Applied Computing and Intelligence
3(1): xx–xx
DOI: xxxx
Received: xx xx 2022
Revised: xx xx 2022
Accepted: xx xx 2022
Published: xx 2022

https://www.aimspress.com/journal/aci

Research Article

Ant cuticle image classification using texture analysis: a comparative study

Noah Gardner[1], John Paul Hellenbrand[2], Anthony Phan[1], Evan Zhu[1], Zhiling Long[4], Min Wang[3], Clint Penick[2], and Chih-Cheng Hung[1*]

[1] Laboratory of Machine Vision and Security Research, College of Computing and Software Engineering, Kennesaw State University, Marietta GA, USA

[2] College of Science and Mathematics, Kennesaw State University, Kennesaw, GA, USA

[3] Department of Mathematics, Kennesaw State University, Kennesaw, GA, USA

[4] Department of Mathematics, Science, and Informatics, Mercer University, Atlanta, GA, USA

* **Correspondence:** chung1@kennesaw.edu

Academic Editor: Pasi Fränti

Abstract: There is a large variety of ant species, and most species are diverse in terms of size, shape, behaviors, and especially cuticle textures. However, the significance of ant cuticle texture is not widely researched. Ant cuticle texture presumably provides some type of function, and therefore is useful to research for ecological applications and bioinspired designs. This research employs image texture analysis and deep machine learning to automatically group similar ant species based on morphological traits. We provide a comparative study of the performance of texture analysis methods on ant images. We evaluate the results of the classification methods with modern visualization techniques. TODO: what is the result? experiments? data?

Keywords: texture analysis; image processing; classification; machine learning; ant cuticle images; ecology

1. Introduction

TODO: refer to ant head images in a consistent way

TODO: refer to traditional/classical in a consistent way

TODO: change ideal and non-ideal to features identified by the biologist and to invalid features (background,

Insects compose half of biodiversity and rank among the most dominant organisms in terrestrial ecosystems [33]. A key factor for the ecological success of insects is their exoskeleton, also known as cuticle. The cuticle protects insects from predation, provides structural support, prevents desiccation, and serves as a canvas for advertising visual and chemical signals [13]. Research has heavily focused on the macrostructures and internal chemical components that make the exoskeleton functional and more recent work is being done to understand the functional aspects of external cuticle micro sculpturing [14, 29, 36].

Due to the extensive number of insect species, manual exploration of insect-based information is difficult and often requires specialized expertise. Therefore, automated entomology is gaining attraction by both biologists and computer scientists and is expected to be a major contribution to the future of insect-based research [28]. One of the most commonly used data types for insect analysis is image data. To develop an image-based system for insect analysis, we can take advantage of existing work in general image analysis methods.

We examine ants (*Formicidae*) as they display an extreme diversity of cuticle micro sculpturing across all subfamilies. Sculpturing ranges from parallel longitudinal ridges to deep oval impressions to erratic protuberances. The sculpturing has arisen convergently and independently throughout ant's evolutionary history, which suggests some inherent function. Cuticle sculpturing on ants may help increase strength and rigidness, resist abrasion, increase internal and external surface area, resist microbial growth, and rear beneficial anti-biotic producing bacteria [3, 6, 20]. These specific functions may be associated with certain sculpturing types. In order to analyze those functions, it is necessary to segment the image and group similar textures for further analysis.

In image texture analysis methods, the general goal is to automatically categorize an object into a set of objects with similar texture-based features. Texture analysis has shown promising results in related fields, such as plant identification [2] TODO: more references and examples. With modern texture analysis methods, the categorization of ants can be automated and the results can be used to study the influence of cuticle texture on ant ecology.

Image texture analysis heavily depends on the spatial relationships among gray levels of pixels [19]. Hence, feature extraction method is usually applied to the spatial relationship. The gray-level co-occurrence matrix (GLCM) and local binary patterns (LBP) are two of methods which are based on the spatial relationships to extracting features for this purpose [15, 30]. Traditional image texture analysis can be divided into four categories: statistical, structural, model-based, and transform-based methods [references]. In the past decade, the deep learning network becomes the mainstream in image classification and segmentation [24]. Compared with the traditional method in which a kernel must be designed for extracting features, the deep learning network can automatically extract the features by itself.

In this study, we used some of both approaches for our work on the ant images.

TODO:Specifically, what approaches used? Why do we use image texture for ant images? What are the contributions?

TODO: briefly discuss results from the research

2. Related Work

TODO: introduce the section

2.1. Sculpturing Identification

Taxonomists have developed extensive terminology describing ant cuticle sculpturing as it is often a useful diagnostic trait to distinguish between closely related ant species [1, 9]. The definitive text on ant cuticle terminology *The Glossary of Surface Sculpturing* contains over 100 terms to describe the cuticle sculpturing patterns of ants [16]. There is often substantial overlap among terms, and closely related cuticle patterns are likely to be functionally similar. For example, the definition for imbricate is, partly overlapping and appearing like shingles on a roof or scales on a fish, which is difficult to distinguish from the definition of tesselate, made up of squares like a chess board, either in sculpturing or in color.

Cuticle sculpturing in ants has been explored thoroughly from a taxonomic perspective; however, the function of nano and microstructures on insect exoskeletons is a developing topic in entomology. Watson et al. review the literature of cuticle nano and microstructure function and propose 21 possible functions associated with these structures [36]. Many of these functions are related to structures not found in ants such as scales and nanostructures. The review does include functions that may relate to ants such as friction control, enhanced surface area, and increased hardness. Watson et al. also describe seven types of cuticle structures ranging from hairs and scales to nano and micro structures [36]. The cuticle sculpturing of ants seems to fall within one type - complex microstructures.

The five broad functional groupings developed that describe the complex microstructures found on ants were derived from reviewing the variation of cuticle sculpturing across ants. These functional groupings were also reflected in Harris as many terms could be grouped together based on similar definitions and comparing the scanning electron microscope photographs provided in the publication [16].

