

A COMPARATIVE STUDY OF TEXTURE ANALYSIS METHODS ON ANT HEAD IMAGES

A Thesis Presented to
The Faculty of the Computer Science Department

by

Noah Gardner

In Partial Fulfillment
of Requirements for the Degree
Master of Science in Computer Science

Kennesaw State University

May 2022

In presenting this thesis as a partial fulfillment of the requirements for an advanced degree from Kennesaw State University, I agree that the university library shall make it available for inspection and circulation in accordance with its regulations governing materials of this type. I agree that permission to copy from, or to publish, this thesis may be granted by the professor under whose direction it was written, or, in his absence, by the dean of the appropriate school when such copying or publication is solely for scholarly purposes and does not involve potential financial gain. It is understood that any copying from or publication of, this thesis which involves potential financial gain will not be allowed without written permission.

Noah Gardner

Notice to Borrowers

Unpublished theses deposited in the Library of Kennesaw State University must be used only in accordance with the stipulations prescribed by the author in the preceding statement.

The author of this thesis is:

Noah Gardner
1100 S Marietta Parkway
Marietta, GA 30060

The director of this thesis is:

Chih-Cheng Hung
1100 S Marietta Parkway
Marietta, GA 30060

Users of this thesis not regularly enrolled as students at Kennesaw State University are required to attest acceptance of the preceding stipulations by signing below. Libraries borrowing this thesis for the use of their patrons are required to see that each user records here the information requested.

Name of user Address Date Type of use (examination only or copying)

A COMPARATIVE STUDY OF TEXTURE ANALYSIS METHODS ON ANT HEAD IMAGES

An Abstract of

A Thesis Presented to

The Faculty of the Computer Science Department

by

Noah Gardner

In Partial Fulfillment

of Requirements for the Degree

Master of Science in Computer Science

Kennesaw State University

May 2022

ABSTRACT

There is a large variety of ant species, and most species are diverse in terms of size, shape, behaviors, and especially skin (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 modern machine learning methods such as 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 modern texture analysis methods on ant head images. We evaluate the results of the classification methods with modern visualization techniques.

A COMPARATIVE STUDY OF TEXTURE ANALYSIS METHODS ON ANT HEAD IMAGES

A Thesis Presented to

The Faculty of the Computer Science Department

by

Noah Gardner

In Partial Fulfillment

of Requirements for the Degree

Master of Science in Computer Science

Advisor: Dr. Chih-Cheng Hung

Kennesaw State University

May 2022

To my family

Never fail to have this attitude of mind, go forward without hurry, learn the essence of things through frequent experiences, taking advantage of every occasion. Fight against all kinds of people and be aware of their mind. Follow a road that is a thousand leagues long one step at a time. Be without haste and be convinced that all these practices are the duty of a bushi. Be victorious today over what you were yesterday; tomorrow be victorious over your clumsiness and then also over your skill.

Miyamoto Musashi ¹

¹Miyamoto Musashi, The Book of Five Rings

ACKNOWLEDGEMENTS

I would have given up a long time ago without the support of my family. So, I would like to thank my loving family for their support throughout the completion of my graduate degree. I would also like to thank Dr. Chih-Cheng Hung for his mentorship for my thesis and for his guidance in my other research projects. I would like to thank Anthony Phan, my undergraduate assistant and friend for his help in running the experiments. Next, I would like to thank Dr. Zhiling Long for his insights for machine learning and texture analysis. I would like to thank John Paul Hellenbrand, his advisor Dr. Clint Penick, and their lab for providing the project and dataset. Finally, I would like to thank Dr. Coskun Tekes for providing the Lambda Labs GPU server for my experiments.

TABLE OF CONTENTS

ABSTRACT	v
ACKNOWLEDGEMENTS	ix
LIST OF FIGURES	xii
LIST OF TABLES	xiv
1 CHAPTER 1: INTRODUCTION	1
1.1 Introduction	1
1.2 Research Question	2
1.3 Proposed Approach	4
1.4 Research Impact	4
1.5 Contributions	5
2 CHAPTER 2: BACKGROUND	6
2.1 Texture Analysis	6
2.2 Deep Learning	7
2.3 Evaluation	8
2.3.1 Quantitative Metrics	8
2.3.2 Qualitative Metrics	11
3 CHAPTER 3: LITERATURE REVIEW	14
3.1 Insect Classification	14
3.2 Automated Ant Identification	18
4 CHAPTER 4: METHODOLOGY	20

4.1	Dataset	20
4.1.1	Sculpture Identification Protocol	20
4.1.2	Data Collection	22
4.1.3	Data Preprocessing	23
4.2	Models	24
4.2.1	Model Training	25
4.2.2	Evaluation	27
5	CHAPTER 5: EXPERIMENTAL RESULTS AND ANALYSIS	28
5.1	Environment	28
5.2	Results	29
5.2.1	VGG Models	29
5.2.2	ResNet Models	31
5.2.3	Fine-tuned ResNet Models	33
5.3	Analysis	35
5.3.1	t-SNE Visualization	35
5.3.2	GradCAM Visualization	37
6	CHAPTER 6: CONCLUSION	43
6.1	Conclusion	43
6.2	Future Work	43
	REFERENCES	45

LIST OF FIGURES

1	An example ant head image from AntWeb of species <i>Polyrhachis abbreviata</i> . CASENT0217419 by Estella Ortega, from AntWeb, is licensed under CC BY 4.0.	3
2	A sample of the describable texture dataset, from Cimpoi <i>et al.</i> [16].	6
3	Example image of a confusion matrix, by Aatish Kayyath, is licensed under CC BY 4.0.	8
4	Example image of Grad-CAM applied to images of 'keyboards' from the COCO dataset, from Selvaraju <i>et al.</i> [22].	11
5	Example image of t-SNE embeddings of MNIST dataset, by Kyle McDonald, is licensed under CC BY 4.0.	12
6	Taxonomy chart showing taxonomy of the honey bee and large milkweed bug. Taxonomy chart by Adam Dolezal, Page Baluch, from Ask a Biologist, is licensed under CC BY 4.0.	14
7	An example image of a set of moth wings for moth species identification, from Feng <i>et al.</i> [28].	15
8	An example image of a scorpion for a scorpion species classifier with the background removed, from Urteaga-Reyesvera <i>et al.</i> [31].	16
9	An example image of three distinct ant species from ImageNet, from Glick <i>et al.</i> [32].	17
10	An example output of an individual ant track from a group of outdoor ants, from Cao <i>et al.</i> [39].	18

11	Examples of rough cuticle texture ant images in the dataset after center cropping.	20
12	Examples of smooth cuticle texture ant images in the dataset after center cropping.	21
13	VGG16 model architecture, from neurohive.io , is licensed under CC BY 4.0.	24
14	ResNet34 model architecture compared to VGG19 model architecture, from He <i>et al.</i> [42].	26
15	Neofetch output for the Lambda Labs GPU server used for experiments.	28
16	Average accuracy for VGG architectures on ant head image dataset.	30
17	Average accuracy for ResNet architectures on ant head image dataset.	32
18	Average accuracy for fine-tuned ResNet architectures on ant head image dataset.	34
19	t-SNE visualization of the embeddings of the second to last layer of the randomly initialized ResNet101 model trained on ant head image dataset.	35
20	t-SNE visualization of the embeddings of the second to last layer of the fine-tuned ResNet101 model trained on ant head image dataset.	36
21	Correctly classified images using ideal features.	39
22	Correctly classified images using non-ideal features.	40
23	Incorrectly classified images using ideal features.	41
24	Incorrectly classified images using non-ideal features.	42

LIST OF TABLES

1	VGG11 confusion matrix on ant head image dataset.	29
2	VGG13 confusion matrix on ant head image dataset.	29
3	VGG16 confusion matrix on ant head image dataset.	29
4	VGG19 confusion matrix on ant head image dataset.	30
5	Average results for VGG architectures on ant head image dataset. .	30
6	ResNet18 confusion matrix on ant head image dataset.	31
7	ResNet50 confusion matrix on ant head image dataset.	31
8	ResNet101 confusion matrix on ant head image dataset.	31
9	Average results for ResNet architectures on ant head image dataset.	32
10	Fine-tuned ResNet18 confusion matrix on ant head image dataset. .	33
11	Fine-tuned ResNet50 confusion matrix on ant head image dataset. .	33
12	Fine-tuned ResNet101 confusion matrix on ant head image dataset.	33
13	Average results for Fine-tuned ResNet architectures on ant head image dataset.	34

1 CHAPTER 1: INTRODUCTION

1.1 Introduction

Insects comprise over half of the world's animal biodiversity [1]. Insects are vital to many ecosystem functions, including nutrient recycling, plant propagation, and maintenance of plant and animal communities [2], [3]. As technology advances, systems which can automatically analyze insect-based information are growing in demand by biologists [4]. For example, automatic insect identification systems can be applied to conserve natural ecosystems, to prevent commercial loss, and to support the study of insect-related diseases [5]–[7].

