_4e/3x3 i_4e/5x5_r i_4e/5x5 i_4e/p
i_4e/p_proj i_4e/o p4/3x3_s2 i_5a/1x1 i_5a/3x3_r i_5a/3x3 i_5a/5x5_r i_5a/5x5 i_5a/p i_5a/p_proj
i_5a/o i_5b/1x1 i_5b/3x3_r i_5b/3x3 i_5b/5x5_r i_5b/5x5 i_5b/p i_5b/p_proj i_5b/o p5/7x7_s1 fc8-20

```
fwd inception_4a/5x5:30 | Back: bprop (from inception_4a/5x5_30, disp raw) | Boost: 0/1 | FPS: 0.3
```

Figure 5.4

The full visualization toolbox with all of the bells and whistles. The particular node being looked at is node 30 from the inception 4a 5x5 layer of the network. It is beginning to filter out smooth, recessed glass entryways showing a larger area with context being used for classification.

importance of intra-class variation in these later layers. A pattern can correspond to many materials and each material can correspond to several patterns.

.

As the CNN investigated begins nearing its final layers, very complex pattern recognition including combinations of colors, shapes, and edges begins to be observed (Figure 5.6). The CNN begins unknowingly performing simple object recognition as it filters out the complex patterns, some of which correspond very closely with a specific object (Figure 5.7). The CNN also begins filtering out foreground versus background objects in many of the images as well. This was not an overall goal of the CNN, but since it is looking most closely at the objects in the foreground, it is an unintended consequence. Many of the filters formed at this stage are difficult to interpret as they become very complex.

Lastly, In addition to the above evaluation of the patch material classification CNN, the full-scene classifier output before CRF implementation can be seen in Figure 5.8. A full-scene classification with the CRF can be seen in Figure 5.9 for comparison. This shows the value of adding a trained CRF to the system. The lines for classification are much smoother. The CRF from [3] can be seen when trained on clicks as well as segments for comparison. The code for the full-scene material classification provided by this thesis is an addition made to the code that [3] made available. This included converting the network to fully-convolutional as well as sliding window; applying it at multiple scales; upsampling; and averaging output maps for final full-scene predictions.

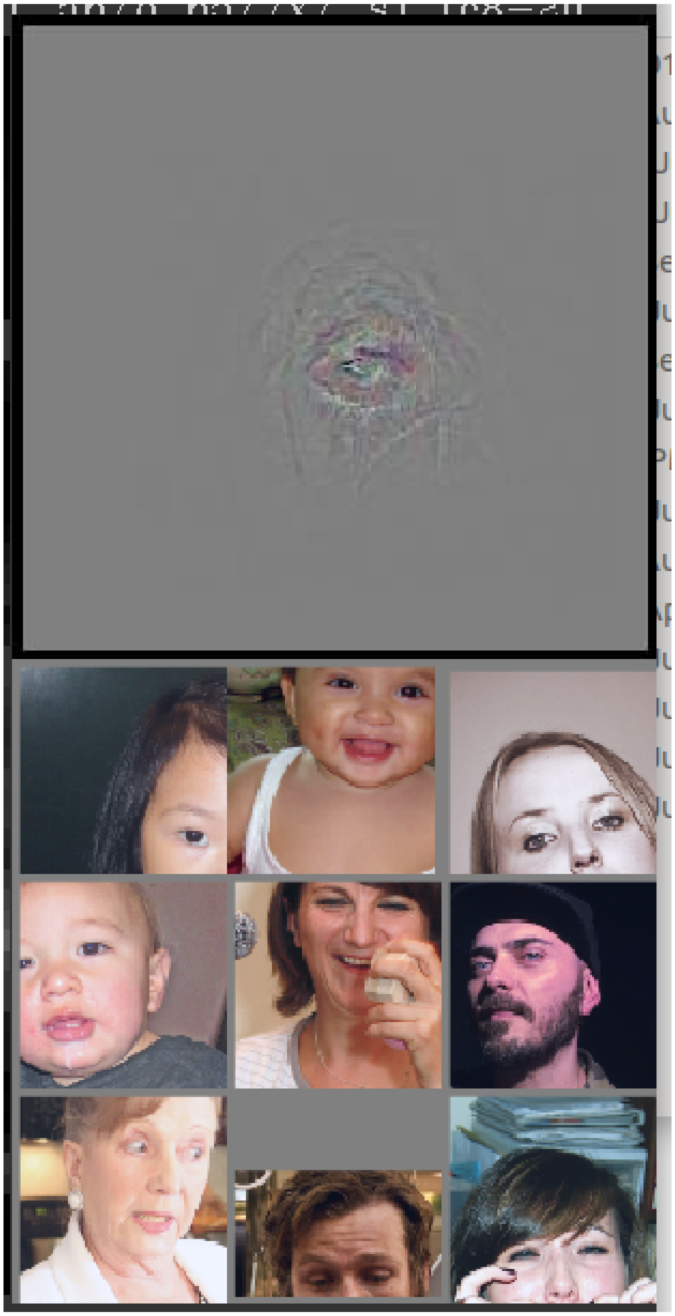


Figure 5.5

The optimized image and corresponding top 9 images from the set of 2500 patches can be seen here. This node is filtering images for an eye showing 3-dimensional filtering.

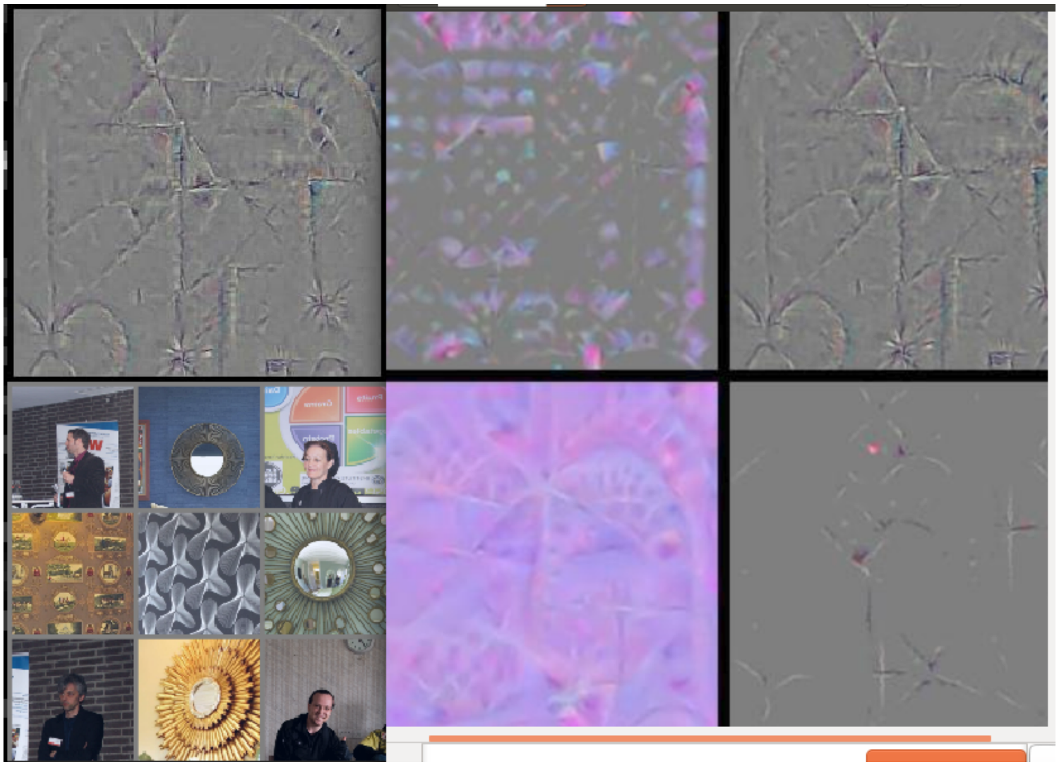


Figure 5.6

This shows the optimized images and top 9 images for a node at the inception 5a 5x5 layer of the network (nearly the end) performing complex pattern recognition that is difficult to understand.

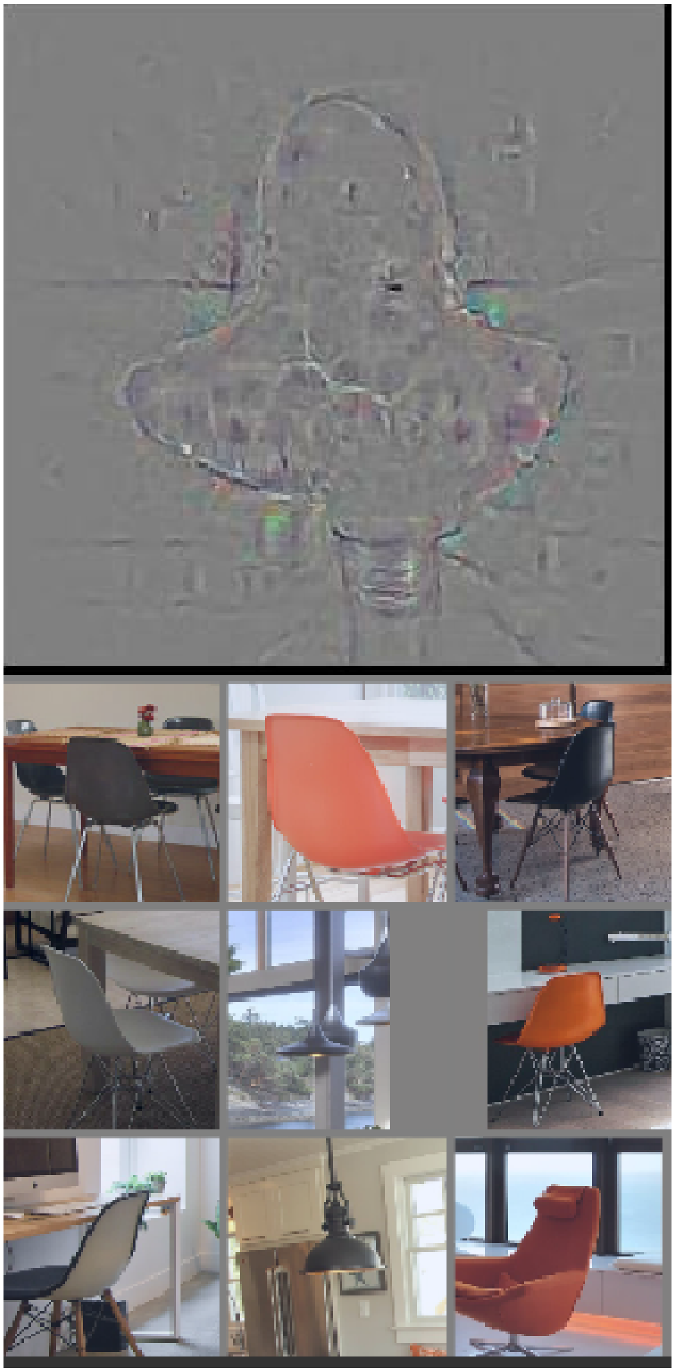


Figure 5.7

This shows the optimized image and top 9 images for a node at the inception 5a 3x3 layer of the network (nearly the end) performing object recognition (unknowingly) on a specific type of chair.

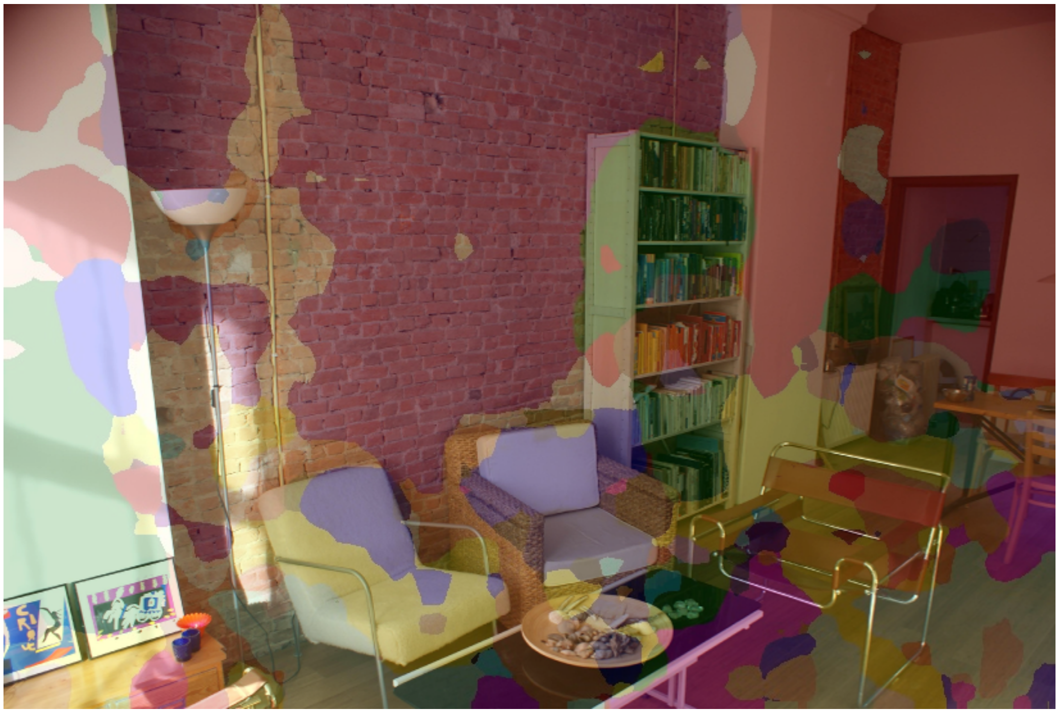


Figure 5.8

A visualization of material classification (color-coded) of a full-scene image using the code provided by this thesis (before CRF implementation) to show the discontinuity across materials and confidences

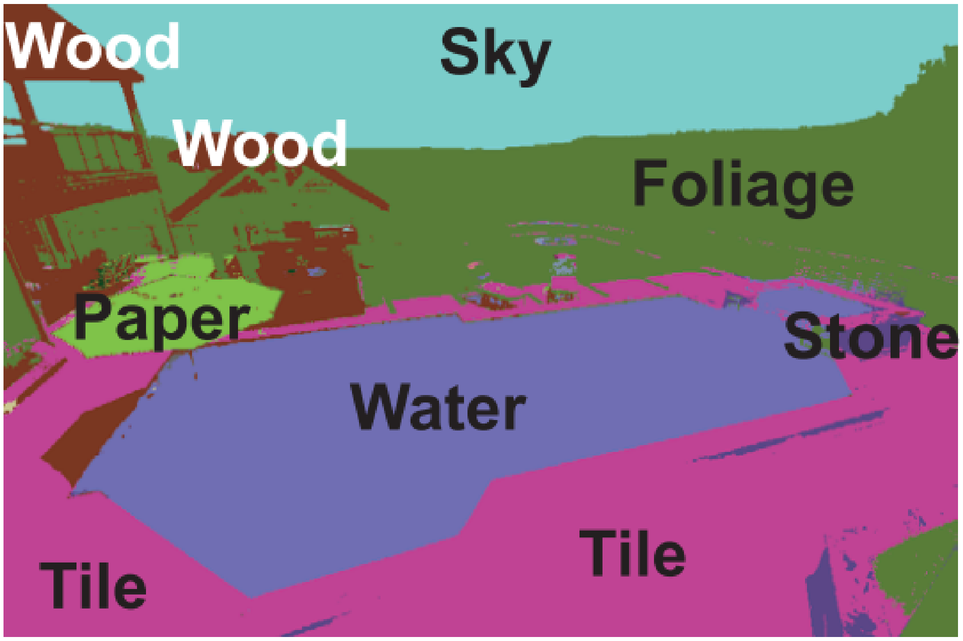


Figure 5.9

Visualization of material classification (color-coded) using the classification mechanism described in [3] that includes a CRF.