TODO: how do we connect this section to image texture analysis? (why do we research this and why image te

2.2. Texture Analysis

Image texture analysis is an important part of image interpretation as many objects appearing in the textural patterns either partially or completely inside an image. This phenomenon appears frequently in many images we encountered in our environment as almost everybody is using the iPhone to capture the photos. Scientists are also studying their subjects depending on taking the images remotely such as remotely sensed satellite images. With an abundant number of textural images surrounding us, it is essential to have a reliable automatic tool for the textural image interpretation.

Classical computer methods are developed for textural images based on the spatial relationship of gray levels of pixels encoded in the texture [Haralick and Shapiro, Hung]. Those methods by de-facto follow the steps used in the statistical pattern recognition [Fukunaga]. Therefore, textural features are extracted, and the classification and segmentation methods are used for the interpretation either using the supervised or unsupervised approaches. The interpretation method is usually taken and modified from the traditional pattern recognition methods. For example, the gray-level co-occurrence matrix (GLCM) and local binary patterns (LBP) are two of methods for extracting features [references]. Some other approaches also have been developed for the textural feature extraction [references].

While most feature vectors are established based on a single pixel, Liu et al. gave a comprehensive survey on textural characterization in which they call this feature extraction as texture representation [Liu]. Their survey on Bag of Words and Convolutional Neural Networks are the spatial relationship concepts. The K-views was developed for taking the spatial information in the clustering approach [Hung]. In the past decade, the deep learning network has become the mainstream in image classification and segmentation [references]. Compared with the traditional method in which a kernel must be designed by an engineer for extracting features, the deep learning network can automatically extract the features through the training. In addition, the deep learning networks achieve the higher accuracy than that of the classical approaches although the deep networks require much more data for training.

Similar to the traditional pattern recognition, classical image texture analysis methods can be grouped into four categories: statistical, structural, model-based, and transform-based methods [references]. Among all of the categories, the statistical method is often used by employing the method from the pattern recognition principle. In doing so, an image texture is represented as feature vectors and then fed into the algorithm. Markov random field models are also studied for textural image interpretation [references]. The transform-based methods use some functions to decompose an image texture into a set of basic feature images. Gabor filters and Wavelet expansions are two of the widely used approaches [references].

TODO: add reference for sun phd thesis (Robust texture classification)

TODO: cleanup transition

Recently, the deep learning network becomes the main stream in the interpretation of textural images [Sun and liu]. In the past decade, deep learning using convolutional neural networks (CNNs) [22] has emerged as the state-of-the-art technology for image analysis. Following this trend, various CNN-based network architectures have been designed to specifically characterize texture images. Among these, the Fisher-vector CNN descriptor (FV-CNN) proposed by Cimpoi et al. [5] is widely accepted as one of the most important pioneering works. It applies FV pooling to deep features obtained via a CNN pre-trained using the ImageNet [22] to obtain encoded features for texture classification. Although it is capable of achieving much improved classification accuracy comparing to traditional hand-crafted texture features, FV-CNN does not support an end-to-end learning, where feature extraction, encoding, and classification are separated from each other. To achieve an end-to-end learning, Zhang et al. [40] proposed the deep texture encoding network (DeepTEN), in which a novel texture encoding layer is added to a standard CNN architecture. Then, Xue et al. [37] constructed the deep encoding pooling network (DEP), which improves over DeepTEN by integrating local spatial characteristics into the texture representation. Based on DeepTEN and DEP, Hu et al. [18] further developed the multi-level texture encoding and representation (MuLTER) network, which embeds a learnable encoding module at each convolutional layer so that encoding is performed for both low-level and high-level features, yielding a multi-level texture representation.

Other network architectures for end-to-end texture learning are also available. For example, in the deep multiple-attribute-perceived network (MAP-Net) [39], multiple perceptual attributes are progressively learned in a mutually reinforced manner through multiple branches. In the deep structure-revealed network (DSR-Net) [38], inherent structural representation for a texture pattern is obtained by employing a primitive capturing module to learn spatial primitives and a dependency learning module to capture the dependency among the primitives. In [27], a residual pooling layer consisting of a residual encoding module and an aggregation module is used to generate discriminative features of low dimensions. In [31], a histogram layer is designed to compute local spatial distribution of CNN features. In [4], an innovative aggregation module is presented to exploit statistical self-similarity across layers. All these architectures customize the standard CNN structure to accomplish the characterization of certain spatial, visual, or statistical nature unique to texture images.

2.3. Insect Classification

In this section, we provide an overview of some insect classification methods. Proposed insect classification methods seek to classify insects at different hierarchical levels, such as species, genus, family, and order. Additionally, some methods may classify insects at a combination of different hierarchical levels. Insect classification methods can be applied to a variety of fields. In agriculture, insect classification methods can be used to identify the presence of pest insects in crops, which can inform crop managers in their choice of pesticides and help prevent crop loss [21, 25].

Feng et al. [8] apply an automated system to classify moth images based on semantic related visual attributes, which are defined as a pattern on the moth wings. Feng et al. [8] use a custom texture descriptor based on the combination of GLCM and *scale-invariant feature transform* (SIFT) features [11, 26]. The method proposed by Feng et al. [8] is used to classify 50 different moth species across 8 families [8]. The results from Feng et al. [8] suggest that traditional feature extraction techniques for the semantic visual attributes of the moth wings are sufficient for training a classifier to classify an image between 10 randomly selected moth species.

Urteaga et al. [35] use machine learning methods in order to classify images between two different scorpion species: *Centruroides limpidus* and *Centruroides noxius*. After applying background distinction based on dynamic color threshold, Urteaga et al. [35] apply feature extraction to extract features from the separated scorpion image such as aspect ratio, rectangularity, and compactness. Urteaga et al. [35] apply three different classification models to classify the image as one of the species: Artificial Neural Network, Regression Tree, and Random Forest classifiers [35]. The results from Urteaga et al. [35] show that after background removal, characteristics from the entire body of the scorpion can be used to create a binary classifier that can classify the image as one of the two species.