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 [8]. 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 processing and texture analysis methods.

Ants are highly diverse, fill many ecological niches, and thus are a major topic of interest [9]. In this work, we explore automated entomology specifically for ants

in order to support ant-based ecological research. Although there is some work regarding grouping ants into categories of similar exoskeleton texture (*cuticle*), automated categorization has yet to become an active area of research. In general, the goal of automated insect classification methods is species identification - *i.e.* the identification of species from a set of observations. Due to the large number of different ant species, large scale ant species identification with standard classification methods is not feasible. Therefore, we must simplify the problem by either other classifying a certain subset of ant species or by applying categories to the entire set of ant species.

In many texture analysis methods, the general goal is to automatically categorize an object into a set of objects with similar texture-based features. The approach of texture analysis to categorize similarly texture-based objects corresponds well to the demands of ant identification. Texture analysis has shown promising results in related fields, such as plant identification [10]. 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.

1.2 Research Question

The overarching question that we wish to address by beginning this research is: *how does the texture of ant cuticle affect the ant ecology?* However, to even begin contemplating this question requires a substantial amount of preliminary

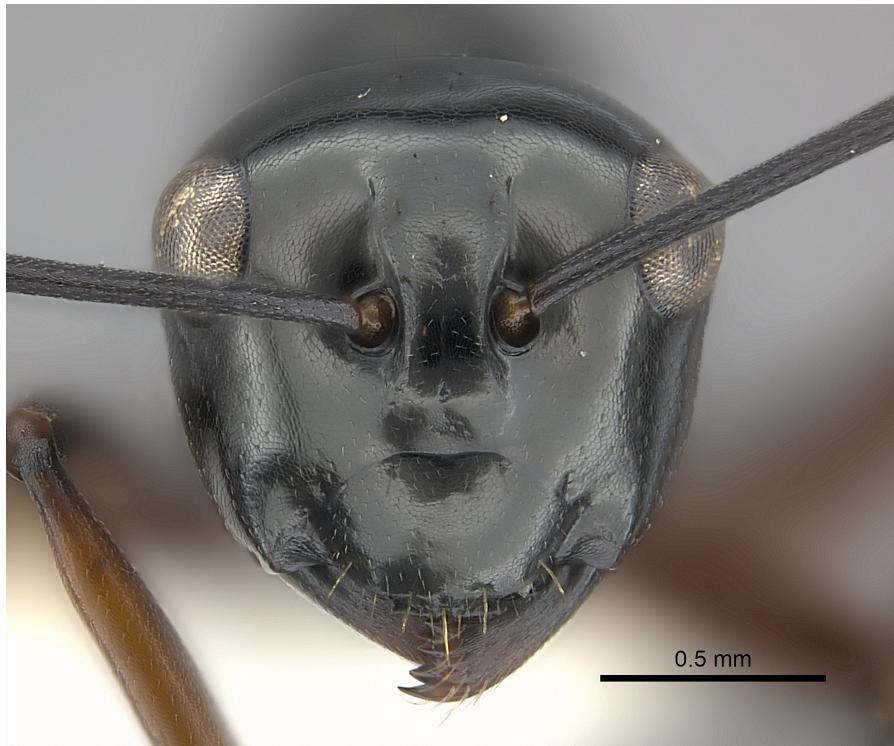


Figure 1: An example ant head image from AntWeb of species *Polyrhachis abbreviata*. CASENT0217419 by Estella Ortega, from [AntWeb](#), is licensed under [CC BY 4.0](#).

research. One point that is necessary to begin this research is to create a method of group ants by texture. Additionally, due to the sheer number of ant species, an automated effort is necessary to group similar ant species. Therefore, we start our endeavor with a more straightforward research question: *are texture analysis methods able to group similar ant species?* By the end of this research, we answer this question by demonstrating a variety of texture analysis methods and comparing their results on a custom dataset.

1.3 Proposed Approach

In order to group similar ant species, we require mostly uniform images that depict the texture of the ant cuticle. We source our raw images from AntWeb [11], a database of ant head images. An example image is shown in Figure 1. In general, the ant head images are centered in the image, facing the front, and share a similar posture. However, some images may not be centered, may show the ant head in a different orientation, or may have a drastically different resolution from the average image. With some data preprocessing methods, these ant head images are suitable for traditional methods of image classification. Therefore, we apply several image classification and texture analysis methods to a custom dataset of ant head images and compare the results using standard evaluation metrics.

1.4 Research Impact

There are two main research impacts that we expect to support from this research. First is the ecological and biodiversity aspect - due to the large number of ant species that are crucial to nearly every ecosystem, by supporting ant ecological research we are able to support biodiversity research which is essential for supporting all life [12]. The second research impact is the application of exoskeleton sculpturing to bioinspired designs, especially for material science. A prime example of bioinspired material comes from the *Sharklet micropattern*, a

surface material that gains inspiration from the skin of sharks [13]. The skin of sharks has a particular physical pattern that has been shown to inhibit the growth of bacteria and other microorganisms [13]. The bioinspired Sharklet micropattern is a promising material that shows similar properties to shark skin [14]. Presumably, the texture of the ant cuticle is functional, and therefore, has some sort of value to bioinspired material design research. Due to the wide variety of ant cuticle textures, it is hypothesized that the sculpturing may have a significant impact on thermoregulation, water retention, and structural protection [15]. By understanding the function of the form, we can replicate the form in order to reproduce a function.

1.5 Contributions

The primary contribution of this research is the development of a unique dataset for ant cuticle texture classification. Additionally, we compare the results of state-of-the-art deep learning and texture analysis methods on our proposed dataset. The secondary contribution is the analysis of the results of these methods and data analysis.

2 CHAPTER 2: BACKGROUND

2.1 Texture Analysis

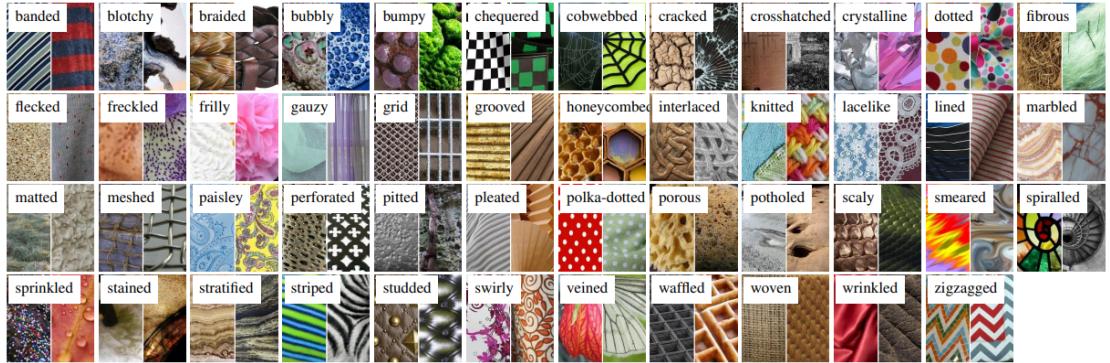


Figure 2: A sample of the describable texture dataset, from Cimpoi *et al.* [16].