5.2 Visualization Tool Analysis

The deep visualization toolbox from [47] was utilized to analyze and understand the patch material classification CNN. This toolbox was extremely helpful in this regard as it supplied a visualization of an input, visualizations of each nodes' activations, the deconvolution of these activations, visualization of the optimized input for each node, and visualization of the top X (9 in this case) images from the training set with regard to each node.

The toolbox has some basic controls for navigation and view modification. These controls include the ability to switch layers and nodes as well as the ability to change the input device. Additionally, it includes controls to change the parameters for the deconvolution and activation views.

Allowing for the input image to change and even for a camera to be used to capture the input was helpful because it allowed for multiple images to be scanned through or for a changing camera input to be used to visualize the activations of the specific neuron the user was observing. This allowed for the user to examine how the activations change with the input and what portions of the image cause the highest activations.

The activations, in the largest view portion of the toolbox, were useful due to their ability to convey the portion of the image that contributed the most weight to the particular node in question. In early layers this may have been quite obvious, but in later layers this becomes more perplexing since the filters become very complex. The activations in conjuction with each of the other visualizations provided by the toolbox allow it to portray valuable information about the CNN.

The deconvolution of these activations which supplies similar information to the user as does the activations. This complementary information helps the user decipher the portion of the input that is causing each neuron's activations to fire.

The input visualization, node activations visualization, and deconvolution of the node activations are all included in the deep visualization toolbox at a minimum. They are each generated using the network structure, weights, and input which do not require external scripts or computation. These are considered the network-based approaches to visualization. However, the optimized input for each node and the visualization of images from the training set that cause the highest activations do require use of external scripts and are considered the data-centric approaches to visualization. Augmentations to these scripts were included as a portion of the work of this thesis.

The optimized input for each node of the network is augmented with additional scripts for automating an entire network's or layer's neurons to be optimized sequentially instead of each individual neuron's optimized input needing a manual process to be started. This improves the speed at which the optimized input are generated for each node. The optimized images are extremely valuable to the visualization toolbox as they provide an insight into exactly what this node is filtering and what is causing the activations. Several parameters can be given to the optimization script (Table 4.1) generating these images and each one was tested (Figure 5.10). The optimizations visualized in the toolbox for this experimented resulted from the one that was the most visually appealing and highly informational. A script for combining the optimized images from various parameters runs is included and some of the visualizations from it are presented.

The images from the training set that cause the highest activations for each node can be obtained using a script that is included with the toolbox. This script looks at the images provided and picks the portion of that image that excites each node the most. This allows for us to look at an image that contains the feature(s) that the particular node-in-question is searching for which provides an even better picture of exactly what the CNN has learned. The user may choose the number of images to select that cause these high activations. The source paper [47], as well as this thesis, chooses 9. A script is also introduced by this work to combine these images for presentation in the toolbox. Additionally, the tool that is used to pick these top images can be used to choose the top images' deconvolutions for even more information. This work does not do utilize this portion of the tool, but it is worth recording its availability.

As a whole this toolbox and the external scripts were very helpful and the augmentations added to the capability providing more efficient and effective production of visualizations. This resulted in the tool being more informative since without these augmentations, it would be difficult or impossible to view the optimized images and top images in the toolbox. There were also a few settings files that needed to be tweaked in order for the toolbox to function correctly, but this was fairly trivial to perform. This augmented toolbox shows several helpful pieces of information for deciphering exactly what the neural network being inspected has learned and how its output is compiled. A full visual of the deep visualization toolbox can be seen in Figure 5.11.

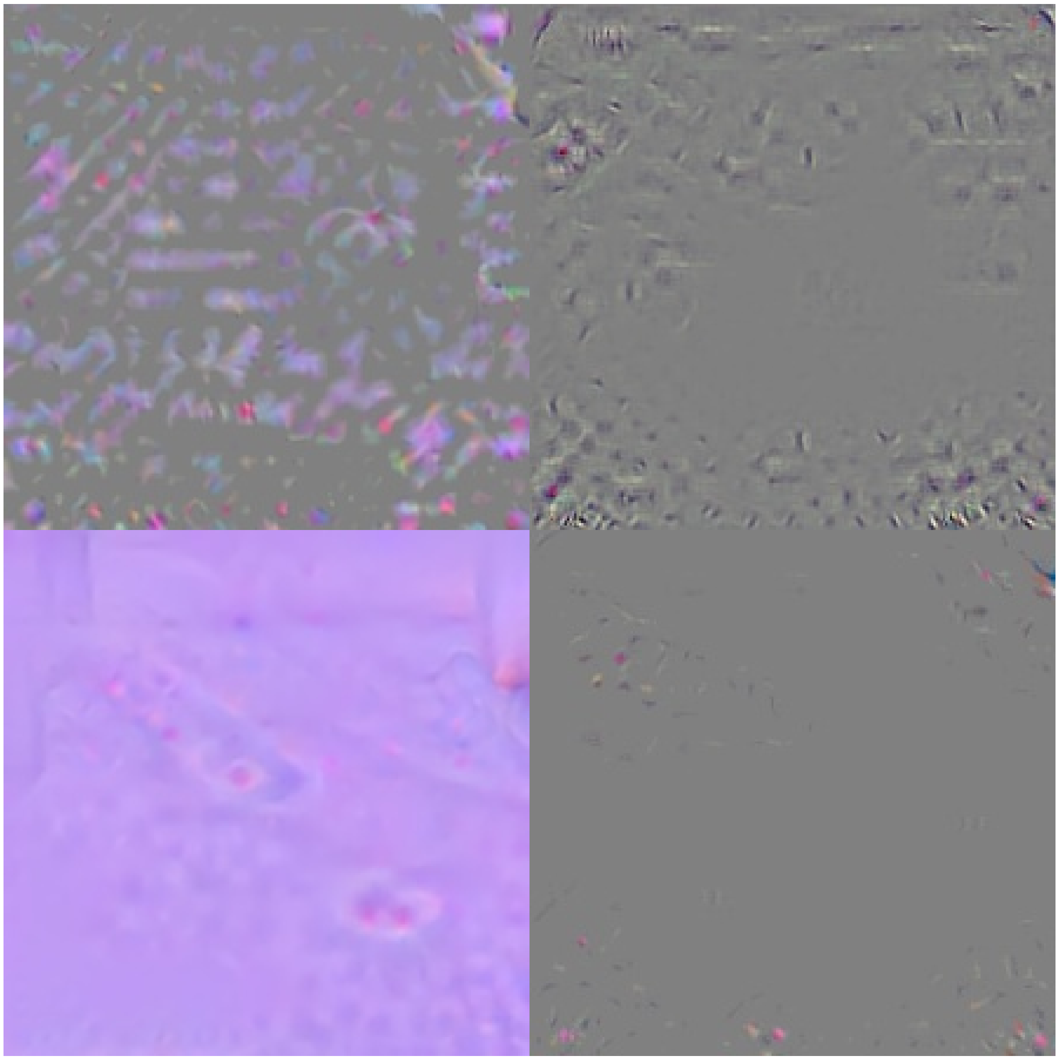
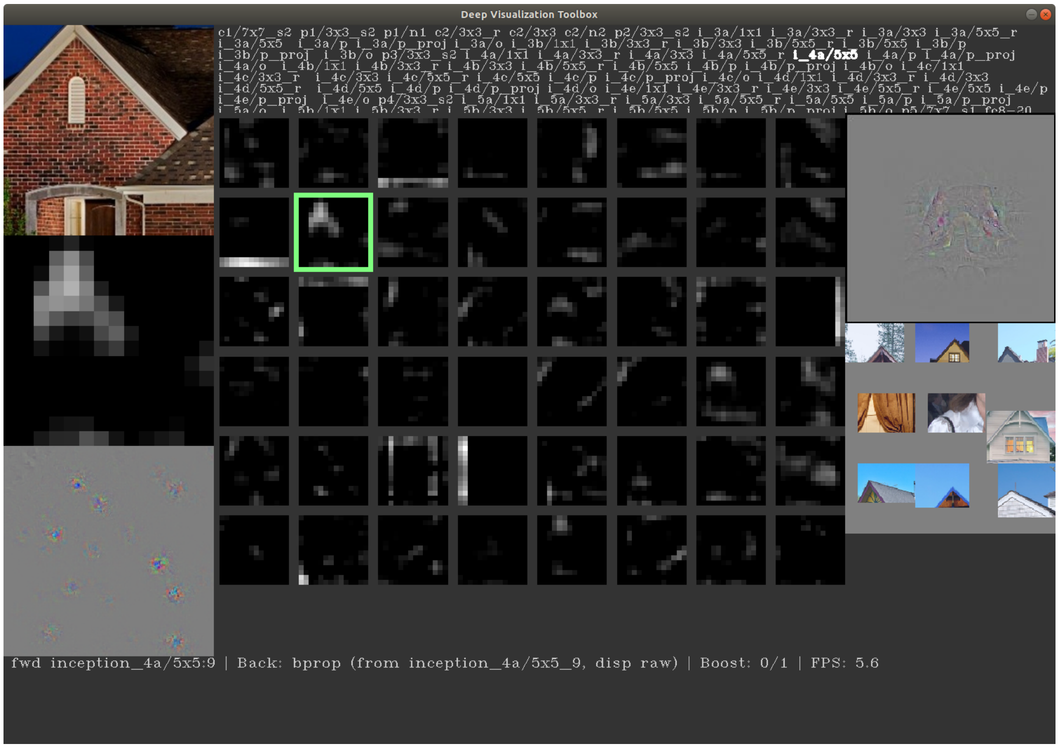


Figure 5.10

Example of optimized images. These optimizations are for the node representing carpet material in the last layer of the network explored here.



```
c1/7x7_s2 p1/3x3_s2 p1/n1 c2/3x3_r c2/3x3 c2/n2 p2/3x3_s2 i_3a/1x1 i_3a/3x3_r i_3a/3x3 i_3a/5x5_r
i_3a/5x5  i_3a/p i_3a/p_proj i_3a/o i_3b/1x1 i_3b/3x3_r i_3b/3x3 i_3b/5x5_r i_3b/5x5 i_3b/p
i_3b/p_proj  i_3b/o p3/3x3_s2 i_4a/1x1 i_4a/3x3_r i_4a/3x3 i_4a/5x5_r i_4a/5x5 i_4a/p i_4a/p_proj
i_4a/o  i_4b/1x1 i_4b/3x3_r i_4b/3x3 i_4b/5x5_r i_4b/5x5 i_4b/p i_4b/p_proj i_4b/o i_4c/1x1
i_4c/3x3_r  i_4c/3x3 i_4c/5x5_r i_4c/5x5 i_4c/p i_4c/p_proj i_4c/o i_4d/1x1 i_4d/3x3_r i_4d/3x3
i_4d/5x5_r  i_4d/5x5 i_4d/p i_4d/p_proj i_4d/o i_4e/1x1 i_4e/3x3_r i_4e/3x3 i_4e/5x5_r i_4e/5x5 i_4e/p
i_4e/p_proj  i_4e/o p4/3x3_s2 i_5a/1x1 i_5a/3x3_r i_5a/3x3 i_5a/5x5_r i_5a/5x5 i_5a/p i_5a/p_proj
i_5a/o  i_5b/1x1 i_5b/3x3_r i_5b/3x3 i_5b/5x5_r i_5b/5x5 i_5b/p i_5b/p_proj i_5b/o p5/7x7_s1 fc8-20
```

```
fwd inception_4a/5x5:9 | Back: bprop (from inception_4a/5x5_9, disp raw) | Boost: 0/1 | FPS: 5.6
```

Figure 5.11

A visualization of the deep visualization toolbox including each of its tools: input image (top left), activation for selected node (middle left), deconvolution of activation of selected node (bottom left), list of layers (top center), node activations (center), optimized input image for selected node (top right), top 9 images from set of 2500 patches (middle right).

5.3 In-Depth Material Classification Technique Comparison

This section will perform an in-depth comparison of previous material classification techniques in three separate sections: 1) Material Databases, 2) Material Features and Recognition, and 3) Neural Networks.

5.3.1 Material Databases

Table 5.3

Summary of the databases observed and conclusion of potential benefit to the dataset used to train the classifier observed in this thesis (The MINC dataset).

Name	**Samples and Technique**	**Benefit MINC?**
CUReT [8]	61 real-world textures, 205 images of each with several different viewing and lighting conditions	Possibly
KTH-TIPS [4], [17]	Supplements CUReT samples with 81 (9 scales, 3 directions and 3 poses) images for 10 of the materials. This shows better performance and flexibility on recognition tasks	Possibly
VIPS [25]	Added rendered samples to KTH-TIPS-based database showing the advantages of utilizing rendered and real-world samples	Yes
FMD [37]	First dataset to add real-world context to material samples. 100 images per 10 categories	No
MINC [3]	MINC extended the idea of the FMD database by maintaining real-world context and adding millions of samples as well as segmentations for training	No

The CUReT database [8] establishes the importance of capturing variances in pose and illumination in sample images by using 7 different camera angles taken utilizing a spherical representation of the material sample. This results in 205 viewing and illumination conditions captured for each of the 61 materials. These viewing and illumination positions can be seen in Figure 5.12 and Figure 5.13 respectively. The MINC dataset utilized for the experiment of this thesis contains a vast number of images from the real-world that contain various lighting conditions (Figure 5.19). Since these images are not generated in an exhaustive manner, there could be missing lighting or viewing conditions. One can observe and consider this possibility by imagining a typical scene in a house. While imagining this, the light was probably an overhead light or possibly a light at or above the viewing perspective. This is most likely because in the real world, most of the time, there is a standard orientation to objects, but this could limit the ability of the material classifier on input with novel lighting or viewing conditions. On the other hand, since this classifier is targeted to real-world use, it may actually be of importance that these orientations are learned.

.