Lim et al. [23] apply a CNN-based algorithm for insect classification. Lim et al. [23] classify a subset of insect species and families based on the classes available in the ImageNet dataset. ImageNet is a widely used dataset of images labeled by experts with millions of images and thousands of categories [7]. In the ImageNet dataset, there are some categories that specify the class of the insect on a species level, *e.g.* *monarch butterfly* and *ringlet butterfly* as well as some categories that specify the class of the insect on a family level, *e.g.* *ant*, *fly*, and *bee* [12]. Lim et al. [23] use a modified AlexNet architecture and experiment with different numbers of kernels and their effect the performance of the model. Glick et al. [10] employ a similar approach by classifying 277 insect classes from ImageNet using a hierarchical convolutional neural network. The results from Lim et al. [23] and Glick et al. [10] suggest that a CNN is capable of differentiating between different hierarchical classes of insects.

3. Methodology

TODO: do not jump into subsection directly without any context

3.1. Dataset preparation

In this section, we describe the creation of the custom dataset used in this research. In our dataset, we use ant head images from AntWeb [32] and define two categories for them based on the appearance

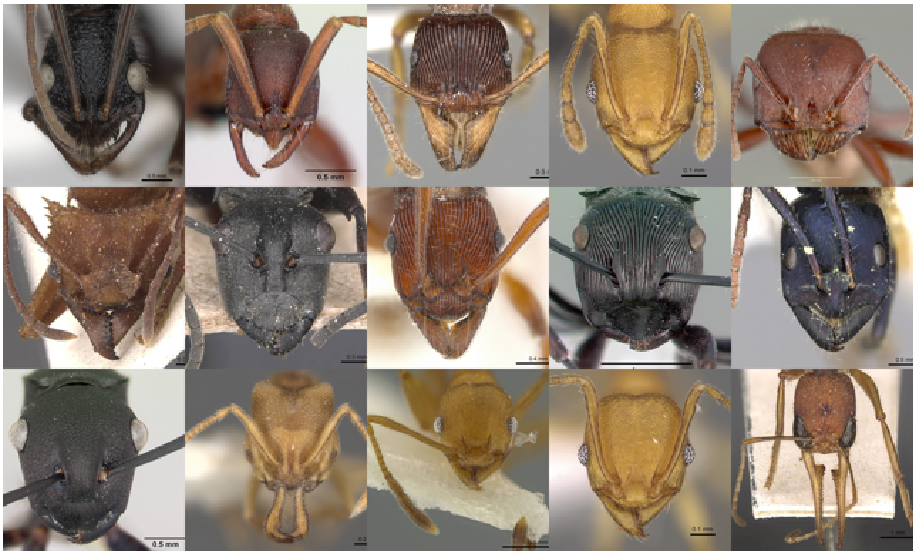


Figure 1. Examples of rough cuticle texture ant images in the dataset after center cropping, from AntWeb [32].

of the cuticle texture: *rough* and *smooth*. Some randomly selected images from each category are shown in Figures 1 and 2.

To begin, a master spreadsheet was created with the 2,499 different ant species to be identified for the primary dataset. The team was trained to identify cuticle sculpturing through a process which consisted of one 45-minute introductory lesson explaining the project and texture categories. Then, the team was given a training set of photos to identify from the genus *Polyrhachis*. The sculpture identification protocol describes the two primary categories: *rough* and *smooth*.

Initially, the sculpture identification protocol had 8 subcategories of cuticle texture, including dimpled, ridged, and differing levels of smooth texture. For simplicity, we work only with the two main categories. The training set identifications were reviewed together as a group by the assistants. Once training was complete, assistants were assigned the same genera of ants to identify independently each week. A weekly meeting was held to discuss identifications and assign new ones. These identifications were collected in the master spreadsheet and the identifications were assigned to individual ant species on a majority basis.

To collect the images, the assistants followed the taxonomy information available in the master spreadsheet to the appropriate AntWeb page [32]. In many cases, there are multiple ant head images of the same species, and occasionally there are multiple image resolution available from a single image. To simplify the data collection process, the assistants were instructed to download the first ant head image of the species being identified in the highest resolution possible. Each image was named with an identifier that corresponds with the row number in the master spreadsheet. The same ant head images that were downloaded in the data collection phase were the same ones used in the sculpture identification protocol. Ant species which did not have any images of the head were excluded from the dataset. Additionally, ant species which only had a head image of a queen ant were excluded from the dataset.

Ant specimen images taken from AntWeb [32] are created by different photographers and therefore have different attributes, such as environment, resolution, and lighting. In the ant head images, the ant

Figure 2. Examples of smooth cuticle texture ant images in the dataset after center cropping, from AntWeb [32].

head is in the center of the image and the body is pointing away from the camera. The focus of the ant head image is centered on the head, with the background and image artifacts from the ant body typically blurred. In most ant head images, there is a bar which indicates the scale of the image due to the variety in the sizes of different ant species. In a few ant head images, there exists some text denoting the specimen identifier and other information. In terms of texture, some ant specimens are very old, so their head images have other abnormalities such as cracks in the cuticle and the presence of dust.

Due to the variety of the ant head image attributes, we apply simple preprocessing before the images are used in our model. We want the images to have a uniform size for simplicity in our classification process. Since the ant head images are typically centered in the image, we apply a center crop to each image to create a square image of the same size. Once the image is square, we resize each image to a fixed size of 256x256 pixels. We leave other discrepancies in the images untouched.