A visual attribute is a characteristic of an object that is measured visually and also has some semantic meaning [16]. For example, the color of an object is a visual attribute. By categorizing objects based on their common visual attributes, we can obtain more detailed information about the relationships between objects. Patterns and textures are important visual attributes in many applications, such as image processing, pattern recognition, and computer vision. Analysis of textures can be broken into three main categories: texture classification, texture segmentation, and texture synthesis [17]. The process of classifying a texture into a set of categories and relies on three different approaches. In this paper, we focus on a *model-based approach* which attempts to extract parameters to reveal common patterns and use those parameters to automatically distinguish between different textures [18].

2.2 Deep Learning

In the past, automated classification methods depended on human-engineered feature extraction processes, which tended to not generalize to other domains [19]. More recently, neural networks have been shown to be able to automatically extract features from input data [20]. More specifically, *convolutional neural networks* (CNN) can be applied to multidimensional data (especially images) and learn high-level features that can be used to automatically classify data [21]. CNNs have been around for a few decades currently, but improvements have been made over recent years which have allowed CNNs to have very good performance on a wide range of tasks.

2.3 Evaluation

2.3.1 Quantitative Metrics

In estimating the performance of a classification model, we wish to evaluate the ability of the model to make correct classifications. In order to do so, we simply count the number of correct and incorrect classifications on a test dataset. To visualize a summary of the models' performance, we can use a confusion matrix. A confusion matrix is especially useful in a setting with two classes, but can also be used in a setting with more than two distinct classes. An example of a confusion matrix is shown in Figure 3. In a two class setting, the confusion matrix is a square matrix with four entries. The columns of the matrix represent the true value of the data, typically as determined by an expert. The rows of the matrix represent the predicted value of the data, as determined by the model. Using the confusion matrix, we can derive several meaningful evaluations of the model's performance.

		Actual Values	
		Positive (1)	Negative (0)
Predicted Values	Positive (1)	TP	FP
	Negative (0)	FN	TN

Figure 3: Example image of a confusion matrix, by [Aatish Kayyath](#), is licensed under [CC BY 4.0](#).

To begin, we distinguish the following notations:

- *True Positive* (TP) - The number of times the model correctly classifies a sample as a positive (belonging to the '1' class).
- *False Positive* (FP) - The number of times the model incorrectly classifies a sample as a positive (belonging to the '1' class).
- *True Negative* (TN) - The number of times the model correctly classifies a sample as a negative (belonging to the '0' class).
- *False Negative* (FN) - The number of times the model incorrectly classifies a sample as a negative (belonging to the '0' class).

Accuracy is a measure of the correctly classified samples as a percentage of the total number of samples. The accuracy of a classification model is calculated as:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

Accuracy is a standard evaluation metric for classification models, however, it can hide the fact that a classification model may be overfitted when the dataset is imbalanced. We add some other evaluation metrics based on the confusion matrix.

Recall is a measure of the correctly classified samples as a percentage of the samples that belong to the positive class. The recall of a classification model is calculated as:

$$Recall = \frac{TP}{TP + FN} \quad (2)$$

Recall is typically used in conjunction with the precision. Precision is a measure of the correctly classified samples as a percentage of the samples that were predicted to belong to the positive class in addition to the samples that were actually positive. The precision of a classification model is calculated as:

$$Precision = \frac{TP}{TP + FP} \quad (3)$$

Finally, we examine the F1 score. The F1 score is the harmonic mean of the precision and recall. The F1 score of a classification model is calculated as:

$$F1\ Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (4)$$

2.3.2 Qualitative Metrics

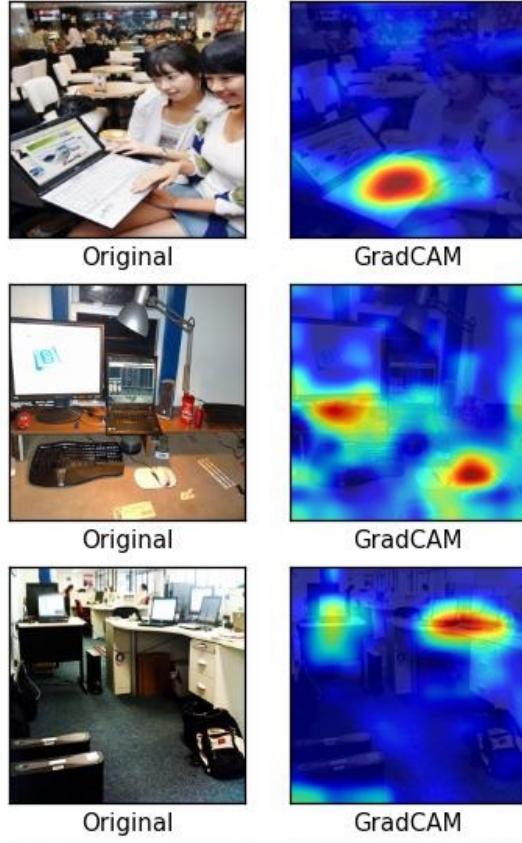


Figure 4: Example image of Grad-CAM applied to images of 'keyboards' from the COCO dataset, from Selvaraju *et al.* [22].

In general, CNN have very good performance on visual recognition tasks.

However, because the learning strategy of a CNN is end-to-end, it is often called a *black box* - *i.e.* it is difficult to visually interpret the internal workings of a CNN, yet it is easy to understand the inputs and outputs [23]. One method proposed to produce *visual explanations* for CNNs is the *gradient-weighted class activation mapping* (Grad-CAM) method [22]. By using the gradients of a class from the final convolutional layer, the important regions that lead to the classification can be identified and visualized. With Grad-CAM, we can understand the important features learned by the model and how the final classification was made. Figure 4 shows an example of Grad-CAM applied to a

model that classifies the images of keyboards from the COCO dataset. The left-side of the figure shows the image that was input to the model, while the right-side shows the output of Grad-CAM. The heatmap clearly shows the important features of the image which made the model classify the image as a keyboard. Applying Grad-CAM will allow us to increase the interpretability of our models.

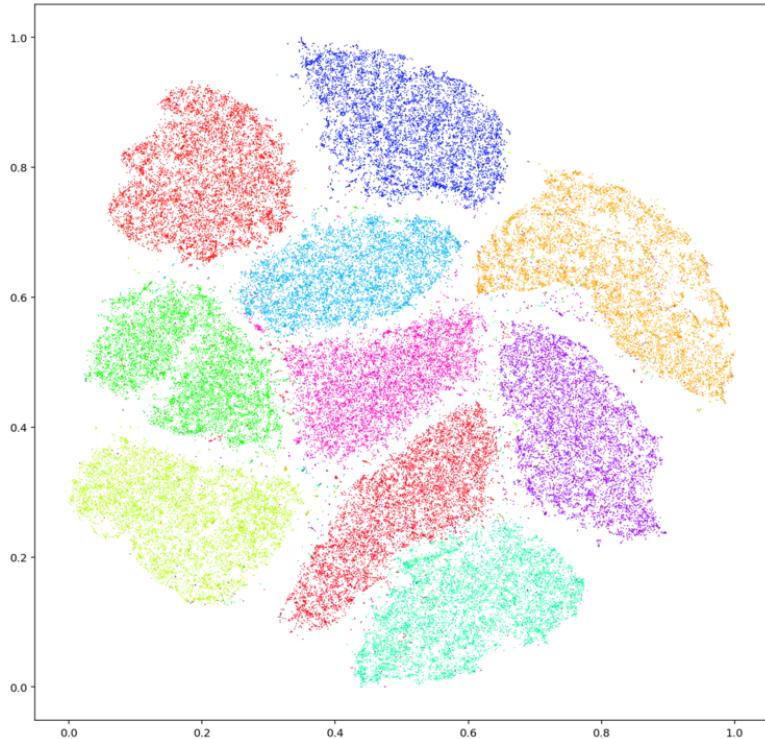


Figure 5: Example image of t-SNE embeddings of MNIST dataset, by [Kyle McDonald](#), is licensed under [CC BY 4.0](#).