The KTH-TIPS database was formed in [4] and [17]. [17] adds an additional variance to pose and illumination, scale, which takes into account the distance and size of the material in view. It also utilizes an SVM for classification purposes showing that an SVM has a higher accuracy and that scale is an important factor to consider in material classification (Figure 5.14). [4] finishes compiling the KTH-TIPS database with varied scale, illumination, and pose using a filter bank with varied scale (Figure 5.15). The CNN approach directly takes scale into consideration in its classification by training at multiple scales (Figure 5.16), and [3] experimented with patch material scales and varying scale on the full-scene classification performed (Figure 5.17). The convolutional filters in the CNN grow in size and learn the scale of the patch input they are convolving over, but there

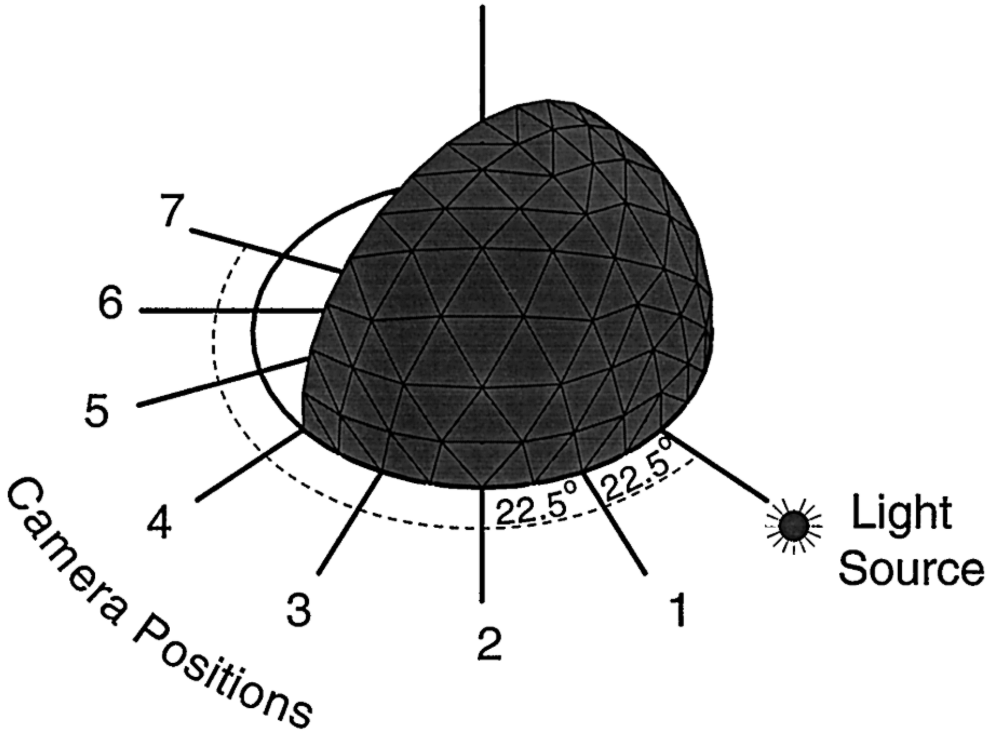


Figure 5.12

The above image represents the 7 different camera positions used in [8]. These seven locations correspond to angular deviations of 22.5, 45, 67.5, 90, 112.5, 135, and 157.5 degrees from the light source direction.

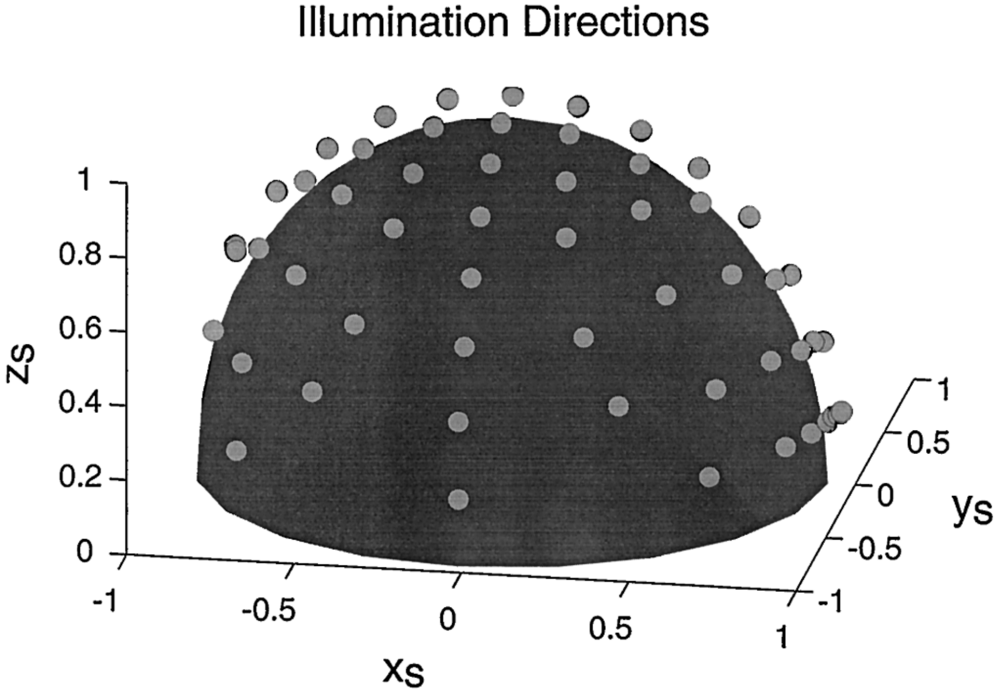
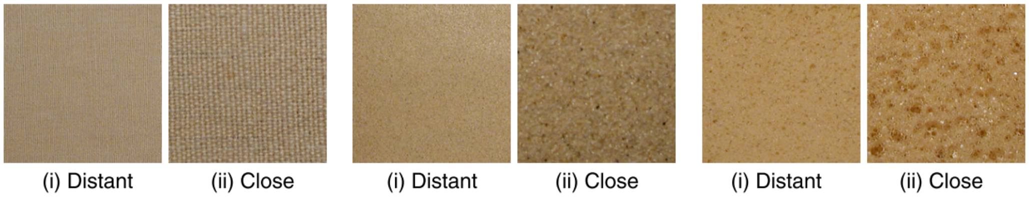


Figure 5.13

Each circular marker represents a distinct illumination direction. For each of these illumination directions, the sample is imaged from seven viewing directions, corresponding to the seven camera positions shown in Figure 5.12 [8]



(a) Cotton (b) Sandpaper (c) Sponge

Figure 5.14

An example from [17] of how appearance can differ with distance (scale).

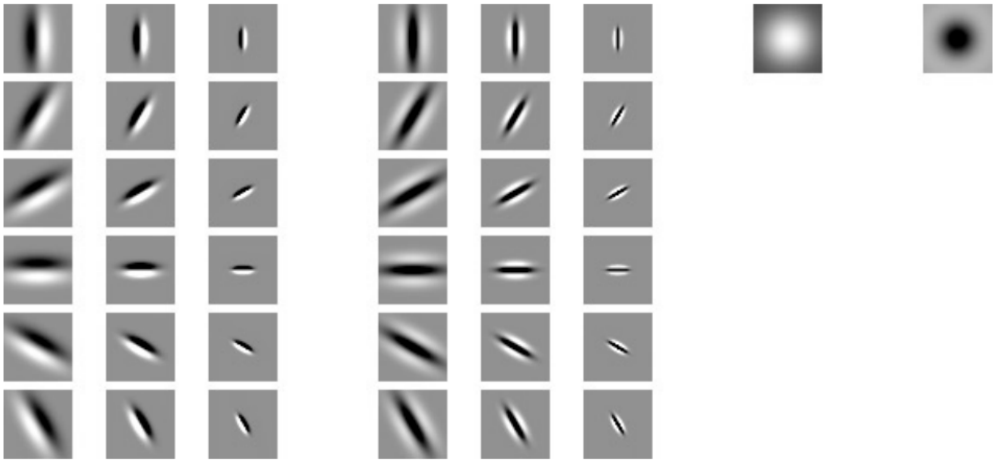


Figure 5.15

Filter bank consisting of edge and bar filters at 3 scales and 6 orientations from [17].

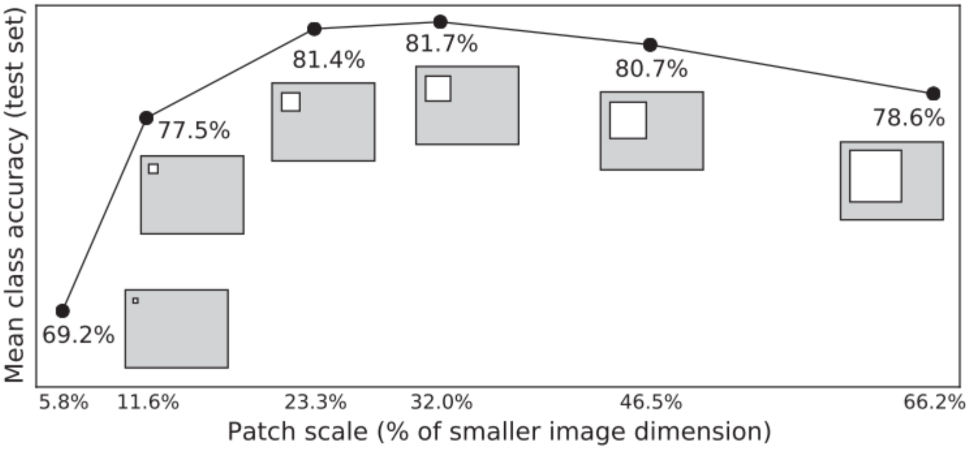


Figure 5.16

Various patch scales experimented with in [3]. It was found a tradeoff between context and resolution was the best.

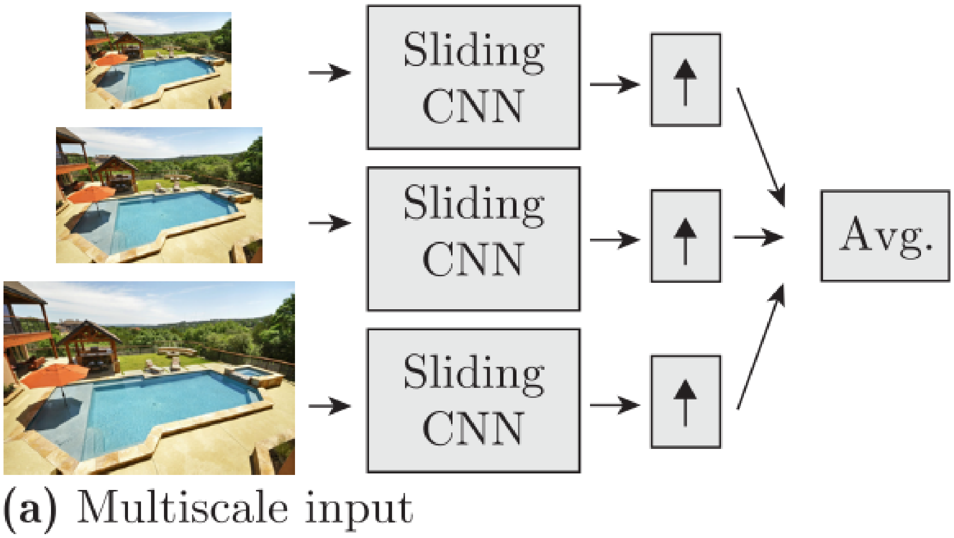


Figure 5.17

The input image to full-scene material classification from [3] at 3 scales [1, sqrt(2), 1/sqrt(2)].

may still be minor improvements to be had here regarding scale. [17] also investigates material categorization versus identification of a previous sample using decision trees with independently selected kernels and descriptors. The CNN allows for this in the decision of architectural structure of the CNN itself and with the dynamicity and variations of the filters.

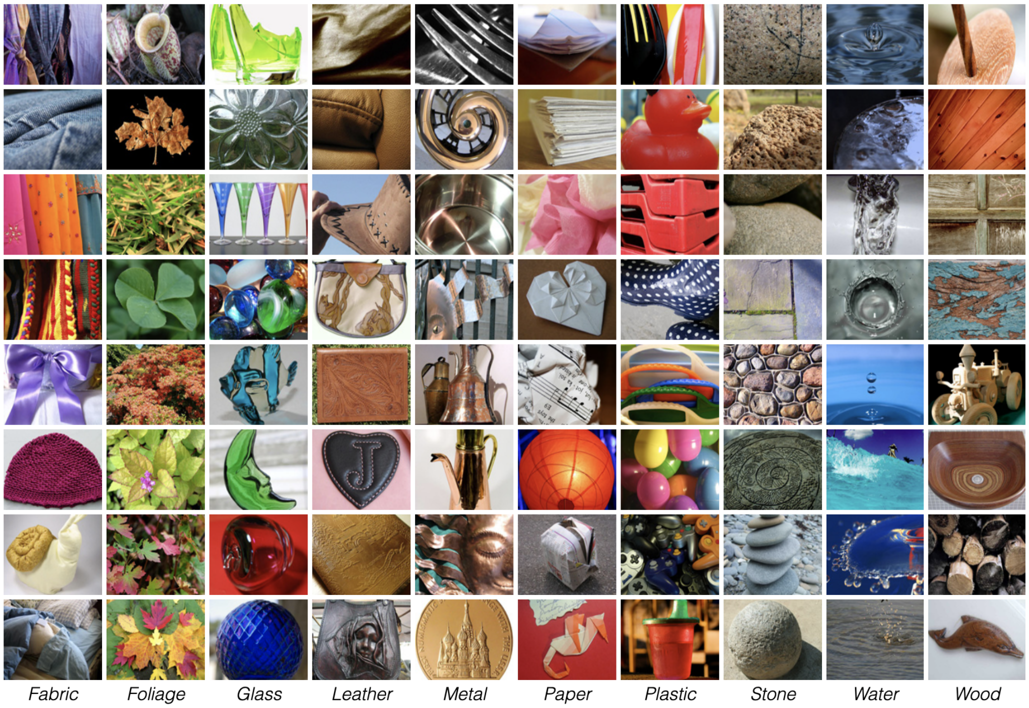


Figure 5.18

Examples from the FMD with real-world context [37].

The FMD dataset from [37] added real-world context to material samples (Figure 5.18). This dataset was constructed using images from Flickr with 100 images in 10 categories. The OpenSurfaces dataset takes this a step further by adding in material segmentations and

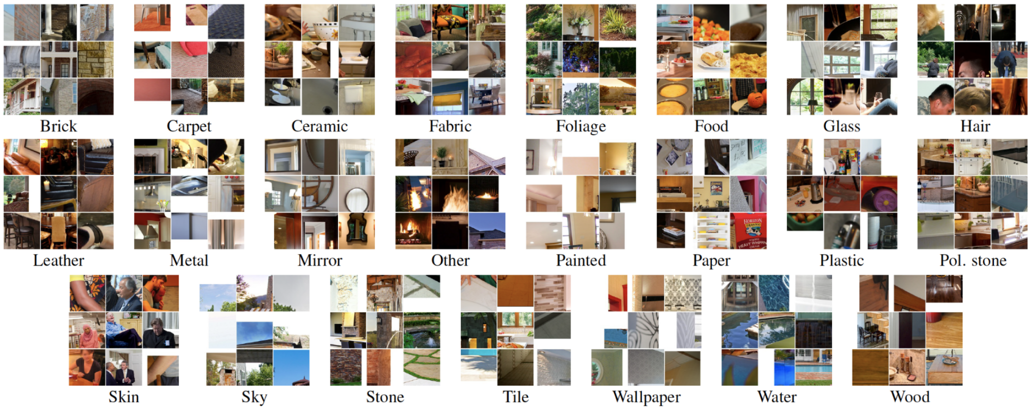


Figure 5.19

MINC database samples, much larger than FMD, also with real-world context [3].

significantly more material samples. [3] experiments with the FMD dataset and concludes that having more samples (as in MINC) and utilizing a CNN improves classification accuracy for materials versus training on FMD and utilizing an SVM. The MINC dataset is an extension of the OpenSurfaces dataset with an order of magnitude more samples and segmentations. It also contains real-world context (Figure 5.19), which as mentioned above is slightly biased in its material features since real-world images have a certain consistency to them. This could be an important factor and aid in material classification, but it could also keep the classifier from classifying materials in novel conditions correctly.

Rendered and real-world material samples are utilized in [25] to form the Virtual Texture under varying Illumination, Pose, and Scales (VIPS) database and shown to improve classification accuracy. This suggests that augmenting the MINC real-world image dataset with rendered material samples will lead to performance improvements (Figure 5.20).