Our custom dataset of ant head images contains 2,499 images. 1072 samples of rough textured ant cuticle textures comprise 43% of the dataset. The remaining 1427 samples of smooth textured ant cuticle textures comprise 57% of the dataset. To handle the imbalance of the dataset, we apply undersampling for each class for the training dataset. By using random stratified sampling, we construct a training set with 800 images per class. The remaining images are randomly split between test and validation, which turns out to roughly a 60%/20%/20% train, test, and validation data split. With 272 rough samples and 627 smooth samples left over after the stratified split, the test dataset has roughly 136 rough samples and 313 smooth samples. Since these leftover samples are split with code by 50% there will be some rounding variance and therefore the test dataset built at run-time will not always have exactly the same number of samples.

3.2. *Classical texture analysis methods*

TODO: kviews, kmeans, etc. will go here

3.3. *Deep learning models used for the experiments*

Our first model is *visual geometry group* (VGG), a convolutional neural network that takes advantage of very small convolutional filters in a deep network architecture [34]. We compare four architectures of VGG: VGG11, VGG13, VGG16, and VGG19. The primary difference between the architectures is the number of layers in each model. Our second model is *residual network* (ResNet), a deep network architecture that includes shortcut connections between layers (residual connections) [17]. We compare three architectures of ResNet: ResNet18, ResNet50, and ResNet101. Again, the primary difference between the architectures is the number of layers in each model.

For our ResNet models, we have two versions: randomized and pretrained. The randomized version is the same architecture, but the weights are randomly initialized. The pretrained version has weights from training on the CIFAR dataset, an image dataset with 1000 classes. In this case, we are fine-tuning the pretrained model. For VGG, we are only using the randomized version. The base VGG architecture also has an output layer of size 1000. Since we are working with a binary classification problem, we modify the architecture for all models to have an output layer of size 2. Each model is trained over 100 epochs, using stochastic gradient descent with momentum. The batch size is set to 16 images. We apply a learning rate of 0.001 and momentum parameter of 0.9.

TODO: I suggest that at least one deep learning based texture classification algorithm be applied to the dataset for the comparative study. When this is the case, one more subsection should be included here to describe that/those algorithm(s).

3.4. *Evaluation*

We evaluate the performance of the models according to standard evaluation methods. Since we are working with a binary classification problem, we use a standard confusion matrix to evaluate the accuracy, precision, and F1 score. (TODO: AUC and ROC can also be used as additional evaluation methods.) We also apply Grad-CAM with manual inspection to visualize the activation weights for classified images to visualize which features lead to the classification result. Finally, we apply t-SNE to visualize the separation learned for the model to further analyze the classifications made by the model.

4. Results

4.1. *Environment*

Experiments are run on an Ubuntu 18.04 LTS Lambda Labs GPU server. The server contains 8 NVIDIA GeForce RTX 2080 Ti graphics cards with 12GB of memory each. The server uses an Intel Xeon Silver 4116 with 48 total threads and maximum frequency of 3.000 GHz, and has 256GB of RAM.

4.2. *VGG Models*

To begin, we share the results on the VGG model architectures on our custom dataset. The classification results are summarized in a confusion matrix for each model. Then, the statistics for each VGG architecture are shared in Table 1. The results are collected on 8 runs of training and averaged. The results are rounded to 2 decimal places where appropriate.

Table 1. Average results for VGG architectures on ant head image dataset.

	Recall	Precision	F1 Score	Accuracy
VGG11	0.83	0.88	0.85	0.80
VGG13	0.87	0.87	0.87	0.82
VGG16	0.86	0.86	0.86	0.80
VGG19	0.87	0.87	0.87	0.81

4.3. ResNet Models

Next, we share the results on the ResNet model architectures on our custom dataset with random weight initialization. The classification results are summarized in a confusion matrix for each model. Then, the statistics for each ResNet architecture are shared in Table 2. The results are collected on 8 runs of training and averaged. The results are rounded to 2 decimal places where appropriate.

Table 2. Average results for ResNet architectures on ant head image dataset.

	Recall	Precision	F1 Score	Accuracy
ResNet18	0.84	0.87	0.85	0.80
ResNet50	0.86	0.83	0.84	0.77
ResNet101	0.83	0.84	0.83	0.76

4.4. Fine-tuned ResNet Models

Next, we share the results on the fine-tuned ResNet model architectures on our custom dataset with pretrained weights. The classification results are summarized in a confusion matrix for each model. Then, the statistics for each ResNet architecture are shared in Table 3. The results are collected on 8 runs of training and averaged. The results are rounded to 2 decimal places where appropriate.

Table 3. Average results for Fine-tuned ResNet architectures on ant head image dataset.

	Recall	Precision	F1 Score	Accuracy
ResNet18	0.89	0.94	0.91	0.88
ResNet50	0.89	0.92	0.91	0.87
ResNet101	0.89	0.94	0.92	0.88

TODO: Fix tables, decide on how to group models

Table 4. Results

	model	accuracy	precision	recall	f1	roc_auc
0	drp_multi_not_pt	0.87	0.89	0.92	0.91	0.86
1	drp_multi_pt	0.88	0.89	0.93	0.91	0.87
2	drp_single_not_pt	0.88	0.9	0.94	0.92	0.88
3	drp_single_pt	0.88	0.9	0.93	0.91	0.87
4	drp_single_aux_not_pt	0.87	0.89	0.92	0.9	0.86
5	drp_single_aux_pt	0.89	0.9	0.94	0.92	0.89
6	kviews_17	0.62	0.79	0.59	0.68	0.62
7	kviews_19	0.62	0.8	0.59	0.68	0.62
8	kviews_25	0.52	0.71	0.51	0.59	0.52
9	kmeans	0.47	0.07	0.35	0.11	0.47

Table 5. Clustering Results

	model	rand_score	adjusted_rand_score
0	kviews_17	0.53	0.06
1	kviews_19	0.53	0.06
2	kviews_25	0.5	-0.0
3	kmeans	0.5	0.0

5. The analysis with visualization

The results in the previous sections show that the fine-tuned ResNet models outperform the VGG and randomly initialized ResNet models on the task of ant head image classification. It should be noted that due to the class imbalance in the dataset, the F1 score is the preferable metric to the accuracy. On average, the fine-tuned ResNet101 model performed the best with an average F1 score of 0.92. We further analyze the separation learned by both ResNet101 models in the following section.