For large scale visualization of our data, we must first tackle the dimensionality of our image dataset. Image data has a large number of dimensions as a result of each pixel in the image being represented by a tuple of three values (RGB).

T-distributed stochastic neighbor embedding (t-SNE) was proposed for visualizing high-dimensional data in two to three dimensions [24]. t-SNE finds similarity between datapoints by mapping distances into conditional probabilities [24]. For

an example, the MNIST dataset is a set of images of handwritten digits from zero to nine, and have a size of 28x28 pixels, for a total of 784 features [25]. An example output embedding of t-SNE on the MNIST dataset is shown in Figure 5.

3 CHAPTER 3: LITERATURE REVIEW

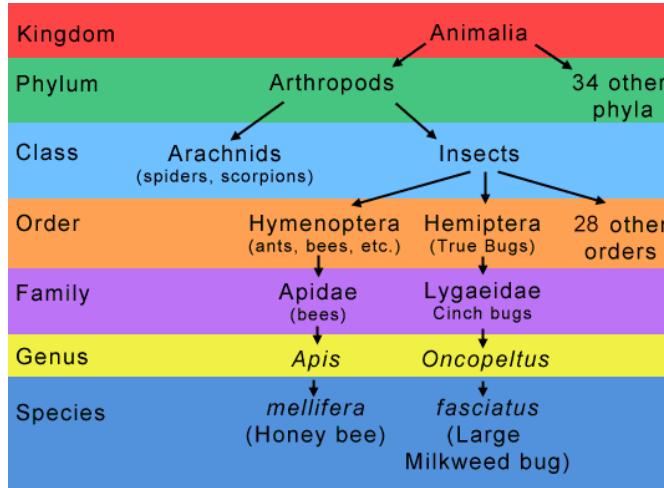


Figure 6: Taxonomy chart showing taxonomy of the honey bee and large milkweed bug. [Taxonomy chart](#) by Adam Dolezal, Page Baluch, from [Ask a Biologist](#), is licensed under [CC BY 4.0](#).

3.1 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. An example taxonomy chart with some insect taxonomy is shown in Figure 6. 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 [26], [27].

Feng *et al.* apply an automated system to classify moth images based on



Figure 7: An example image of a set of moth wings for moth species identification, from Feng *et al.* [28].

semantic related visual attributes, which are defined as a pattern on the moth wings [28]. Feng *et al.* use a custom texture descriptor based on the combination of Grey Level Co-occurrence Matrix and *Scale-Invariant Feature Transform* (SIFT) features [29], [30]. The method proposed by Feng *et al.* is used to classify 50 different moth species across 8 families and is based on standard texture analysis features [28]. A set of example moth wing images is shown in Figure 7. The results from Feng *et al.* 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 [28].

Urteaga-Reyesvera *et al.* use machine learning methods in order to classify images between two different scorpion species: *Centruroides limpidus* and *Centruroides noxius* [31]. After applying background distinction based on dynamic color threshold, Urteaga-Reyesvera *et al.* apply feature extraction to extract features from the separated scorpion image such as aspect ratio, rectangularity, and compactness [31]. Urteaga-Reyesvera *et al.* apply three different classification models to classify the image as one of the species:

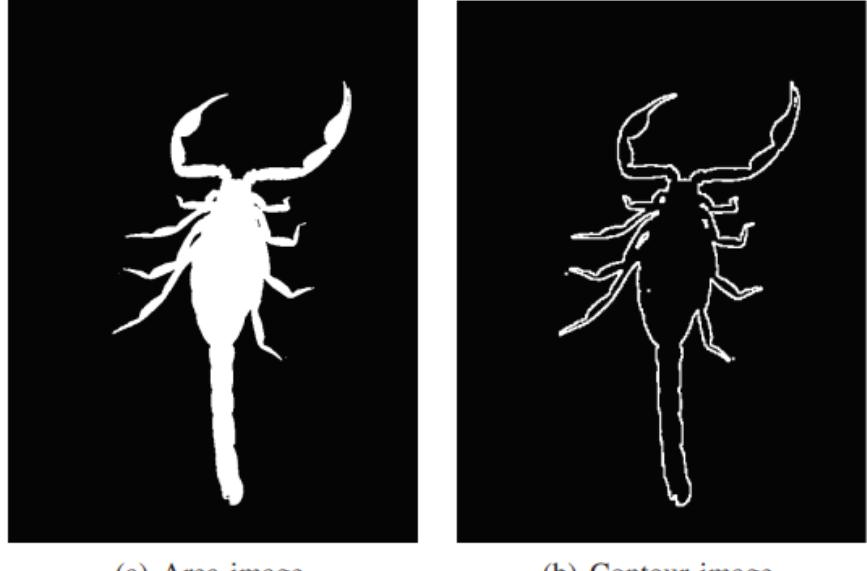


Figure 8: An example image of a scorpion for a scorpion species classifier with the background removed, from Urteaga-Reyesvera *et al.* [31].

Artificial Neural Network, Regression Tree, and Random Forest classifiers [31].

An example image of a scorpion with the background removed is shared in Figure 8. The results from Urteaga-Reyesvera *et al.* 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. Therefore, we may consider an experiment on the presence of the background from the ant head image in order to improve classification accuracy. However, the features extracted from [31] do not make use of any deep learning methods and are not compatible with our texture-based dataset.

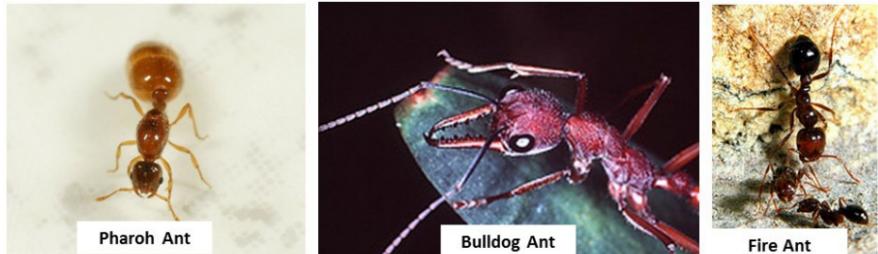


Figure 9: An example image of three distinct ant species from ImageNet, from Glick *et al.* [32].

Lim *et al.* apply a CNN-based algorithm for insect classification [33]. Lim *et al.* classify a subset of insect species and families based on the classes available in the ImageNet dataset [34]. ImageNet is a widely used dataset of images labeled by experts with millions of images and thousands of categories [34]. 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* [35]. Lim *et al.* use a modified AlexNet architecture and experiment with different numbers of kernels and their effect the performance of the model [33]. Glick *et al.* employ a similar approach by classifying 277 insect classes from ImageNet using a hierarchical CNN [32]. The results from Lim *et al.* and Glick *et al.* suggest that a CNN is capable of differentiating between different hierarchical classes of insects. In our research, we are interested in the classification of ant images of a hierarchical level between species and family.

3.2 Automated Ant Identification

In this section, we explore some machine learning methods applied directly to automated ant systems. We define automated ant systems as methods which process information about ants in a computer environment. Many automated ant systems have the purpose of tracking individual ants and their movements. An example of ant tracking software is shown in Figure 10. Systems that can track multiple individuals in a colony are applied to support research that investigates social group behaviors of ants [36]–[38].

Sosiak *et al.* use morphometric traits as features to analyze the relationship between the traits and the ecology of the ant [9]. Sosiak *et al.* apply a supervised Random Forest algorithm to classify ants based on their ecology [9]. The research carried out by Sosiak *et al.* is similar to the end goal of this research - by creating a system that categorizes ants based on their cuticle texture, we hope to support a framework that can be used to analyze the ecological niche of ants in future work.