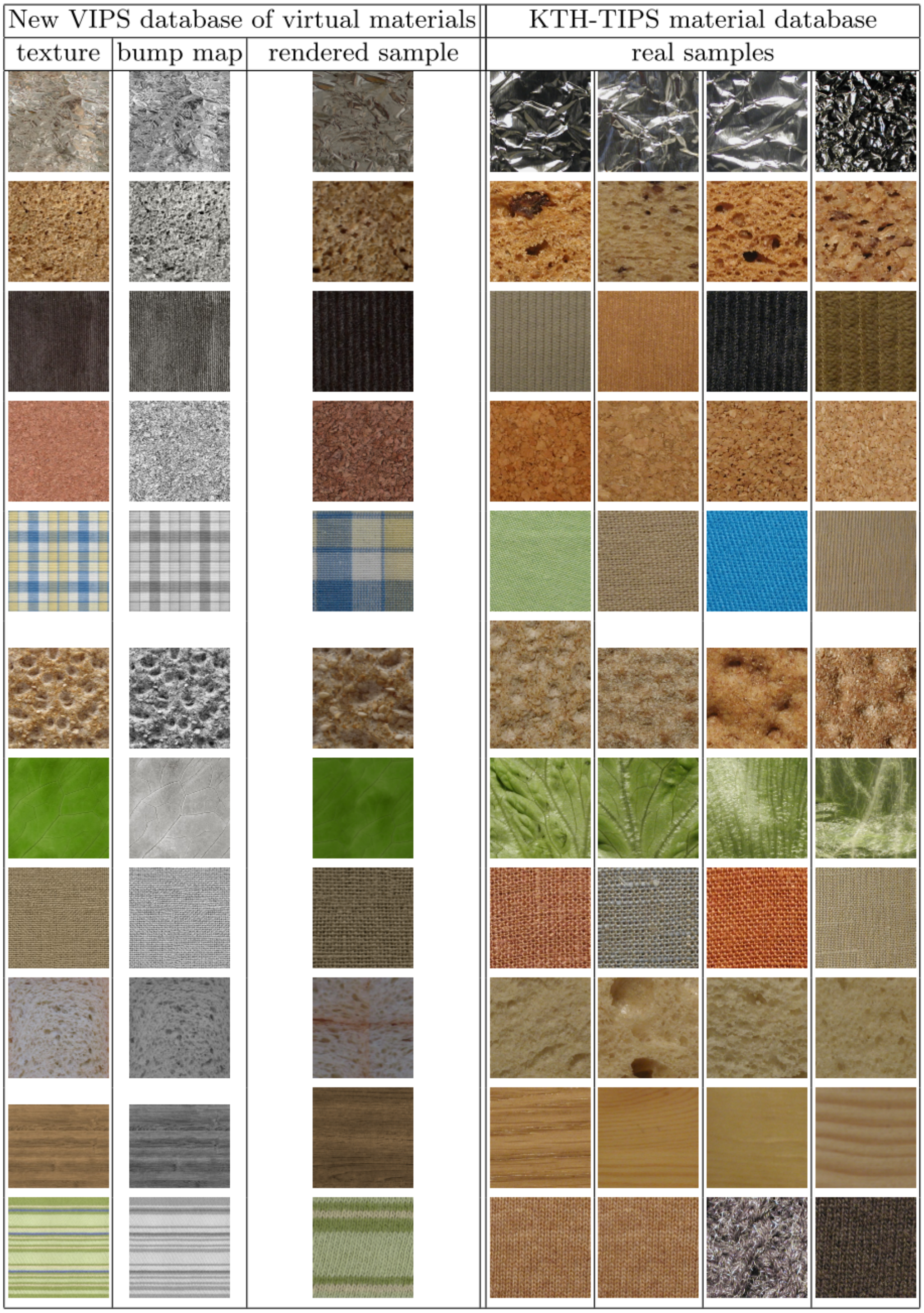

New VIPS database of virtual materials
texture
bump map
rendered sample
KTH-TIPS material database
real samples

Figure 5.20

New virtual materials with rendered examples from material shaders (left) and corresponding examples from the KTH-TIPS database (right) from [25]

5.3.2 Material Features and Recognition Techniques

The experiments from Liu et al. observed the selection of several local features including color, edge-based features, and SIFT (Figure 5.21 and Figure 5.22) finding that these three were of utmost important when classifying materials. It used an augmented LDA, which utilizes the features to complete topic modeling, to perform the classification. This work also concludes that non-local features, such as context are important for material recognition and speculates that object recognition could also be helpful [27]. The material classifier investigated in this thesis can be seen filtering colors and edge-based patterns from the first layer with its convolutional kernels (Figure 5.1). These kernels output maps combine to form more and more complex color and edge-based filters (Figure 5.23). Additionally, Sharan et al. found that an SVM completed material classification with higher accuracy than the aLDA from above [36] suggesting that machine learning models can improve classification accuracy.

Hu et al. researches features of shape, gradient orientation, and color (Figure 5.24) concluding that color is more important for material classification and that gradient orientation as well as shape are not as important. Despite the findings of lesser importance in these features, the CNN solution under observation here, utilizes shape and gradient-based features to perform material classification. Additionally, these findings suggest that performing object and material classification simultaneously could be beneficial [19]. On this note, the material classifier visualized begins to unknowingly perform object-recognition in the deeper layers (Figure 5.7), though it may be beneficial to perform targeted object and material classification simultaneously.

Table 5.4

Summary of the features and techniques observed from previous research and conclusion of potential benefit to the classifier examined in this thesis.

Name	**Technique / Advantage**	**Benefit MINC?**
local features	These features include color, edge-based features, and SIFT. Many of the works observed used these features at a minimum	No
SVM	SVMs are linear classifier that have been shown to perform well, but non-linear classifiers have received much of the focus and shown very promising results	No
aLDA [27]	aLDA is a method of topic modeling, but it was shown that an SVM outperforms it with the same training data on material classification	No
CNN	CNNs are large neural networks that can filter on very complex patterns. This thesis utilizes a CNN to perform material classification	No
Object Detection	Simultaneous object and material classification can benefit the classification accuracy as well as provide both types of output	Yes
Gradient Orientation and Magnitude [19]	These features are similar to edge-based features, but they take it a step further by having both magnitude and orientation	Possibly
Textons [45], [24]	Textons capture 3D microstructures on the surface of a material by filtering and grouping with k-means to form a vocabulary to describe the material.	No
Rotation-invariant features [31]	These features are robust to rotation and have been shown to improve classification especially if utilizing both rendered and real-world data	Possibly
oBIFS and Pooling [43]	Oriented features in a training free classification system with pooling offer a static method for doing material classification with high accuracy	Possibly
Material Traits [34], [5]	These describe materials in a manner that is difficult for a classifier to distinguish such as soft, rough, studded, smeared, etc.	Yes

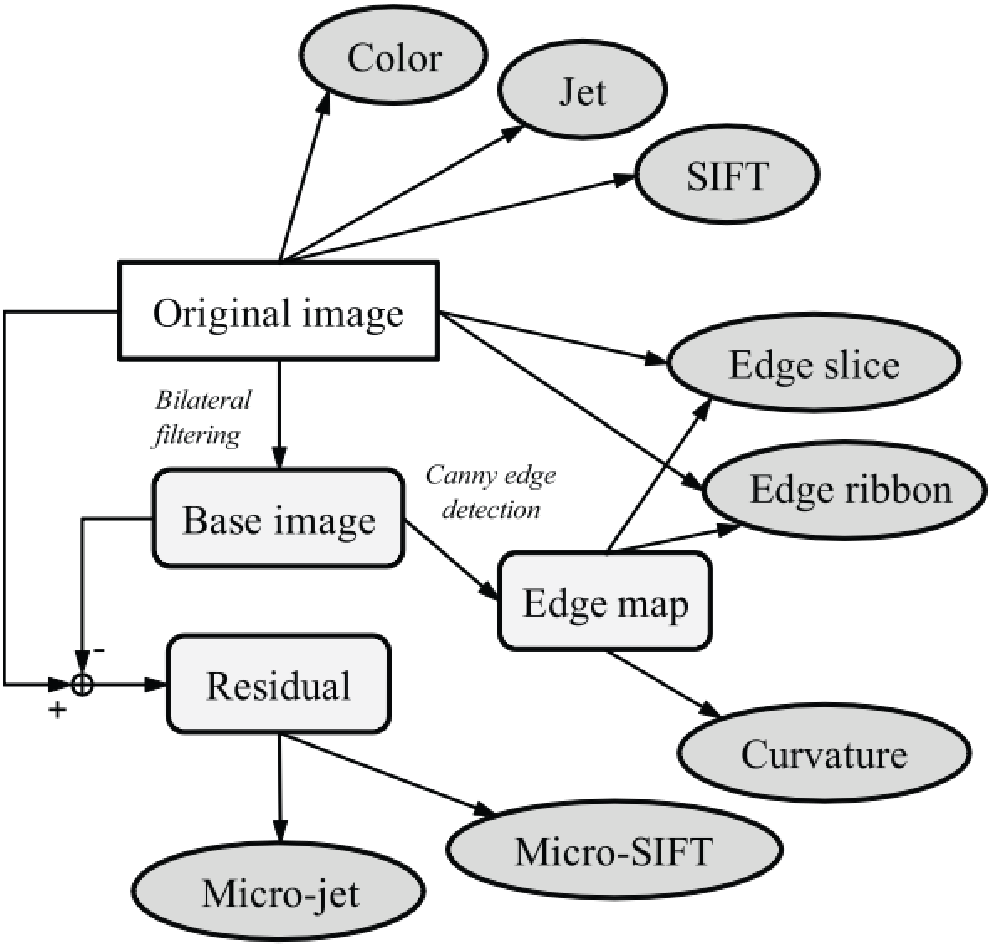


Figure 5.21

Illustration of how the system from Liu et al. [27] generates features.

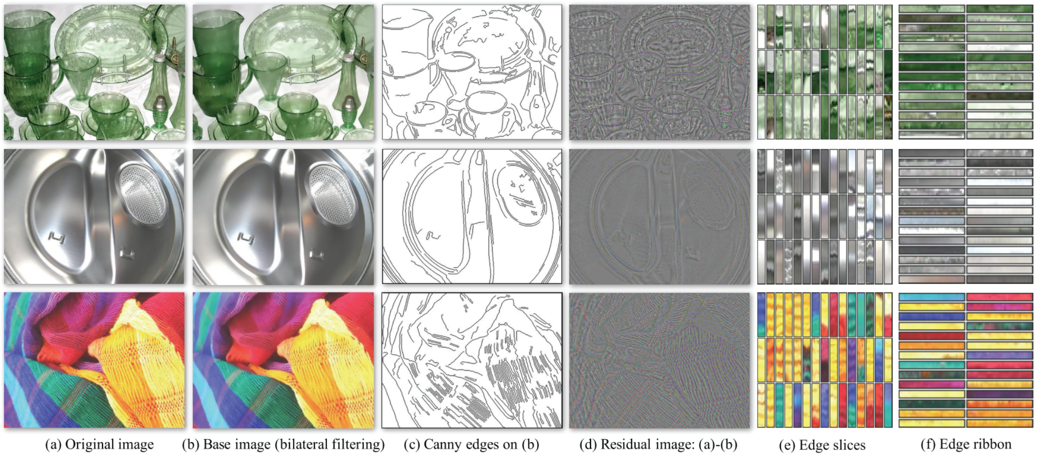


Figure 5.22

Features from Liu et al. [27] for material recognition.

Leung et al. creates a vocabulary of 3D textons that capture microstructures on the surface of materials by utilizing the CUReT database. Since pose and illumination can vary, these 3D textons and specific combinations of them are used to distinguish materials [24]. The MINC dataset in combination with the CNN combine to filter on very similar features, especially in the early layers. The filter bank used to help create the textons (Figure 5.25) and the filters used in the CNN from this thesis (Figure 5.1) are very similar. Considering that the results from these filters are passed through rectified linear unit processing and pooled in the CNN, the process to create and use the textons [24] (Figure 5.26) is almost exactly replicated in the CNN investigated here. We can see some of these 3-dimensional microstructures being filtered out early in the CNN in Figure 5.27.

Similar to the above, Varma et al. creates a texton vocabulary using filter banks, but utilizes them in a distinct manner. Instead of looking for an arrangement of these textons

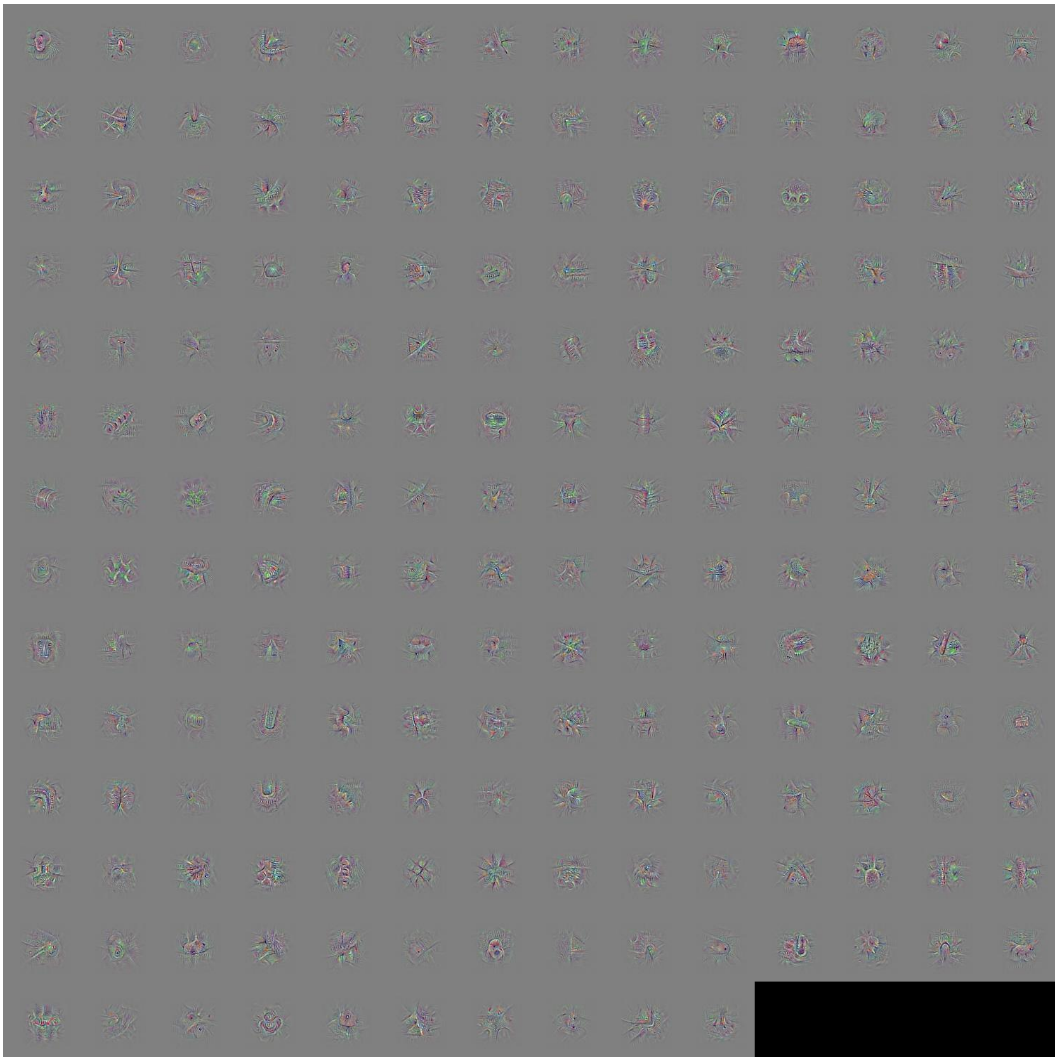


Figure 5.23

Filters of the 3b 5x5 layer of the network used in this work.