5.1. t-SNE Visualization

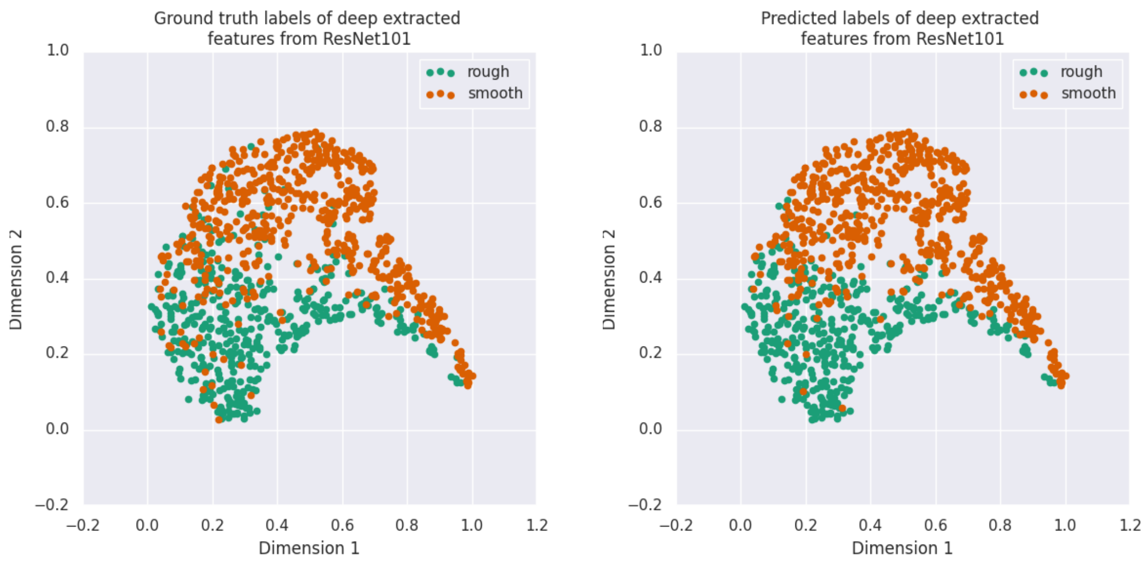


Figure 3. t-SNE visualization of the embeddings of the second to last layer of the randomly initialized ResNet101 model trained on ant head image dataset.

In this section, we provide visualization of the fine-tuned ResNet101 model and the randomly initialized ResNet101 model using t-SNE dimensionality reduction. First, we run the dataset preprocessing method and initialize both models. Then, both models are trained according to the training parameters and the state of each model is saved. To visualize the deep extracted features, we modify each model to obtain the embeddings of the second to last layer. Then, we use the t-SNE algorithm to reduce the dimensionality of the embeddings to 2 dimensions. We plot side-by-side the ground truth and predicted labels for each model. Figure 3 shows the results of the trained randomly initialized model and Figure 4 shows the results of the fine-tuned model.

Based on the visual results of the t-SNE visualization, we can see that the fine-tuned model learned a stronger separation of the two classes, which reinforces the results that the fine-tuned model received a higher average accuracy.

5.2. GradCAM Visualization

In this section, we provide some visual analysis of some correctly and incorrectly classified images using GradCAM. We provide two categories and two subcategories in our analysis. The two categories are correct and incorrect classification. Regardless of the feature activation map, the correctly classified images have the same predicted label as the ground truth, and incorrectly classified images have different predicted labels. The two subcategories are ideal and non-ideal feature activation. In the ideal case, the features that are used to compute the classification are the same as the features used by the assistants in the sculpture identification process. In general, the features used by the assistants

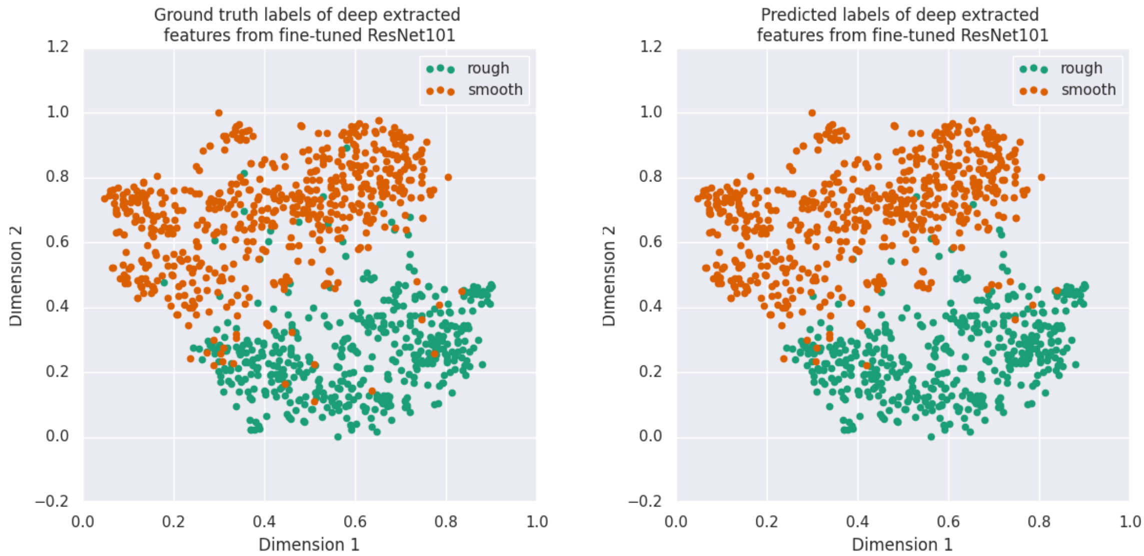
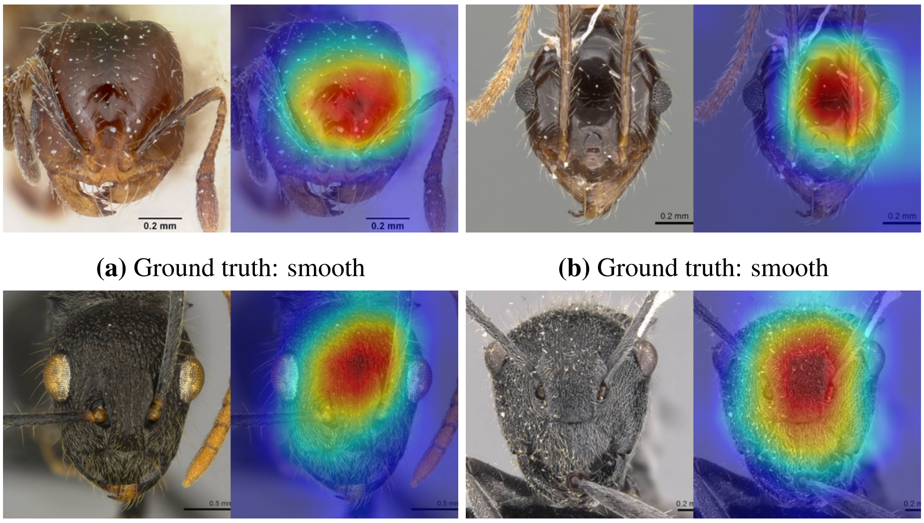