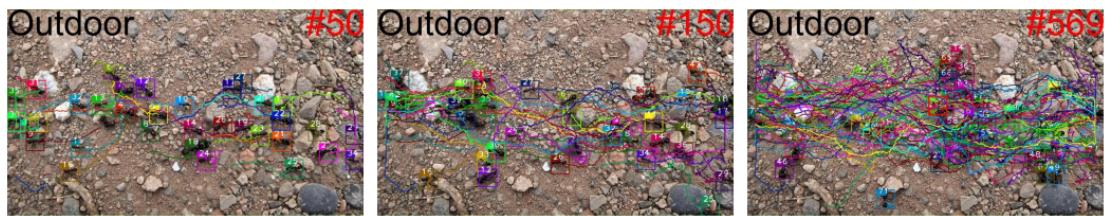


Figure 10: An example output of an individual ant track from a group of outdoor ants, from Cao *et al.* [39].

Cao *et al.* apply ResNet to automatically extract deep features from individual ants for offline learning [39]. Cao *et al.* modify the parameters of the softmax

classifier to obtain a cosine similarity metrix classifier to automatically track 10 individual ants in a group [39]. Cao *et al.* use a dataset of 50 images for training and test their model on a much larger dataset of 24000 images [39]. The results from Cao *et al.* suggest that ResNet is able to differentiate between individual ants.

Gal *et al.* also propose a framework for tracking inidividual insects by taking advantage of color tagging. Gal *et al.* show examples of their software tracking individual ants from a group that are identified with different colors of paint applied to the ants [40]. The method from Gal *et al.* should generalize to any type of insect and have relaxed requirements for image quality due to the constraint of applying color tags [40]. However, the images of the group of ants used for multi object tracking is not similar to our dataset of ant head images, other than the fact that both datasets were collected in a laboratory setting.

4 CHAPTER 4: METHODOLOGY

4.1 Dataset



Figure 11: Examples of rough cuticle texture ant images in the dataset after center cropping.

In this section, we describe the creation of the custom dataset used in this research. In our dataset, we classify ant head images from AntWeb [11] into two categories: *rough* and *smooth*. Some randomly selected images from each category are shown in Figures 11 and 12.

4.1.1 Sculpture Identification Protocol

To begin, a master spreadsheet was created with the 2,499 different ant species to be identified for the primary dataset. A team of three assistants were in charge



Figure 12: Examples of smooth cuticle texture ant images in the dataset after center cropping.

of the manual identification process. 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 and then given a training set of photos to identify from the genus *Polyrhachis*, whose members display among the highest diversity of cuticle patterns all categories. 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.

4.1.2 Data Collection

To collect the images, the assistants followed the taxonomy information available in the master spreadsheet to the appropriate AntWeb page. 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 [11] are created by different photographers and therefore have different attributes, such as environment, resolution, and lighting. In the ant head images, the ant 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 info. 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.

4.1.3 Data Preprocessing

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

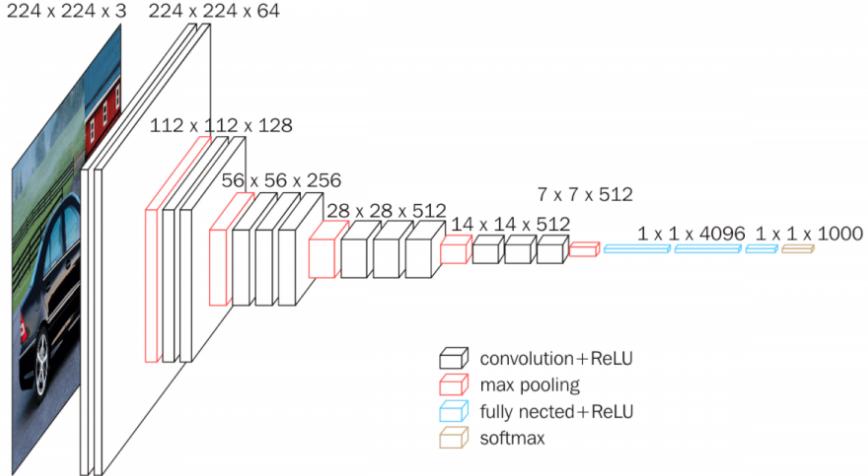


Figure 13: VGG16 model architecture, from [neurohive.io](#), is licensed under CC BY 4.0.

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.

4.2 Models

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 [41]. 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. An example of the VGG16 architecture is shown

in Figure 13.

Our second model is *residual network* (ResNet), a deep network architecture that includes shortcut connections between layers (residual connections) [42]. 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. An example of the ResNet34 architecture next to VGG19 architecture is shown in Figure 14.

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.

4.2.1 Model Training

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.

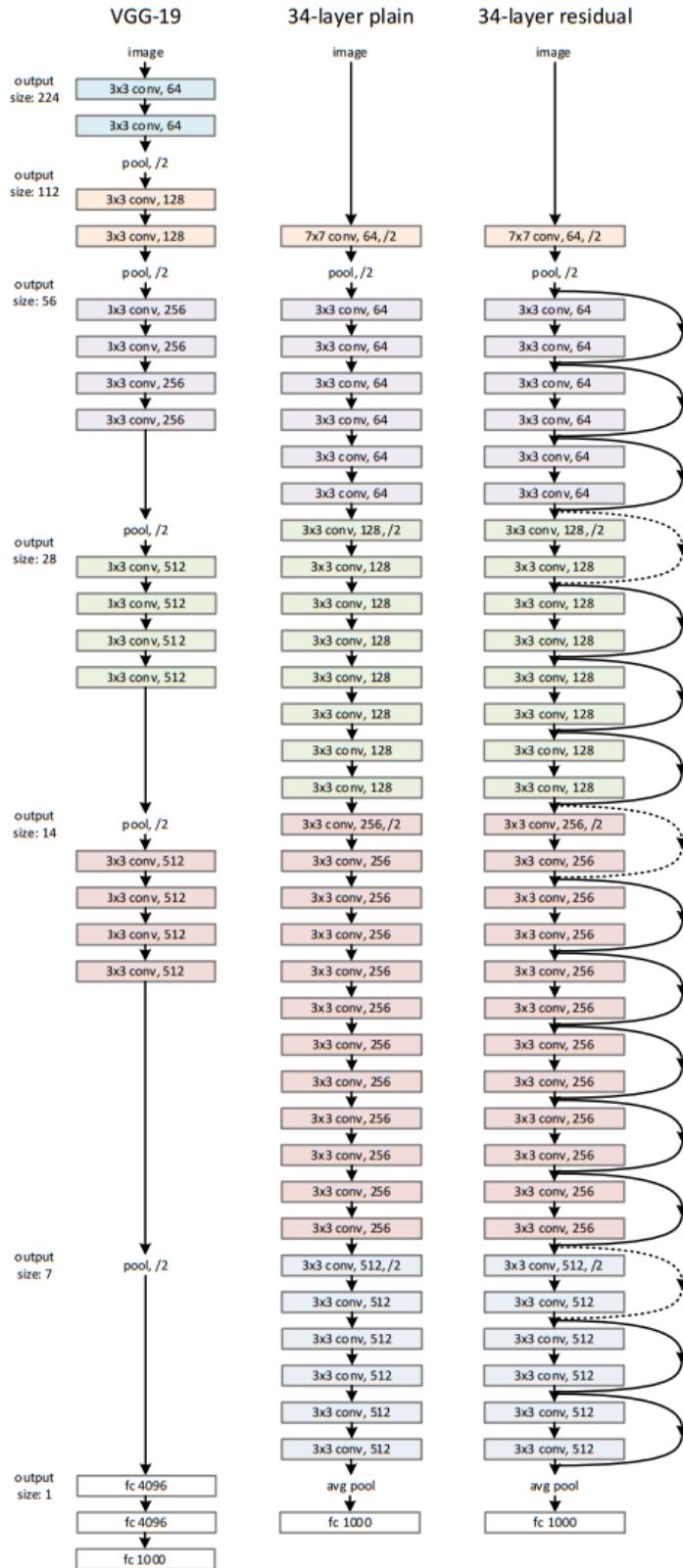


Figure 14: ResNet34 model architecture compared to VGG19 model architecture, from He *et al.* [42].