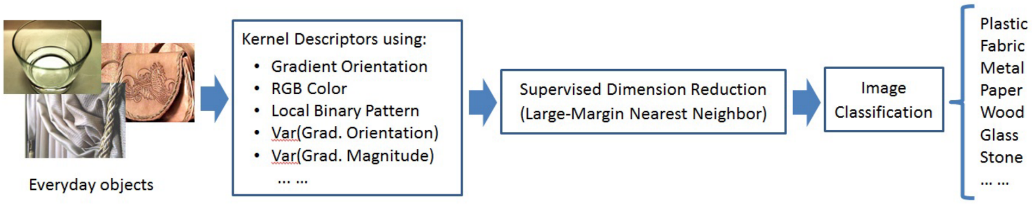


Figure 5.24

Overview of the approach from [19].

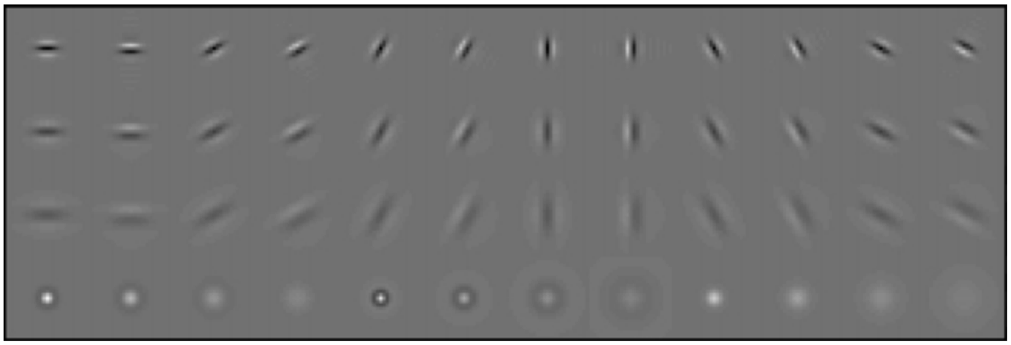


Figure 5.25

The filter bank used in [24]'s analysis. Total of 48 filters: 36 oriented filters, with 6 orientations, 3 scales, and 2 phases, 8 center-surround derivative filters and 4 low-pass Gaussian filters.

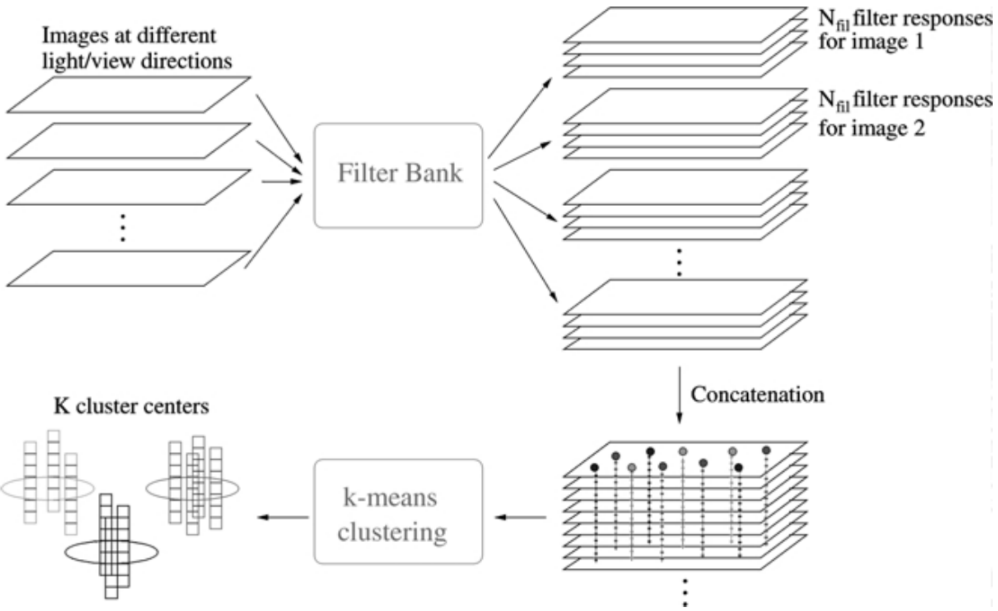
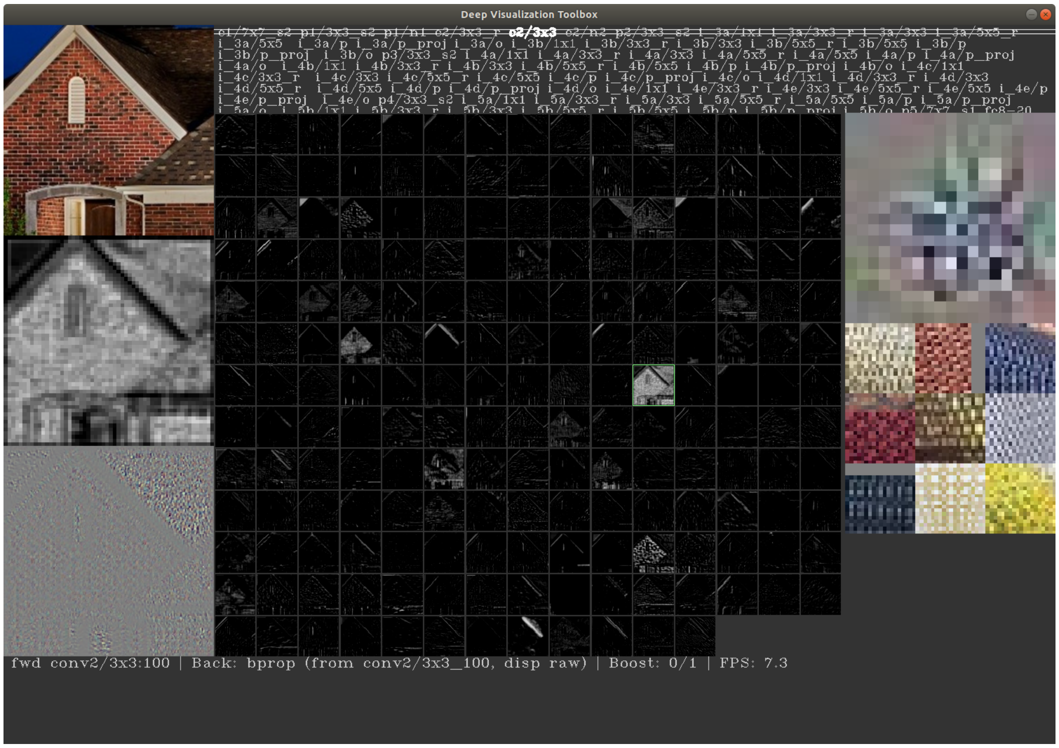

Images at different light/view directions
Filter Bank
N_{fil} filter responses for image 1
N_{fil} filter responses for image 2
Concatenation
K cluster centers
k-means clustering

Figure 5.26

Each image with various light/view directions is filtered using the filter bank from Figure 5.25. The response vectors are concatenated together to form data vectors. These data vectors are clustered using the K-means algorithm. The resulting centers are the 3D textons.



```
c1/7x7_s2 p1/3x3_s2 p1/n1 c2/3x3_r c2/3x3 c2/n2 p2/3x3_s2 i_3a/1x1 i_3a/3x3_r i_3a/3x3 i_3a/5x5_r
i_3a/5x5  i_3a/p i_3a/p_proj i_3a/o i_3b/1x1 i_3b/3x3_r i_3b/3x3 i_3b/5x5_r i_3b/5x5 i_3b/p
i_3b/p_proj  i_3b/o p3/3x3_s2 i_4a/1x1 i_4a/3x3_r i_4a/3x3 i_4a/5x5_r i_4a/5x5 i_4a/p i_4a/p_proj
i_4a/o  i_4b/1x1 i_4b/3x3_r i_4b/3x3 i_4b/5x5_r i_4b/5x5 i_4b/p i_4b/p_proj i_4b/o i_4c/1x1
i_4c/3x3_r  i_4c/3x3 i_4c/5x5_r i_4c/5x5 i_4c/p i_4c/p_proj i_4c/o i_4d/1x1 i_4d/3x3_r i_4d/3x3
i_4d/5x5_r  i_4d/5x5 i_4d/p i_4d/p_proj i_4d/o i_4e/1x1 i_4e/3x3_r i_4e/3x3 i_4e/5x5_r i_4e/5x5 i_4e/p
i_4e/p_proj  i_4e/o p4/3x3_s2 i_5a/1x1 i_5a/3x3_r i_5a/3x3 i_5a/5x5_r i_5a/5x5 i_5a/p i_5a/p_proj
i_5a/o  i_5b/1x1 i_5b/3x3_r i_5b/3x3 i_5b/5x5_r i_5b/5x5 i_5b/p i_5b/p_proj i_5b/o p5/7x7_s1 fc8=20
```

```
fwd conv2/3x3:100 | Back: bprop (from conv2/3x3_100, disp raw) | Boost: 0/1 | FPS: 7.3
```

Figure 5.27

The 3 dimensional microstructure seen in the right panels can be seen. This filters a repeated knit or brick pattern from the image which can be very helpful for material classification.

within a material sample, they create a frequency histogram of the textons and use that to classify the sample [45] (Figure 5.28). As with the work from above, the CNN explored in this thesis accomplishes this with early layer filtering and feature dissemination to deeper layers.

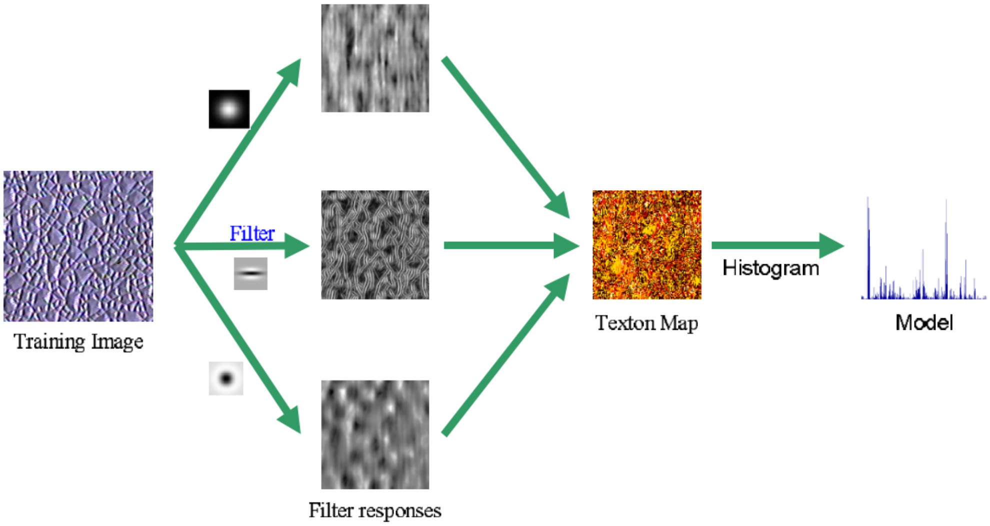


Figure 5.28

Given a training image, a model is generated by first convolving it with a filter bank. After this, each filter response is labelled with the texton which is closest to it in filter response space. The frequency histogram for each texton in the labelling creates this model. This is the process used in [45].

Rotation-invariant, pairwise local binary pattern (LBP) features are utilized in Qi et al. These features are robust to rotation and help to detect certain co-occurrences in the input [31] (Figure 5.29). This is a technique that proved useful in other recognition tasks and could be helpful in material recognition. There are a couple of methods to introduce features of this type to a neural network classifier. One technique is to augment the training dataset with translated and transposed input images. Another technique may be to perform more robust classification by classifying an image several times after translating and transposing it. These both require computational overhead and are being completed by the original training process of the CNN observed in this thesis [3], so a final technique for doing this is to augment the features in the image with rotation-invariant features such as the pairwise LBP from [31].