Figure 4. t-SNE visualization of the embeddings of the second to last layer of the fine-tuned ResNet101 model trained on ant head image dataset.

are the textures of the cuticle on the ant head. In the non-ideal case, the features used to compute the classification are not from the head, for example, from the background, extraneous text, or the body of the ant. We used randomly selected images from the dataset and the fine-tuned ResNet101 model to perform the analysis. We show the GradCAM results in Figures 5, 6, 7, and 8. The left image shows the preprocessed image input to the model. The right image shows the GradCAM output based on the classification. Four specimens were selected randomly from each category and subcategory.

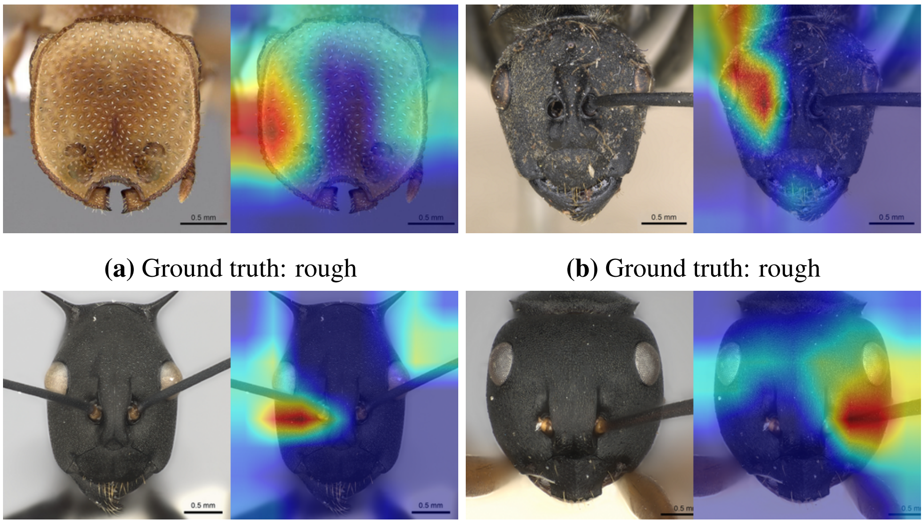
Correctly classified images which use ideal features show the ideal performance of the model. Incorrectly classified images which use ideal features should be further analyzed. In essence, the model in this situation knows *where* to look, but not *what* to look for. In Figure 7a, the features activated are mostly in the correct location on the ant head, and the rough texture is clearly visible, yet the model predicts the incorrect class *smooth*. Similarly in Figure 7c, the features activated are also mostly in the correct location, yet the model predicts the incorrect class *smooth*. In this case, it may be due to the pose of the ant being slightly different from the average pose. In the incorrectly classified images with non-ideal features, analysis shows that the model is unable to find *where* to look, and obtains feature information from other parts of the ant or the background. Cases where the image was correctly classified using the non-ideal features can basically be seen as noise. In order to further analyze this class, we should introduce some parameter such as model confidence to examine further.



(a) Ground truth: smooth **(b)** Ground truth: smooth

(c) Ground truth: rough **(d)** Ground truth: rough

Figure 5. Correctly classified images using ideal features.

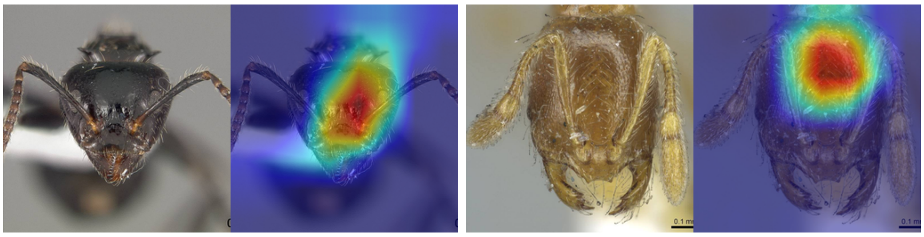


(a) Ground truth: rough **(b)** Ground truth: rough

(c) Ground truth: rough **(d)** Ground truth: smooth

Figure 6. Correctly classified images using non-ideal features.