4.2.2 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. We also apply Grad-CAM with manual inspection to visualize the activation weights for incorrectly classified images to determine which features are interfering with the classification. Finally, we apply t-SNE to visualize the separation learned for the model to further analyze the classifications made by the model.

5 CHAPTER 5: EXPERIMENTAL RESULTS AND ANALYSIS

5.1 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. Anaconda is used to manage standard python packages, such as numpy, torch, and matplotlib [43]–[46]. Extra details about the server are shared in neofetch output in Figure 15.

```
[nix-shell:~]$ neofetch
      .-/+oossssoo+/-.
      `:+ssssssssssssssssssss+:'`+
      +ssssssssssssssssssssyssss+-+
      .osssssssssssssssssdMMMyssso.
      /ssssssssssshdmNNmmyNMMMMhssssss/
      +ssssssssshmydMMMMMMMddddysssssss+-
      /ssssssssshNMMMyhyyyyhmNMMMNhssssssss/
      .sssssssdMMMNhssssssssshNMMMdssssssss.
      +sssshhhyNMMNysssssssssssyNMMMyssssss+-
      ossyNMMMNyMMhssssssssssssshmmmhssssssso
      ossyNMMMNyMMhssssssssssssshmmmhssssssso
      +sssshhhyNMMNysssssssssssyNMMMyssssss+-
      .sssssssdMMMNhssssssssshNMMMdssssssss.
      /ssssssshNMMMyhyyyyhdNMMMNhssssssss/
      +sssssssssdmydMMMMMMMddddysssssss+-
      /sssssssssssshdNNNNmmyNMMMMhssssss/
      .osssssssssssssssssdMMMyssso.
      +ssssssssssssssssssyyssss+-+
      `:+ssssssssssssssssss+:'`+
      .-/+oossssoo+/-.

ngardn10@lambdaprdapp01
-----
OS: Ubuntu 18.04.6 LTS x86_64
Host: SYS-4029GP-TRT 123456789
Kernel: 5.4.0-97-lowlatency
Uptime: 6 days, 23 hours, 29 mins
Packages: 77451 (apt), 172 (nix-user)
Shell: bash 4.4.23
Resolution: 1024x768
Terminal: node
CPU: Intel Xeon Silver 4116 (48) @ 3.000GHz
GPU: NVIDIA GeForce RTX 2080 Ti
Memory: 15535MiB / 257591MiB
```



Figure 15: Neofetch output for the Lambda Labs GPU server used for experiments.

5.2 Results

5.2.1 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 5 and Figure 16. The results are collected on 8 iterations of training and averaged. The results are rounded to 2 decimal places where appropriate.

Table 1: VGG11 confusion matrix on ant head image dataset.

	True Rough	True Smooth
Predicted Rough	101.75	53
Predicted Smooth	37.375	257.875

Table 2: VGG13 confusion matrix on ant head image dataset.

	True Rough	True Smooth
Predicted Rough	95.25	39.375
Predicted Smooth	43.25	272.25

Table 3: VGG16 confusion matrix on ant head image dataset.

	True Rough	True Smooth
Predicted Rough	87.5	42.625
Predicted Smooth	47.375	272.5

Table 4: VGG19 confusion matrix on ant head image dataset.

	True Rough	True Smooth
Predicted Rough	91.25	42.125
Predicted Smooth	43.25	273.375

Table 5: 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

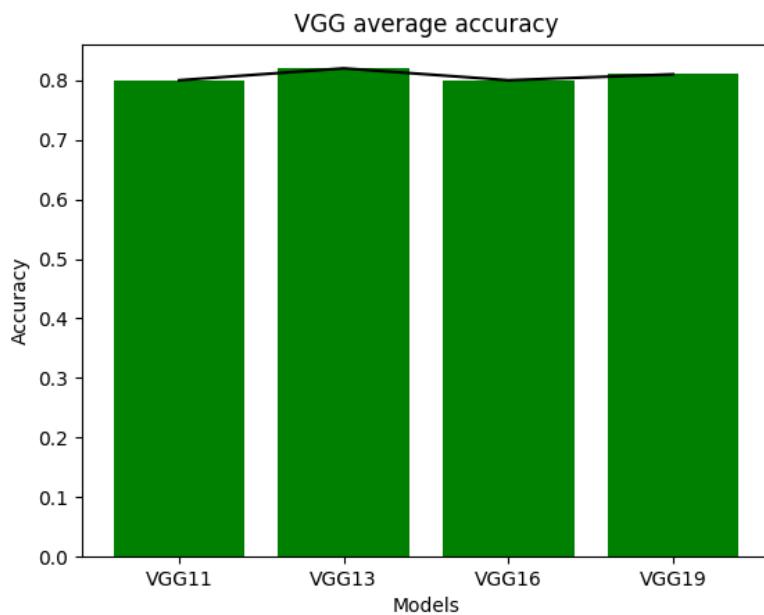


Figure 16: Average accuracy for VGG architectures on ant head image dataset.

5.2.2 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 9 and Figure 17. The results are collected on 8 iterations of training and averaged. The results are rounded to 2 decimal places where appropriate.

Table 6: ResNet18 confusion matrix on ant head image dataset.

	True Rough	True Smooth
Predicted Rough	98.75	48.75
Predicted Smooth	39.25	262

Table 7: ResNet50 confusion matrix on ant head image dataset.

	True Rough	True Smooth
Predicted Rough	81.375	43.625
Predicted Smooth	55.875	269.125

Table 8: ResNet101 confusion matrix on ant head image dataset.

	True Rough	True Smooth
Predicted Rough	85.75	53.875
Predicted Smooth	50.125	260.25

Table 9: 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

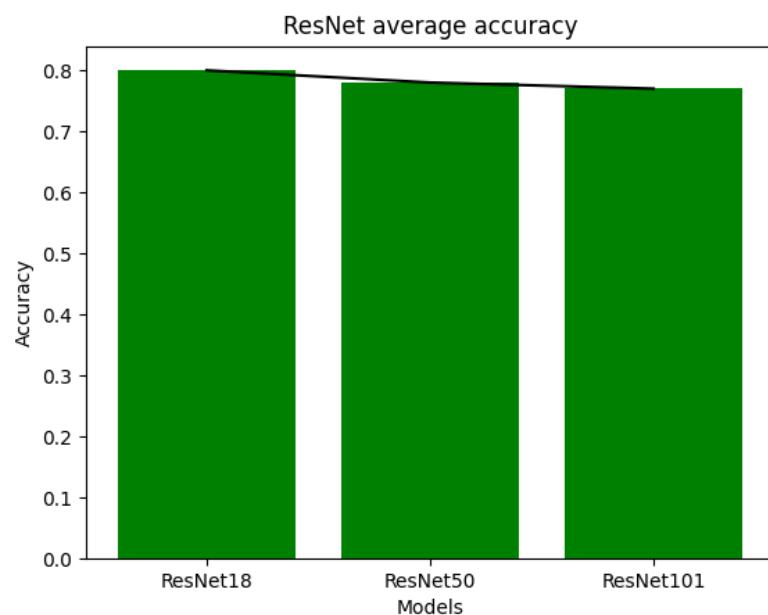


Figure 17: Average accuracy for ResNet architectures on ant head image dataset.

5.2.3 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 13 and Figure 18. The results are collected on 8 iterations of training and averaged. The results are rounded to 2 decimal places where appropriate.

Table 10: Fine-tuned ResNet18 confusion matrix on ant head image dataset.

	True Rough	True Smooth
Predicted Rough	117.375	35.125
Predicted Smooth	17.875	279.625

Table 11: Fine-tuned ResNet50 confusion matrix on ant head image dataset.

	True Rough	True Smooth
Predicted Rough	114.5	33.25
Predicted Smooth	24.125	278.125

Table 12: Fine-tuned ResNet101 confusion matrix on ant head image dataset.