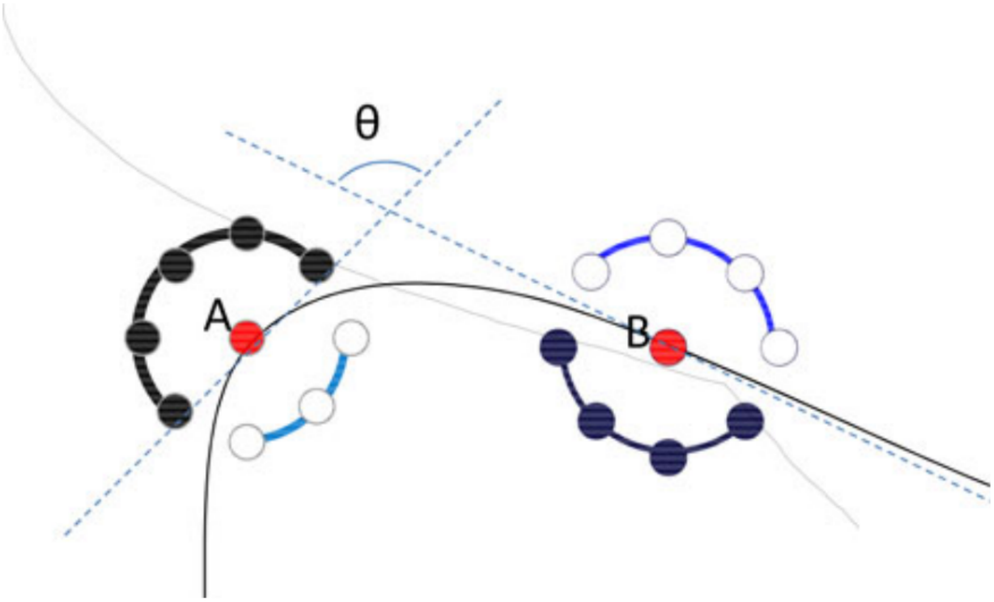


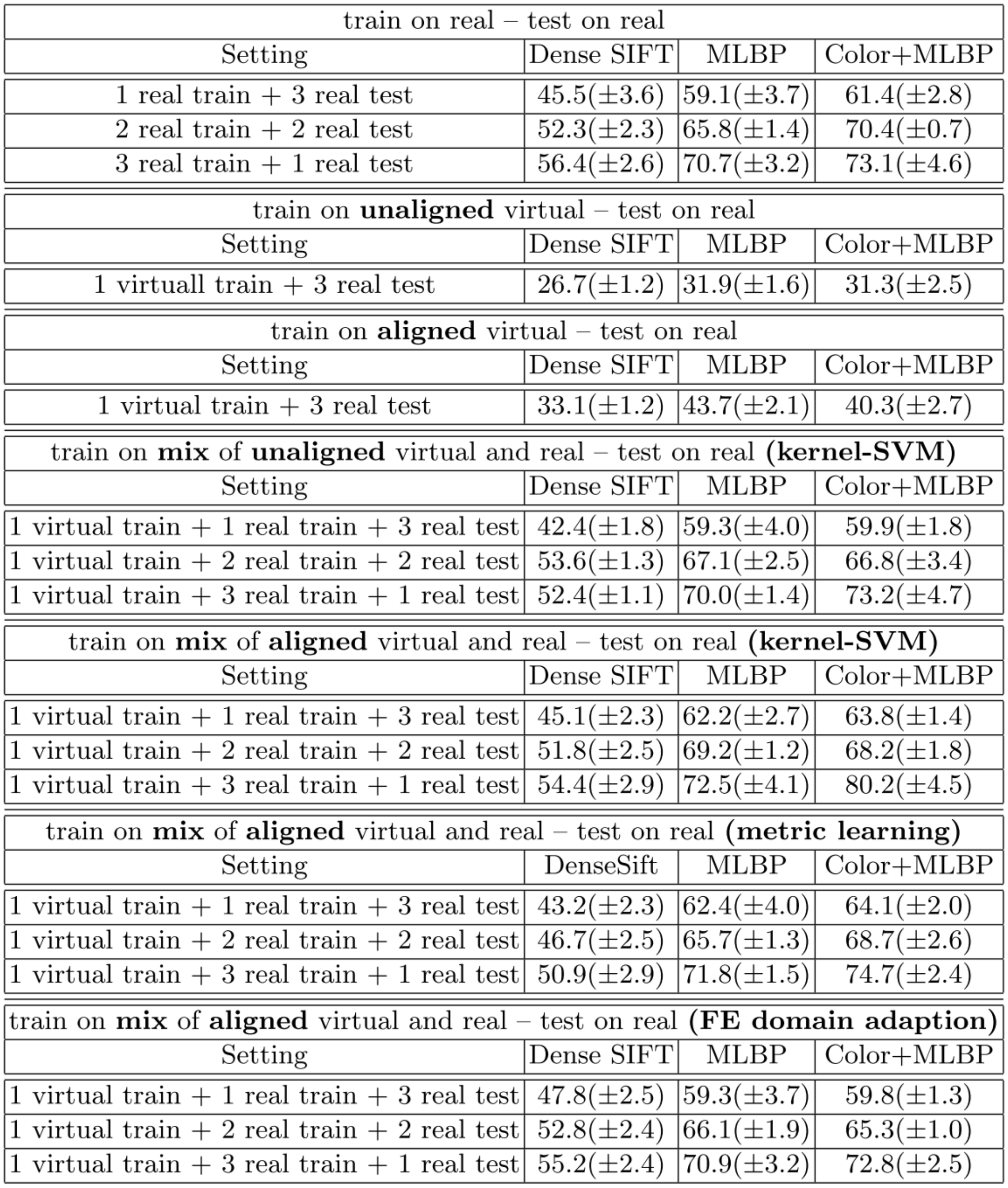
Figure 5.29

Illustration of co-occurrence local binary pattern from [31].

Li et al. explores the use of rendered material samples to augment real-world samples and which features work well when using this design. It finds that LBP-based features tend to apply to both rendered and real-world samples better than SIFT and that utilizing these features for both types of samples results in higher classification accuracy [25] (Figure 5.30). This suggests an area for improvement for the CNN-based method in that training on a dataset containing both of these types of samples could lead to accuracy improvements and certain features informational contributions changing.

Timofte et al. takes a creative approach to texture and material classification by experimenting with a training free classification system (Figure 5.31). It uses Oriented Basic Image Features (oBIFs) for pixel level descriptions. Additionally, it also conducts pooling with spatial pyramid matching and bag-of-regions which adds variance in scale, translation, rotation, viewpoint, and illumination to the classification system. This does increase the computation needed to perform classification significantly (by an order equal to the total regions). These two pooling techniques allow for multi-level image representation. This representation is then used to perform classification with sparse and collaborative methods that do not require parameter tuning such as neural network weights. This method seems to offer up an abundant amount of information about performing classification in a robust and fixed manner with fixed features. The BIFs that the oBIFs are created from resemble convolutional filters very closely (Figure 5.32 and Figure 5.33). These are then used in a repeated manner with a pooling and robustification technique [43]. The pooling is similar to how a neural network might perform pooling, and the robustification is accomplished by supplementing the training data for the CNN with perturbations, scales, and shifts.

Another technique that is shown to perform well on multiple datasets is that of Schwatrz et al. It uses material traits from object-independent, per pixel material information to classify and segment materials. To determine these traits through unsupervised fea-



train on real – test on real			
Setting	Dense SIFT	MLBP	Color+MLBP
1 real train + 3 real test	45.5(±3.6)	59.1(±3.7)	61.4(±2.8)
2 real train + 2 real test	52.3(±2.3)	65.8(±1.4)	70.4(±0.7)
3 real train + 1 real test	56.4(±2.6)	70.7(±3.2)	73.1(±4.6)

train on **unaligned** virtual – test on real			
Setting	Dense SIFT	MLBP	Color+MLBP
1 virtuall train + 3 real test	26.7(±1.2)	31.9(±1.6)	31.3(±2.5)

train on **aligned** virtual – test on real			
Setting	Dense SIFT	MLBP	Color+MLBP
1 virtual train + 3 real test	33.1(±1.2)	43.7(±2.1)	40.3(±2.7)

train on **mix** of **unaligned** virtual and real – test on real **(kernel-SVM)**			
Setting	Dense SIFT	MLBP	Color+MLBP
1 virtual train + 1 real train + 3 real test	42.4(±1.8)	59.3(±4.0)	59.9(±1.8)
1 virtual train + 2 real train + 2 real test	53.6(±1.3)	67.1(±2.5)	66.8(±3.4)
1 virtual train + 3 real train + 1 real test	52.4(±1.1)	70.0(±1.4)	73.2(±4.7)

train on **mix** of **aligned** virtual and real – test on real **(kernel-SVM)**			
Setting	Dense SIFT	MLBP	Color+MLBP
1 virtual train + 1 real train + 3 real test	45.1(±2.3)	62.2(±2.7)	63.8(±1.4)
1 virtual train + 2 real train + 2 real test	51.8(±2.5)	69.2(±1.2)	68.2(±1.8)
1 virtual train + 3 real train + 1 real test	54.4(±2.9)	72.5(±4.1)	80.2(±4.5)

train on **mix** of **aligned** virtual and real – test on real **(metric learning)**			
Setting	DenseSift	MLBP	Color+MLBP
1 virtual train + 1 real train + 3 real test	43.2(±2.3)	62.4(±4.0)	64.1(±2.0)
1 virtual train + 2 real train + 2 real test	46.7(±2.5)	65.7(±1.3)	68.7(±2.6)
1 virtual train + 3 real train + 1 real test	50.9(±2.9)	71.8(±1.5)	74.7(±2.4)

train on **mix** of **aligned** virtual and real – test on real **(FE domain adaption)**			
Setting	Dense SIFT	MLBP	Color+MLBP
1 virtual train + 1 real train + 3 real test	47.8(±2.5)	59.3(±3.7)	59.8(±1.3)
1 virtual train + 2 real train + 2 real test	52.8(±2.4)	66.1(±1.9)	65.3(±1.0)
1 virtual train + 3 real train + 1 real test	55.2(±2.4)	70.9(±3.2)	72.8(±2.5)

Figure 5.30

Results on KTH TIPS and the VIPS database from [25].

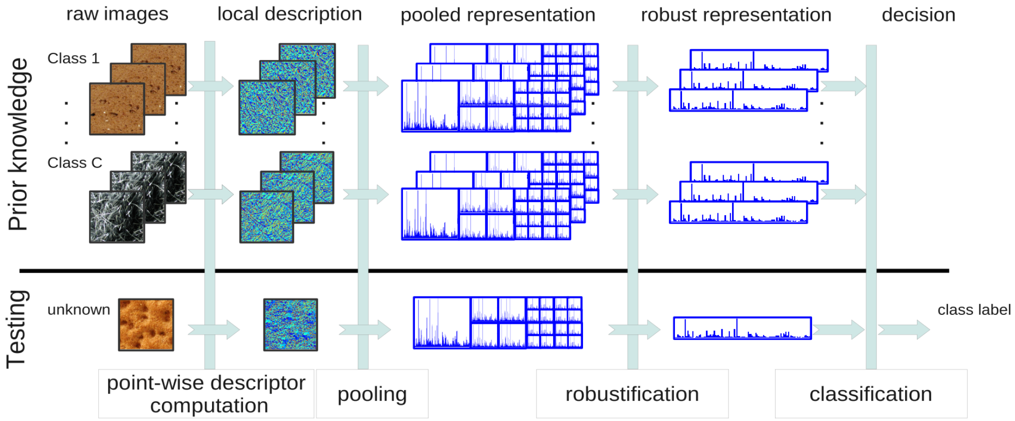


Figure 5.31

Scheme of the texture classification framework from [43].

ture learning using convolutional filters (Figure 5.34). It also supplements these features with Local Binary Pattern features since [7] suggests that non-continuous features such as these cannot be learned. It then captures the distribution of material traits across different materials to perform classification [34] (Figure 5.35). This research suggests that adding additional features, such as LBP, that may not be able to be represented by artificial neural networks, can improve classification accuracy. The CNN utilized in this thesis for material classification performs its own material trait recognition by combining filters (especially those learning 3-dimensional microstructures) throughout and by learning which of these complex filters apply to which materials. However, it seems that CNN's are much better at shape and color filtering than material trait filtering, so adding these material trait labels to data could be helpful in determining the material class.

Cimpoi et al. [5] showed that texture attributes (Figure 5.36), which are very similar to the material traits from above can be used to perform material classification. These

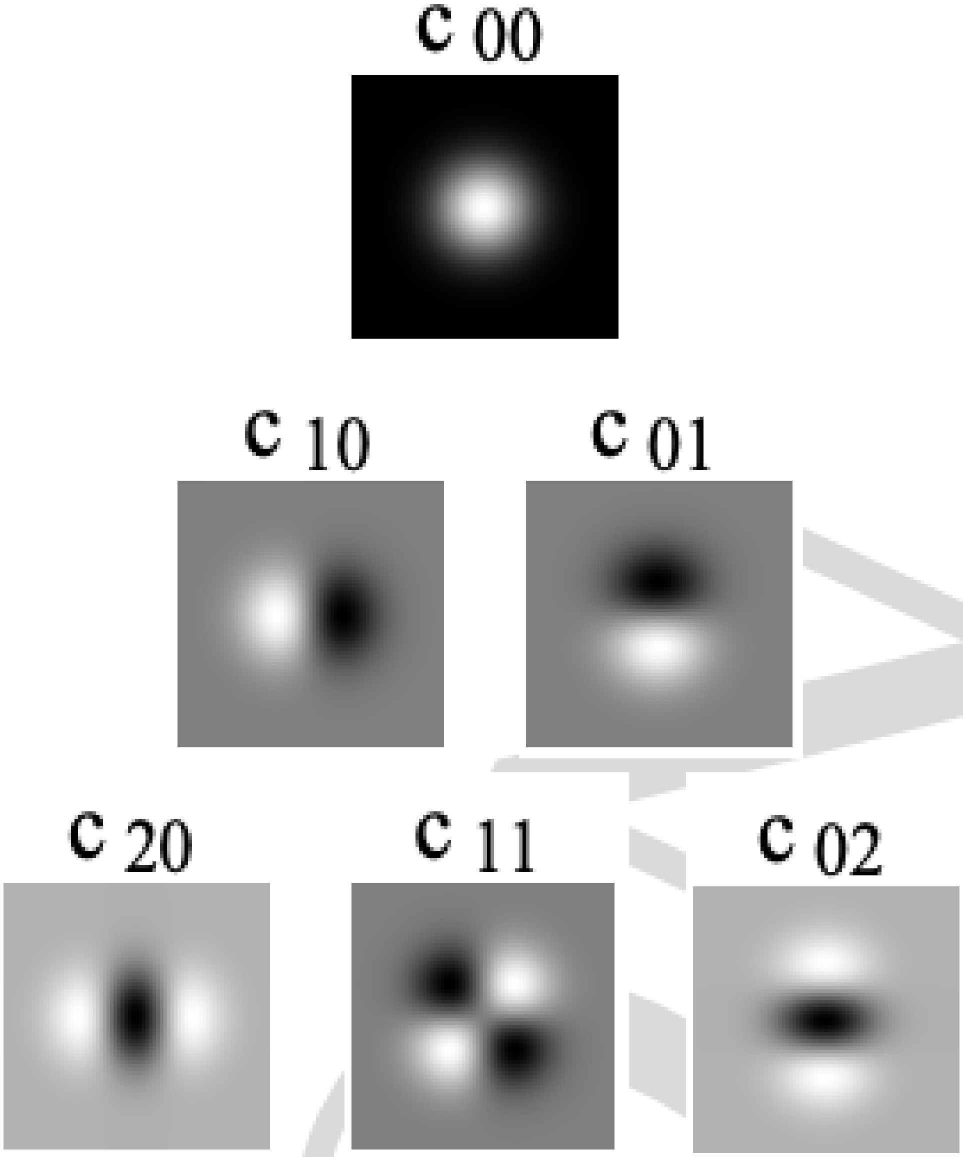


Figure 5.32

Filters used in creation of Basic Image Features (BIFs) [6].

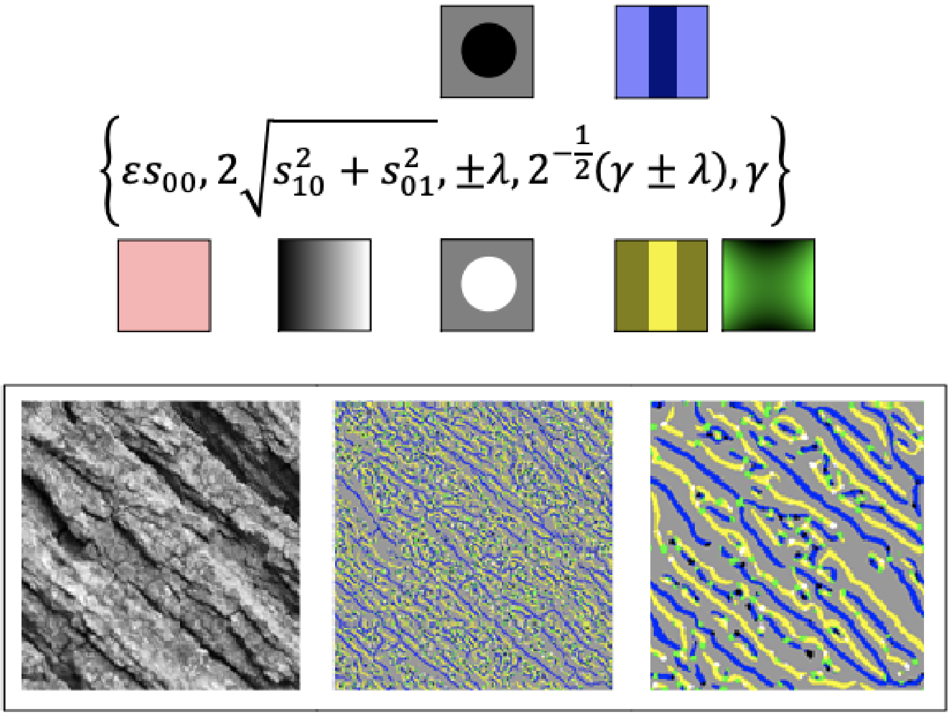

$\left\{\varepsilon s_{00}, 2\sqrt{s_{10}^2 + s_{01}^2}, \pm\lambda, 2^{-\frac{1}{2}}(\gamma \pm \lambda), \gamma\right\}$

Figure 5.33

Example of the use of BIFs to create features for use in classification tasks [6].