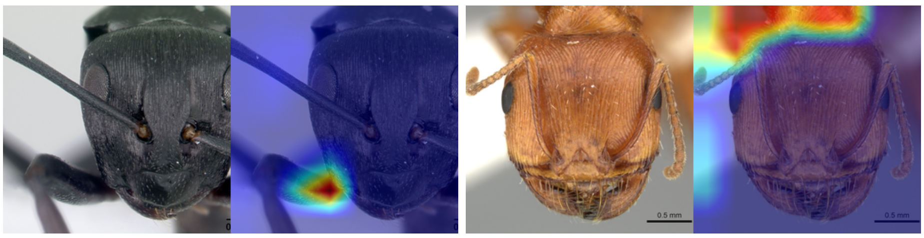
(a) Ground truth: rough **(b)** Ground truth: smooth



(c) Ground truth: smooth **(d)** Ground truth: smooth

Figure 7. Incorrectly classified images using ideal features.

(a) Ground truth: rough **(b)** Ground truth: rough



(c) Ground truth: rough **(d)** Ground truth: rough

Figure 8. Incorrectly classified images using non-ideal features.

6. Conclusion

Ant cuticle texture presumably has some function, but without the proper tools, evaluating the function based on thousands of species is infeasible. We have shown in this work that a deep learning approach and classical image texture analysis methods can be used to automatically categorize ants based on their cuticle texture, therefore supporting research on the evaluation of the function in future work. Our categorization system is novel in the field of automated insect identification due to the broad number of species captured by it. Additionally, a model that is pre-trained on a diverse image task such as ResNet can be transferred to our domain of texture analysis. All code is publicly available on GitHub (https://github.com/ngngardner/cuticulus).

TODO: mention how future work may include extra classes

References

1. B. B. Blaimer, Taxonomy and Natural History of the Crematogaster (Decacrema)-group (Hymenoptera: Formicidae) in Madagascar, *Zootaxa*, **2714** (2019), 1, URL `https://biotaxa.org/Zootaxa/article/view/zootaxa.2714.1.1`.
2. S. Boudra, I. Yahiaoui and A. Behloul, Plant identification from bark: A texture description based on Statistical Macro Binary Pattern, in *2018 24th International Conference on Pattern Recognition (ICPR)*, 2018, 1530–1535, ISSN: 1051-4651.
3. A. Brckner, M. Heethoff and N. Blthgen, The relationship between epicuticular long-chained hydrocarbons and surface area - volume ratios in insects (Diptera, Hymenoptera, Lepidoptera), *PLOS ONE*, **12** (2017), e0175001, URL `https://dx.plos.org/10.1371/journal.pone.0175001`.
4. Z. Chen, F. Li, Y. Quan, Y. Xu and H. Ji, Deep Texture Recognition via Exploiting Cross-Layer Statistical Self-Similarity, in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, Nashville, TN, USA, 2021, 5227–5236, URL `https://ieeexplore.ieee.org/document/9577662/`.
5. M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed and A. Vedaldi, Describing Textures in the Wild, in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 2014, 3606–3613, ISSN: 1063-6919.
6. C. R. Currie, Coevolved Crypts and Exocrine Glands Support Mutualistic Bacteria in Fungus-Growing Ants, *Science*, **311** (2006), 81–83, URL `https://www.sciencemag.org/lookup/doi/10.1126/science.1119744`.
7. J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li and L. Fei-Fei, ImageNet: A large-scale hierarchical image database, in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, 248–255, ISSN: 1063-6919.
8. L. Feng and B. Bhanu, Automated identification and retrieval of moth images with semantically related visual attributes on the wings, in *2013 IEEE International Conference on Image Processing*, 2013, 2577–2581, ISSN: 2381-8549.
9. B. L. Fisher and S. P. Cover, *Ants of North America: a guide to the genera*, University of California Press, Berkeley, 2007, OCLC: ocm80180487.

10. J. A. Glick and K. Miller, Insect Classification With Heirarchical Deep Convolutional Neural Networks Convolutional Neural Networks for Visual Recognition (CS 231 N), 2016, URL https://www.semanticscholar.org/paper/Insect-Classification-With-Heirarchical-Deep-Neural-Glick-Miller/bb8316bb841bf6667431bf755f6fe01ec013b8d5.

11. C. C. Gotlieb and H. E. Kreyszig, Texture descriptors based on co-occurrence matrices, *Computer Vision, Graphics, and Image Processing*, **51** (1990), 70–86, URL https://linkinghub.elsevier.com/retrieve/pii/S0734189X05800635.

12. C. Guestrin and E. Fox, text: imagenet 1000 class idx to human readable labels, URL https://gist.github.com/yrevar/942d3a0ac09ec9e5eb3a.

13. P. J. Gullan and P. S. Cranston, *Insects: an Outline of Entomology.*, Wiley, Hoboken, 2009, URL http://qut.eblib.com.au/patron/FullRecord.aspx?p=233169, OCLC: 1048427241.

14. S. Gunderson and R. Schiavone, The insect exoskeleton: A natural structural composite, *JOM*, **41** (1989), 60–63, URL http://link.springer.com/10.1007/BF03220386.

15. R. M. Haralick, K. Shanmugam and I. Dinstein, Textural Features for Image Classification, *IEEE Transactions on Systems, Man, and Cybernetics*, **SMC-3** (1973), 610–621, URL http://ieeexplore.ieee.org/document/4309314/.

16. R. A. Harris, A Glossary Of Surface Sculpturing., *Occasional Papers in Entomology*, URL https://zenodo.org/record/26215.

17. K. He, X. Zhang, S. Ren and J. Sun, Deep Residual Learning for Image Recognition, in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, 770–778, ISSN: 1063-6919.

18. Y. Hu, Z. Long and G. AlRegib, Multi-Level Texture Encoding and Representation (Multer) Based on Deep Neural Networks, in *2019 IEEE International Conference on Image Processing (ICIP)*, IEEE, Taipei, Taiwan, 2019, 4410–4414, URL https://ieeexplore.ieee.org/document/8803640/.