	True Rough	True Smooth
Predicted Rough	117.25	33.25
Predicted Smooth	18.375	281.125

Table 13: 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

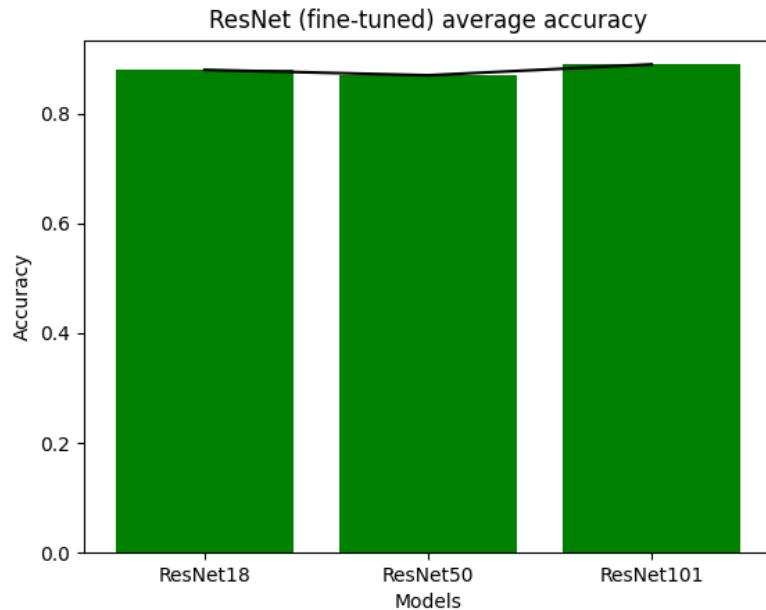


Figure 18: Average accuracy for fine-tuned ResNet architectures on ant head image dataset.

5.3 Analysis

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.

5.3.1 t-SNE Visualization



Figure 19: 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 19 shows the results of the trained randomly initialized model and Figure 20 shows the results of the fine-tuned model.

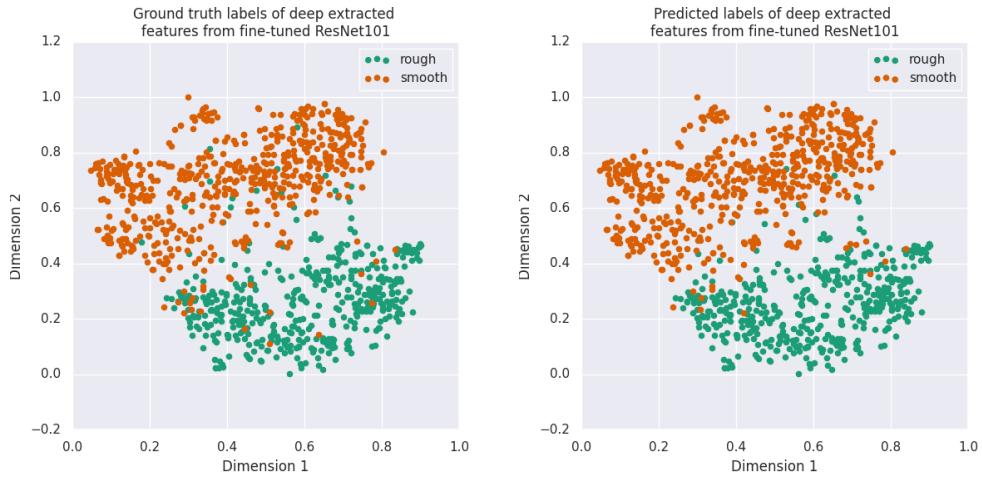


Figure 20: t-SNE visualization of the embeddings of the second to last layer of the fine-tuned ResNet101 model trained on ant head image dataset.

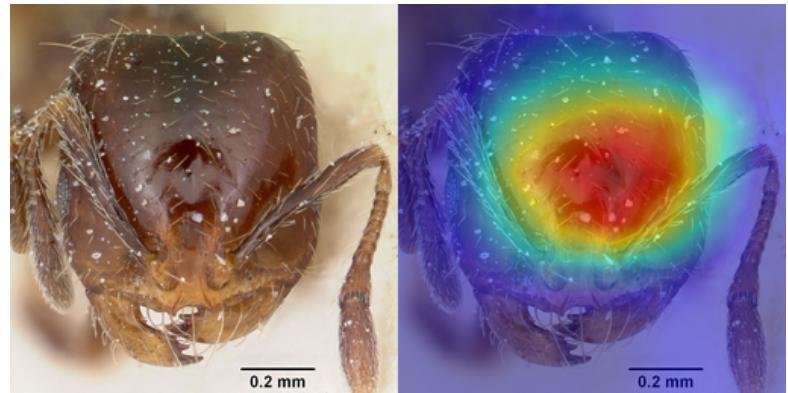
Based on the visual results of the t-SNE visualization, we can see that the fine-tuned model learned a stronger separation of the two classes, which reinforces the results that the fine-tuned model received a higher average accuracy.

5.3.2 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 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 21, 22, 23, and 24. 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 analysis. In essence, the model in this situation knows *where* to look, but not *what* to look for. In Figure 23a, 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 23c, 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.