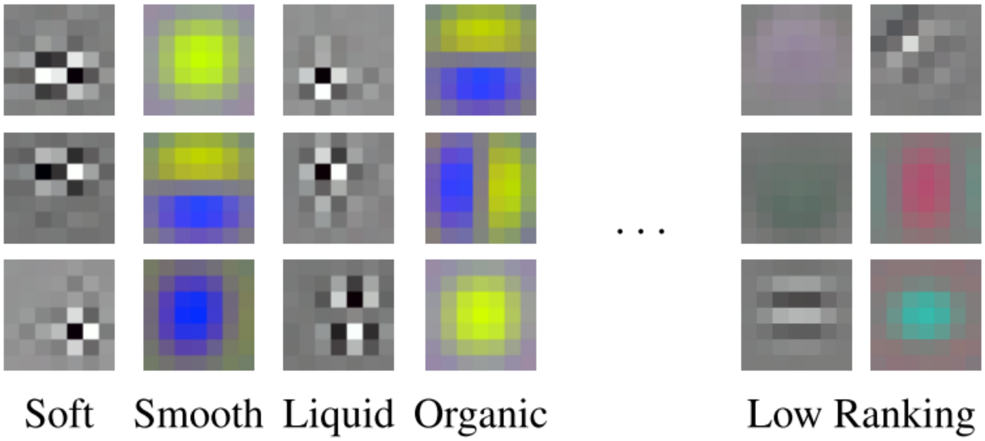


Figure 5.34

7 x 7 filters learned by the convolutional auto-encoder from [34].

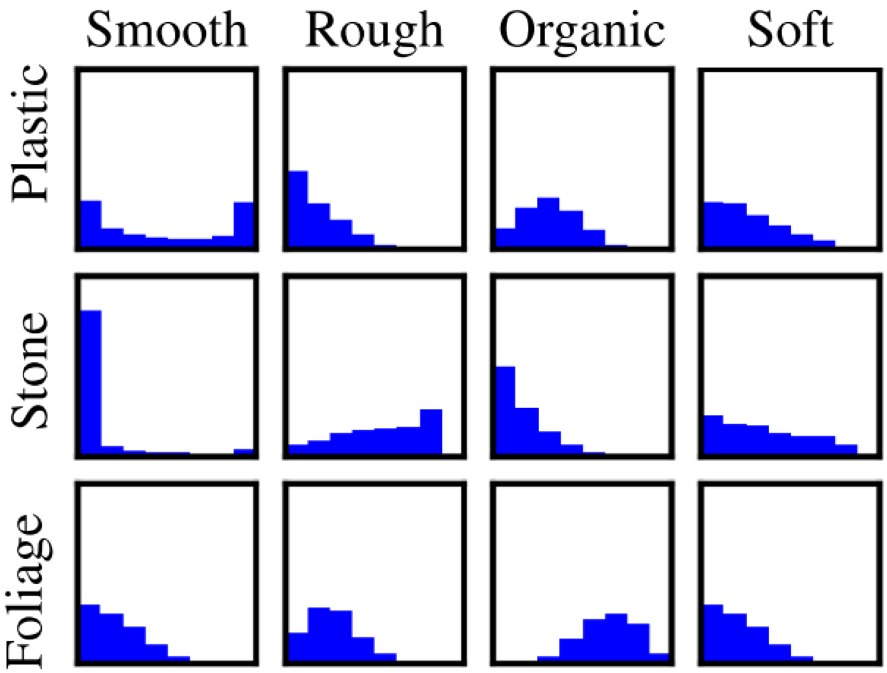


Figure 5.35

Material trait distributions for each material class from [34].

attributes are created in the form of improved fisher vectors (IFVs) which have been used for object-recognition. IFVs extract local SIFT features and normalize them. Bell et al. [3] experimented with and compared IFV usage against the CNN investigated here and found the CNN outperformed it. However, a collaborative method combining both IFVs and CNN technique from [3] could lead to improved accuracy since the IFVs have been shown as promising features in the past.

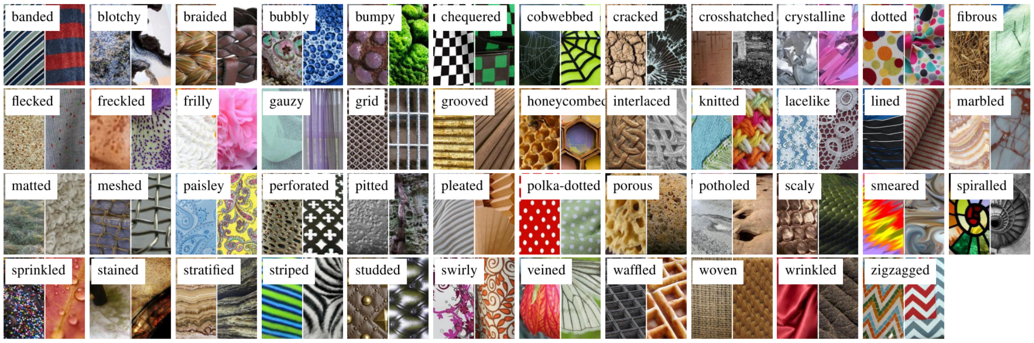
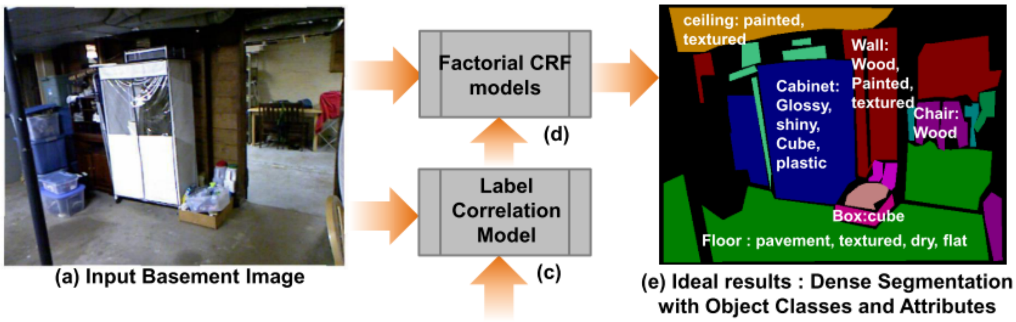


Figure 5.36

Examples of texture attributes from [5].

Object and material classification are jointly experimented with in Zheng et al. [49]. The findings conclude that utilizing object and material information when classifying and segmenting an image (Figure 5.37) can be helpful. The CNN that this thesis utilizes begins to perform object recognition ignorantly at deeper layers, but performing both simultaneously and purposefully could yield positive results, especially in a system that has the need to complete both of these classifications.


(a) Input Basement Image
Factorial CRF models
(d)
Label Correlation Model
(c)
ceiling: painted, textured
Wall: Wood, Painted, textured
Cabinet: Glossy, shiny, Cube, plastic
Chair: Wood
Box:cube
Floor : pavement, textured, dry, flat
(e) Ideal results : Dense Segmentation with Object Classes and Attributes

	Objects	Attributes
Region-Level	Things(e.g. Chair, TV monitor, Box, etc.)	Shape (e.g. Cube) , Material, Surface ...
Pixel-Level	Things and stuff (e.g. Chair, TV monitor, Box, Wall, ceiling, floor, etc.)	Material (e.g. plastic, wood), Surface (e.g. glossy), ...

(b) Semantic Space

Figure 5.37

(a) shows the input image, a scene image from NYU dataset. (b) represents the semantic label space with pixel- and region-levl objects and attributes. (e) shows conceptual results for segmentation with objects and at- tributes [49].

5.3.3 Convolutional Neural Networks

The work by Krizhevsky et al. [23] shows that deep neural networks can perform visual recognition tasks with state-of-the-art accuracy. It shows that the specific depth of five convolutional layers (Figure 2.6) is necessary to maintain the level of accuracy obtained. The paper from which the material classifier utilized by this thesis is taken [3] experiments with AlexNet (the neural network from [23]) for comparison. It finds that AlexNet performs comparatively to GoogLeNet with slightly lower accuracy on material recognition tasks. The filters used in both are very similar (Figure 5.38).

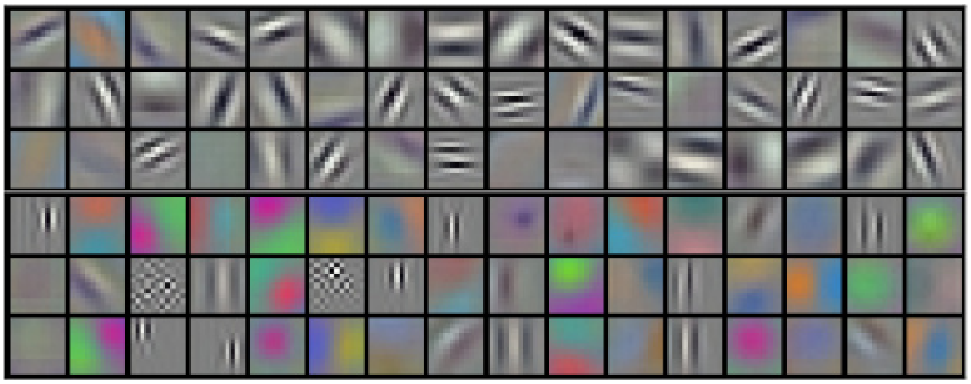


Figure 5.38

Filters learned by the first layer of the CNN from Krizhevsky et al. [23].

Additionally, Szegedy et al. [41] asserts the GoogLeNet neural network architecture (Figure **??**) that is an efficient twenty-two layer network that outperforms previous state-of-the-art networks on visual recognition tasks. This is the network experimented with here,

Table 5.5

Summary of previous research relating to CNNs and the potential for improvement to the CNN observed by this thesis.

Name	**Details**	**Benefit MINC?**
AlexNet [23]	One of the first CNNs shown to excel at visual tasks. It was looked at in previous work [3] and performed similarly to the CNN observed in this thesis	No
GoogLeNet [41]	Builds upon AlexNet to form a larger (but also still efficient) CNN. GoogLeNet is utilized in this work, but the technique utilized in Szegedy et al. could aid in future material classification applications	Possibly
Region Proposals [13]	Region proposals suggest an area in the visual representation that could represent a classification suggesting localization and segmentation	Yes
Simultaneous Classification and Localization [35]	Bounding box convergence is used to complete this task both locating the class and classifying it	Possibly
Transfer learning	Training a network on a similar visual task, then transferring the weights learned during this training for a subset of layers to a new task	No

but the work by [41] suggests that further hand-crafting of the network could be beneficial with the addition, and possibly subtraction, of layers from the network while maintaining efficiency. This could be accomplished through approximating the expected optimal sparse structure by readily available dense building blocks.

The work by Girshick et al. [13] portrays the advantage of neural network feature extraction and that region proposals (Figure 5.40) can aid in object detection. Region proposals could be utilized with material classification as well to improve the classification accuracy by suggesting segments and localizations of those materials.

Sermanet et al. [35] simultaneously completes classification and localization tasks and shows advantages of doing so. It uses bounding box convergence to perform this (Figure 5.41). The material classifier utilized in [3], and replicated here, utilizes a CRF, after a sliding-window, fully-convolutional neural network runs classification across an image, to complete localization in a more detailed manner (Figure 5.9).

Farabet et al. [12] shows how a CRF can be utilized for material segmentation (Figure 5.42). [3] shows an improvement upon this with the dense CRF from [22], so this technique is already completed.

Lastly, Oquab et al. [30] is able to conclude the benefit of transfer learning to similar tasks (Figure 5.43) which the work here is doing with the ImageNet dataset similar to [30].

5.4 Recommendations

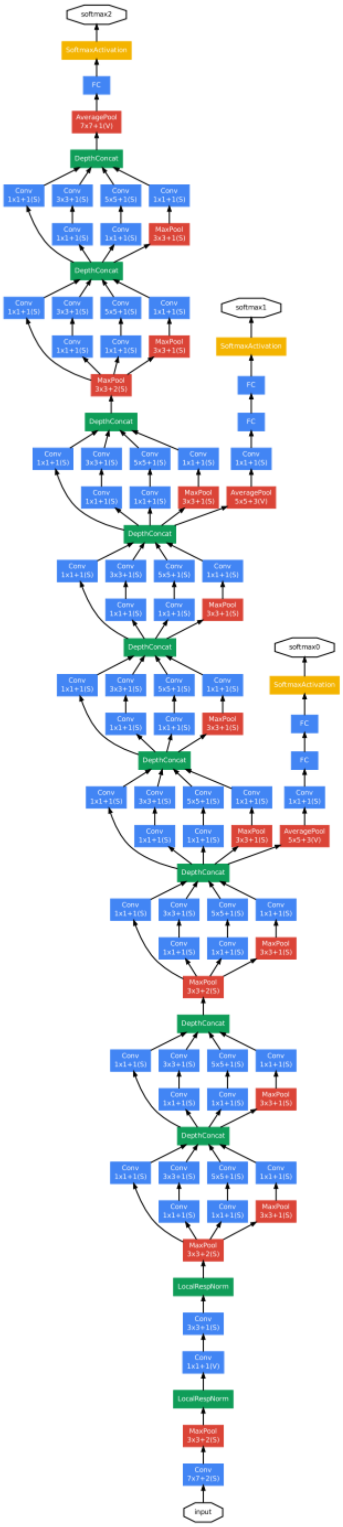


Figure 5.39

GoogLeNet architecture from [41].

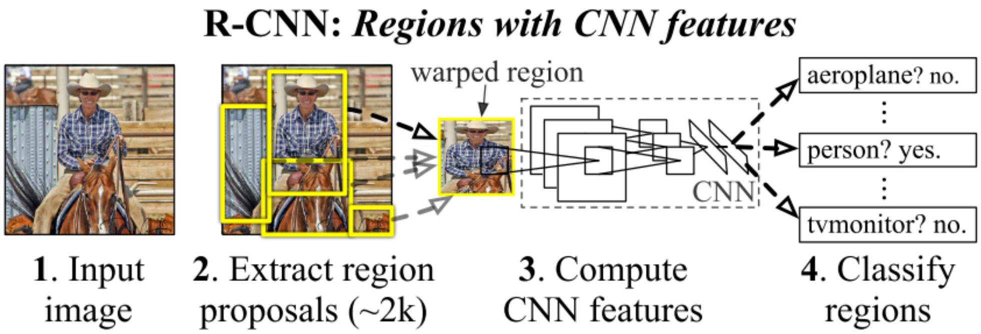


Figure 5.40

object detection system overview from [13].

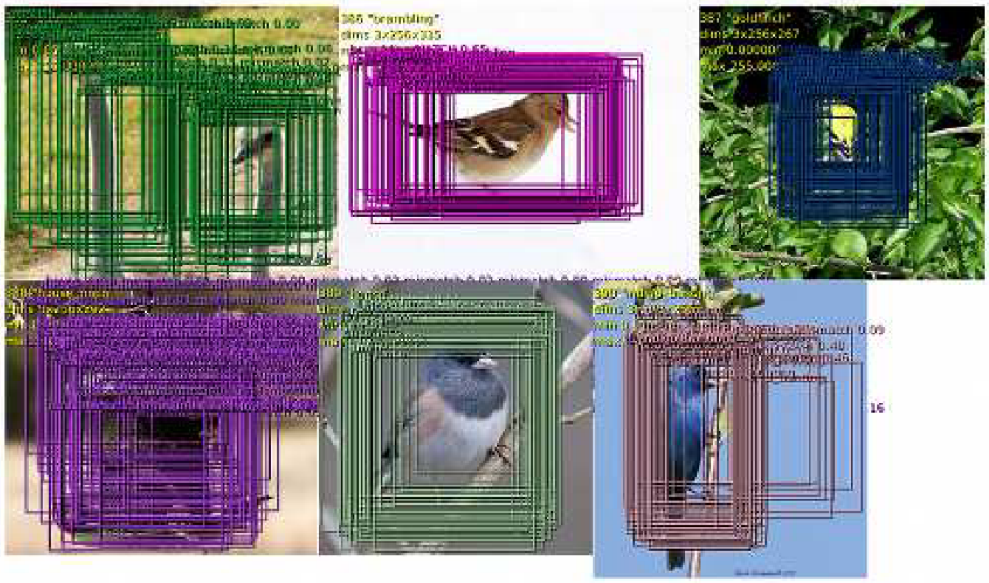


Figure 5.41

Examples of the bounding boxes produced by the work from [35].