19. C.-C. Hung, E. Song and Y. Lan, *Image Texture Analysis: Foundations, Models and Algorithms*, Springer International Publishing, Cham, 2019, URL http://link.springer.com/10.1007/978-3-030-13773-1.

20. R. A. Johnson, A. Kaiser, M. Quinlan and W. Sharp, Effect of cuticular abrasion and recovery on water loss rates in queens of the desert harvester ant *Messor pergandei*, *Journal of Experimental Biology*, **214** (2011), 3495–3506, URL https://journals.biologists.com/jeb/article/214/20/3495/10482/Effect-of-cuticular-abrasion-and-recovery-on-water.

21. T. Kasinathan and S. R. Uyyala, Machine learning ensemble with image processing for pest identification and classification in field crops, *Neural Computing and Applications*, **33** (2021), 7491–7504, URL https://doi.org/10.1007/s00521-020-05497-z.

22. A. Krizhevsky, I. Sutskever and G. E. Hinton, ImageNet classification with deep convolutional neural networks, *Communications of the ACM*, **60** (2017), 84–90, URL https://dl.acm.org/doi/10.1145/3065386.

23. S. Lim, S. Kim and D. Kim, Performance effect analysis for insect classification using convolutional neural network, in *2017 7th IEEE International Conference on Control System, Computing and Engineering (ICCSCE)*, 2017, 210–215.

24. L. Liu, J. Chen, P. Fieguth, G. Zhao, R. Chellappa and M. Pietikinen, From BoW to CNN: Two Decades of Texture Representation for Texture Classification, *International Journal of Computer Vision*, **127** (2019), 74–109, URL `http://link.springer.com/10.1007/s11263-018-1125-z`.

25. L. Liu, R. Wang, C. Xie, P. Yang, F. Wang, S. Sudirman and W. Liu, PestNet: An End-to-End Deep Learning Approach for Large-Scale Multi-Class Pest Detection and Classification, *IEEE Access*, **7** (2019), 45301–45312.

26. D. G. Lowe, Distinctive Image Features from Scale-Invariant Keypoints, *International Journal of Computer Vision*, **60** (2004), 91–110, URL `http://link.springer.com/10.1023/B:VISI.0000029664.99615.94`.

27. S. Mao, D. Rajan and L. T. Chia, Deep residual pooling network for texture recognition, *Pattern Recognition*, **112** (2021), 107817, URL `https://linkinghub.elsevier.com/retrieve/pii/S0031320321000042`.

28. M. Martineau, D. Conte, R. Raveaux, I. Arnault, D. Munier and G. Venturini, A survey on image-based insect classification, *Pattern Recognit.*, **65** (2017), 273–284.

29. S. Muthukrishnan, S. Mun, M. Y. Noh, E. R. Geisbrecht and Y. Arakane, Insect Cuticular Chitin Contributes to Form and Function, *Current Pharmaceutical Design*, **26** (2020), 3530–3545, URL `https://www.eurekaselect.com/182235/article`.

30. T. Ojala, M. Pietikinen and T. Menp, Gray Scale and Rotation Invariant Texture Classification with Local Binary Patterns, in *Computer Vision - ECCV 2000* (eds. G. Goos, J. Hartmanis and J. van Leeuwen), vol. 1842, Springer Berlin Heidelberg, Berlin, Heidelberg, 2000, 404–420, URL `http://link.springer.com/10.1007/3-540-45054-8_27`.

31. J. Peeples, W. Xu and A. Zare, Histogram Layers for Texture Analysis, *IEEE Transactions on Artificial Intelligence*, 1–1, URL `https://ieeexplore.ieee.org/document/9652037/`.

32. V. Perrichot and B. Fisher, AntWeb: digitizing Recent and fossil insects for an online database of the ants of the world, in *Digital Fossil International Conference*, 2012, URL `https://hal-insu.archives-ouvertes.fr/insu-00805243`.

33. A. Sheikh, N. Rehman and R. Kumar, Diverse adaptations in insects: A review, *Journal of entomology and zoology studies*, **5** (2017), 343–350.

34. K. Simonyan and A. Zisserman, Very Deep Convolutional Networks for Large-Scale Image Recognition, *arXiv:1409.1556 [cs]*, URL `http://arxiv.org/abs/1409.1556`, ArXiv: 1409.1556.

35. J. C. Urteaga-Reyesvera and A. Possani-Espinosa, Scorpions: Classification of poisonous species using shape features, in *2016 International Conference on Electronics, Communications and Computers (CONIELECOMP)*, 2016, 125–129.

36. G. S. Watson, J. A. Watson and B. W. Cribb, Diversity of Cuticular Micro- and Nanostructures on Insects: Properties, Functions, and Potential Applications, *Annual Review*

of Entomology, **62** (2017), 185–205, URL `http://www.annualreviews.org/doi/10.1146/annurev-ento-031616-035020`.

37. J. Xue, H. Zhang and K. Dana, Deep Texture Manifold for Ground Terrain Recognition, in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Salt Lake City, UT, USA, 2018, 558–567, URL `https://ieeexplore.ieee.org/document/8578163/`.
38. W. Zhai, Y. Cao, Z.-J. Zha, H. Xie and F. Wu, Deep Structure-Revealed Network for Texture Recognition, in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, Seattle, WA, USA, 2020, 11007–11016, URL `https://ieeexplore.ieee.org/document/9157507/`.
39. W. Zhai, Y. Cao, J. Zhang and Z.-J. Zha, Deep Multiple-Attribute-Perceived Network for Real-World Texture Recognition, in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, IEEE, Seoul, Korea (South), 2019, 3612–3621, URL `https://ieeexplore.ieee.org/document/9009995/`.
40. Q.-s. Zhang and S.-c. Zhu, Visual interpretability for deep learning: a survey, *Frontiers of Information Technology & Electronic Engineering*, **19** (2018), 27–39, URL `https://doi.org/10.1631/FITEE.1700808`.

AIMS Press

© 2022 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0)