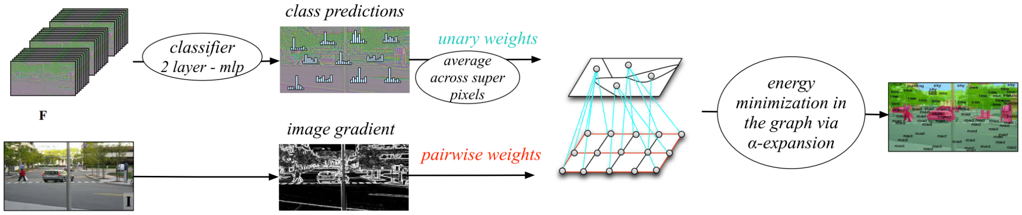


Figure 5.42

Labelling strategy using a CRF from [12].

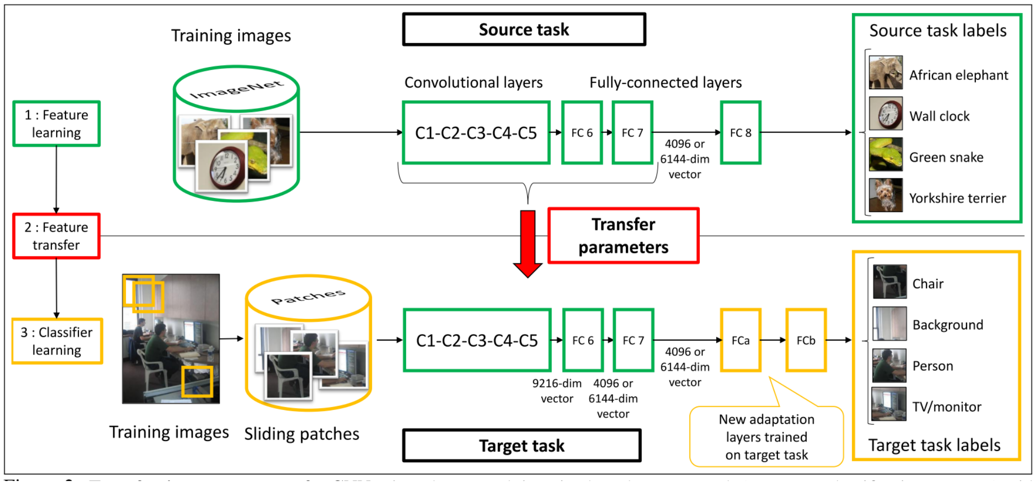


Figure 5.43

Transferring properties of a CNN. First a neural network is trained for a source task. Parameters are then transferred from this trained network to a target task. Lastly, additional layers are trained just for the target task. This technique comes from [30].

Firstly, for the dataset, the experimentation of more variation in lighting and viewing angles is recommended as it could add robustness and flexibiliy to the classification. Additionally, scale could be altered more than currently being completed for the same purposes. Rendered material samples may be necessary for these but are also the final recommendation for the dataset for material classificaiton as rendered samples have been shown to improve classification accuracy.

As for the features and recognition, purposeful and targeted object and material classification, especially when the application of the network requires it is a recommendation since it could improve the overall accuracy of both tasks. Material and texture traits, possibly using IFVs could be supplemental to the CNN's current features and allow for more accuracy in classification of materials. Lastly, rotation-invariant features seem to perform well when the dataset includes both real-world and rendered samples. This could be a combinatorial approach to improving the material classifier.

The CNN itslef could be supplemented with some of the above features, but the architecture itself could be manipulated to be more efficient or accurate by adding or subtracting layers while maintaining efficiency similar to the method from [41]. In addition to this, the CNN features could be supplemented with region proposals suggesting areas for specific materials. This could help with both classification and segmentation.

One last consideration involves the reasons that this CNN may be completing material classification better than with previous datasets, techniques, features, and CNNs. It could be due to its ability to do something that previous research has not. This is the novel ability of this CNN above others to perform material classification. It is difficult to spot this novelty, since one does not know exactly what to look for in this respect. This CNN is learning on a particular architecture with a very large and particular dataset that enables it to do well in this area. Due to it being a CNN, it also has a capability to automatically generate these filters that allow it to extract features and make decisions on the dataset. This thesis shows that the CNN extracts low-level simple features first, and moves toward more complex structure and colors in deeper layers. It is very interesting to observe exactly what it has learned. As such, all of the visualizations from this thesis are available for other interprations (https://github.com/jdonovancs/dvtb/).

CHAPTER 6

DISCUSSION

The results here show the strengths and weaknesses of the CNN at the material classification task and provides evidence of the importance of material classification and recommendations for improving the state-of-the-art. There are several methods that have not been discussed or experimented with for material classification that could prove interesting. One of these is evolutionary techniques for the composition of the neural network. With growing computational power and parallelization of algorithms, evolutionary algorithms have become more widespread. Future research could also focus on real-time material classification as that is a growing area of interest for robotics and decision-making systems. Real-time object classification has been experimented with some, but material classification has lacked the same attention. Another area of future research could involve broadening the material datasets (MINC and others). The implications of this future research could yield more generic material classifiers that can span larger and more diverse domain problems. The visualization tools utilized here could be applied to other domains where neural networks are used for visual analysis and possibly even domains not directly related to visual tasks.

CHAPTER 7

CONCLUSION

This thesis has provided three main contributions to the research areas of neural network visualization and material classification. The first of these contributions is the replication and analysis of the state-of-the-art material classification algorithm from [3]. This replication serves as a baseline on which future research can be performed and compared. It also solidifies the research of [3] and provides further implementation of the technique to the public. The analysis of this network provides information on the functionality of the CNN and the features that it has learned throughout its training. This provides a vast amount of knowledge on why it is as successful as it is. This analysis is provided through the use of the visualization tools of [47]. In the process, augmentations are made to these visualization tools for expandability and ease-of-use. Additionally, an analysis of these tools and the benefit each of them served is observed. This leads to intuitions of further applications for these tools. Lastly, and possibly of the utmost importance is the in-depth comparison of the material database, features, and neural network classifier utilized in this thesis to previous datasets, features, and techniques for classification. This provided a strong indication of areas that the neural network performed better than previous methods, such as its ability to filter very complex patterns of colors and structures as well as providing possible areas of improvement for the classifier, such as joint object and material classification, including material traits as features, and more robust training data (possibly provided through augmentations). This last contribution points to areas of further experimentation and possibly better performance from the classifier while also suggesting some areas that the classifier has outperformed previous methods.

REFERENCES

[1] S. Bell, P. Upchurch, N. Snavely, and K. Bala, *Material recognition in the wild with the Materials in Context Database (Supplemental Material)*, Tech. Rep.

[2] S. Bell, P. Upchurch, N. Snavely, and K. Bala, "OpenSurfaces," *ACM Transactions on Graphics*, 2013.

[3] S. Bell, P. Upchurch, N. Snavely, and K. Bala, "Material recognition in the wild with the Materials in Context Database," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2015, vol. 07-12-June, pp. 3479–3487.

[4] B. Caputo, E. Hayman, and P. Mallikarjuna, *Class-Specific Material Categorisation*, Tech. Rep.

[5] M. Cimpoi, S. Maji, I. Kokkinosécole, S. Mohamed, and A. Vedaldi, *Describing Textures in the Wild*, Tech. Rep.

[6] M. Crosier and L. D. Griffin, *Using Basic Image Features for Texture Classification*, Tech. Rep.

[7] G. Cybenkot, *Mathematics of Control, Signals, and Systems Approximation by Superpositions of a Sigmoidal Function**, Tech. Rep., 1989.

[8] K. J. Dana, J. J. Koenderink, K. J. Dana, S. K. Nayar, and J. J. Koenderink, *Reflectance and Texture of Real-World Surfaces*, Tech. Rep. 1, 1999.

[9] D. Erhan, Y. Bengio, A. Courville, and P. Vincent, *Visualizing Higher-Layer Features of a Deep Network Département d'Informatique et Recherche Opérationnelle*, Tech. Rep., 2009.

[10] Z. Erickson, S. Chernova, and C. C. Kemp, *Semi-Supervised Haptic Material Recognition for Robots using Generative Adversarial Networks*, Tech. Rep.

[11] M. Everingham, ·. Luc, V. Gool, ·. Christopher, K. I. Williams, J. Winn, A. Zisserman, M. Everingham, L. Van Gool, K. U. Leuven, B. C. K. I. Williams, J. Winn, and A. Zisserman, *The PASCAL Visual Object Classes (VOC) Challenge*, Tech. Rep.

[12] C. Farabet, C. Couprie, L. Najman, and Y. Lecun, “Learning hierarchical features for scene labeling,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, 2013, pp. 1915–1929.

[13] R. Girshick, J. Donahue, T. Darrell, and J. Malik, *Rich feature hierarchies for accurate object detection and semantic segmentation Tech report (v5)*, Tech. Rep.

[14] X. Glorot, A. Bordes, and Y. Bengio, *Deep Sparse Rectifier Neural Networks*, Tech. Rep., 2011.

[15] I. J. Goodfellow, J. Shlens, and C. Szegedy, *EXPLAINING AND HARNESSING ADVERSARIAL EXAMPLES*, Tech. Rep.

[16] A. Hannun, C. Case, J. Casper, B. Catanzaro, G. Diamos, E. Elsen, R. Prenger, S. Satheesh, S. Sengupta, A. Coates, and A. Y. Ng, *Deep Speech: Scaling up end-to-end speech recognition*, Tech. Rep.

[17] E. Hayman, B. Caputo, M. Fritz, and J.-O. Eklundh, *On the Significance of Real-World Conditions for Material Classification*, Tech. Rep.

[18] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, *Improving neural networks by preventing co-adaptation of feature detectors*, Tech. Rep.

[19] D. Hu, *Toward Robust Material Recognition for Everyday Objects*, Tech. Rep.

[20] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, “Caffe: Convolutional Architecture for Fast Feature Embedding,” *arXiv preprint arXiv:1408.5093*, 2014.

[21] E. Khan, E. Reinhard, R. Fleming, and H. Bülthoff, *Image-Based Material Editing*, Tech. Rep.

[22] P. Krähenbühl and V. Koltun, “Parameter Learning and Convergent Inference for Dense Random Fields,” *International Conference on Machine Learning*, vol. 3, no. June, 2013, pp. 513–521.

[23] A. Krizhevsky, I. Sutskever, and G. E. Hinton, *ImageNet Classification with Deep Convolutional Neural Networks*, Tech. Rep.

[24] T. Leung and J. Malik, *Representing and Recognizing the Visual Appearance of Materials using Three-dimensional Textons*, Tech. Rep. 1, 2001.

[25] W. Li and M. Fritz, *Recognizing Materials from Virtual Examples*, Tech. Rep.

[26] X. Li, G. Zhang, H. Howie Huang, Z. Wang, and W. Zheng, “Performance Analysis of GPU-based Convolutional Neural Networks,” 2016.

[27] C. Liu, L. Sharan, E. H. Adelson, and R. Rosenholtz, *Exploring Features in a Bayesian Framework for Material Recognition*, Tech. Rep.

[28] A. Mahendran and A. Vedaldi, *Understanding Deep Image Representations by Inverting Them*, Tech. Rep.

[29] A. Nguyen, J. Yosinski, and J. Clune, *Deep Neural Networks are Easily Fooled: High Confidence Predictions for Unrecognizable Images*, Tech. Rep.

[30] M. Oquab, L. Bottou, I. Laptev, J. Sivic, and L. Bottou, *Learning and Transferring Mid-Level Image Representations using Convolutional Neural Networks*, Tech. Rep.

[31] X. Qi, R. Xiao, C.-G. Li, Y. Qiao, S. Member, J. Guo, and X. Tang, "Pairwise Rotation Invariant Co-Occurrence Local Binary Pattern," .

[32] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, L. Fei-Fei, O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, *ImageNet Large Scale Visual Recognition Challenge*, Tech. Rep.

[33] F. Schroff and J. Philbin, *FaceNet: A Unified Embedding for Face Recognition and Clustering*, Tech. Rep.

[34] G. Schwartz and K. Nishino, "Visual Material Traits: Recognizing Per-Pixel Material Context," 2013.

[35] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. Lecun, *OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks*, Tech. Rep., 2014.

[36] L. Sharan, C. Liu, R. Rosenholtz, E. H. Adelson, L. Sharan, C. Liu, R. Rosenholtz, ·. E. H. Adelson, and E. H. Adelson, "Recognizing Materials Using Perceptually Inspired Features," *Int J Comput Vis*, vol. 103, 2013, pp. 348–371.

[37] L. Sharan, R. Rosenholtz, and E. Adelson, "Material perception: What can you see in a brief glance?," *Journal of Vision*, vol. 9, no. 8, 2010, p. 784784.

[38] K. Simonyan, A. Vedaldi, and A. Zisserman, *Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps*, Tech. Rep.

[39] K. Simonyan and A. Zisserman, *VERY DEEP CONVOLUTIONAL NETWORKS FOR LARGE-SCALE IMAGE RECOGNITION*, Tech. Rep., 2015.

[40] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, *Going Deeper with Convolutions*, Tech. Rep.

[41] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, *Intriguing properties of neural networks*, Tech. Rep.

[42] Y. Taigman, M. Y. Marc', A. Ranzato, and L. Wolf, *DeepFace: Closing the Gap to Human-Level Performance in Face Verification*, Tech. Rep.

[43] R. Timofte and L. Van Gool, *A Training-free Classification Framework for Textures, Writers, and Materials*, Tech. Rep., 2012.

[44] R. Uetz and S. Behnke, *Large-scale Object Recognition with CUDA-accelerated Hierarchical Neural Networks*.

[45] M. Varma and A. Zisserman, *A Statistical Approach to Texture Classification from Single Images*, Tech. Rep.

[46] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, *How transferable are features in deep neural networks?*, Tech. Rep.

[47] J. Yosinski, J. Clune, A. Nguyen, T. Fuchs, and H. Lipson, *Understanding Neural Networks Through Deep Visualization*, Tech. Rep.

[48] M. D. Zeiler, G. W. Taylor, and R. Fergus, *Adaptive Deconvolutional Networks for Mid and High Level Feature Learning*.

[49] S. Zheng, M.-M. Cheng, J. Warrell, P. Sturgess, V. Vineet, C. Rother, and P. H. S. Torr, *Dense Semantic Image Segmentation with Objects and Attributes*, Tech. Rep.

[50] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva, *Learning Deep Features for Scene Recognition using Places Database*, Tech. Rep.