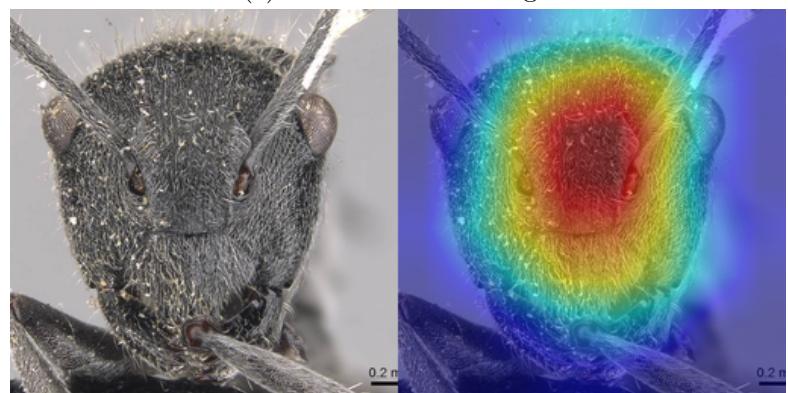
(a) Ground truth: smooth



(b) Ground truth: smooth

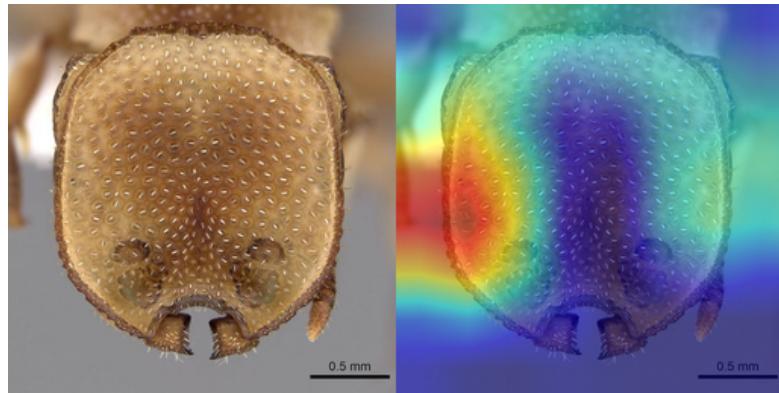


(c) Ground truth: rough

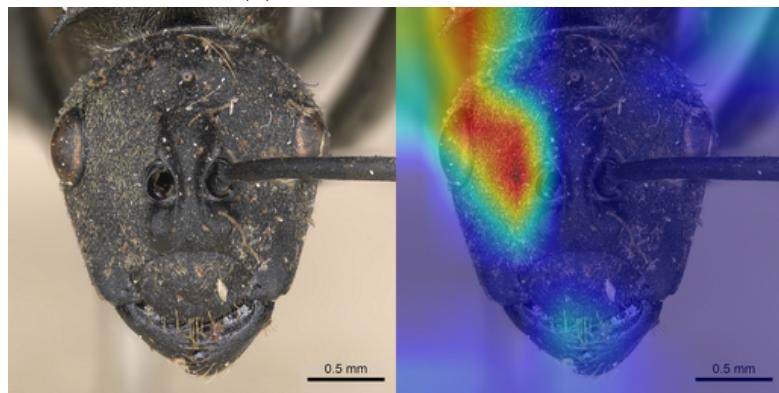


(d) Ground truth: rough

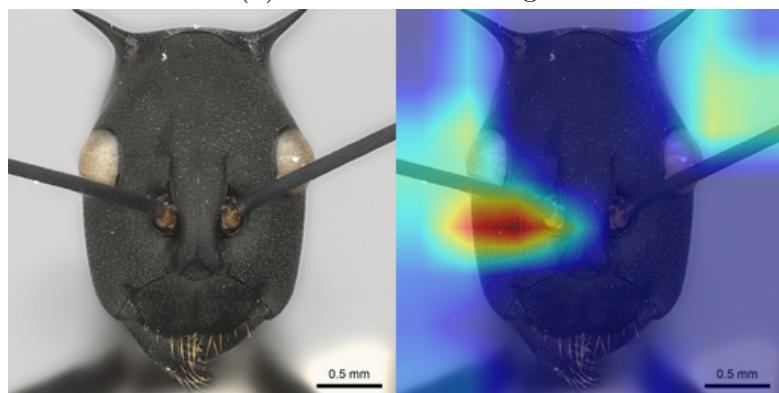
Figure 21: Correctly classified images using ideal features.



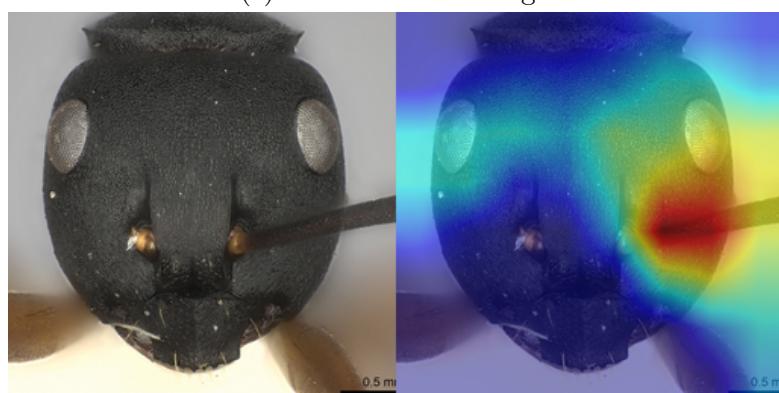
(a) Ground truth: rough



(b) Ground truth: rough



(c) Ground truth: rough

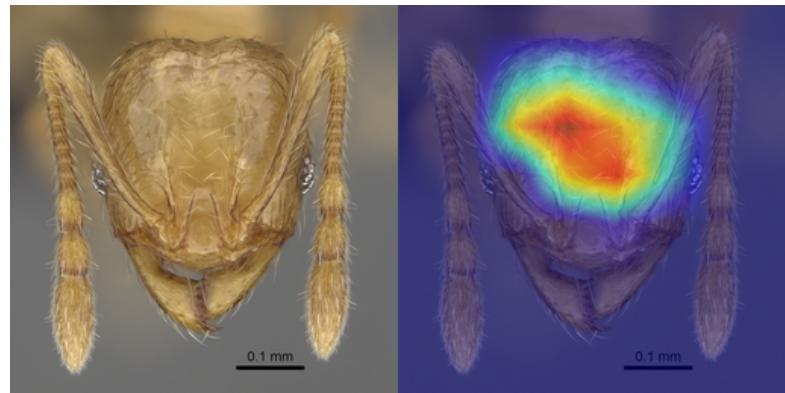


(d) Ground truth: smooth

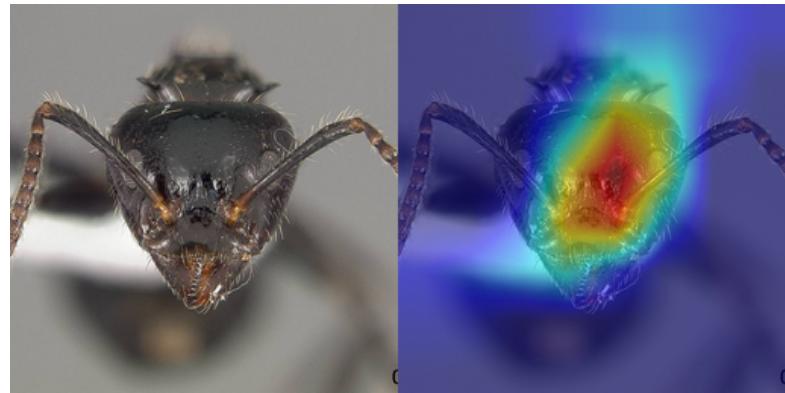
Figure 22: Correctly classified images using non-ideal features.



(a) Ground truth: rough



(b) Ground truth: smooth



(c) Ground truth: smooth



(d) Ground truth: smooth

Figure 23: Incorrectly classified images using ideal features.



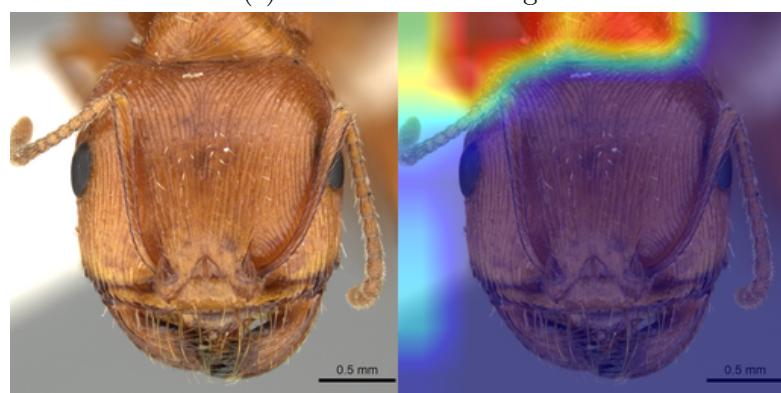
(a) Ground truth: rough



(b) Ground truth: rough



(c) Ground truth: rough



(d) Ground truth: rough

Figure 24: Incorrectly classified images using non-ideal features.

6 CHAPTER 6: CONCLUSION

6.1 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 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 GitHub².

6.2 Future Work

Since we analyze the results of the VGG model only with randomly initialized weights, future work should also include the results of fine-tuned VGG models to compare to the fine-tuned ResNet models. Additionally, other texture analysis methods which are trained on texture datasets should be added to the comparison. Future work will have a custom algorithm for solving the problem of ant cuticle texture categorization. In addition, future work should include

²<https://github.com/ngngardner/cuticulus>

some data augmentation method to deal with the imbalanced dataset problem.

REFERENCES

- [1] E. Tihelka, C. Cai, M. Giacomelli, J. Lozano-Fernandez, O. Rota-Stabelli, D. Huang, M. S. Engel, P. C. Donoghue, and D. Pisani, “The evolution of insect biodiversity,” *Current Biology*, vol. 31, no. 19, R1299–R1311, 2021, ISSN: 0960-9822. DOI: <https://doi.org/10.1016/j.cub.2021.08.057>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0960982221011933>.
- [2] P. J Gullan and P. S Cranston, *Insects: an Outline of Entomology*. English. Hoboken: Wiley, 2009, OCLC: 1048427241, ISBN: 9781405144575. [Online]. Available: <http://qut.eblib.com.au/patron/FullRecord.aspx?p=233169>.
- [3] M. Berenbaum, *Bugs In The System: Insects And Their Impact On Human Affairs*, en. Basic Books, Jun. 1996, ISBN: 9780465024452.
- [4] S. N. A. Hassan, N. S. A. Rahman, Z. Z. Htike, and S. L. Win, “Advances in Automatic Insect Classification,” *Electrical and Electronics Engineering: An International Journal*, vol. 3, no. 2, pp. 51–63, May 2014, ISSN: 22005846. DOI: [10.14810/elelij.2014.3204](https://doi.org/10.14810/elelij.2014.3204). [Online]. Available: <http://www.wireilla.com/engg/eeeij/papers/3214elelij04.pdf>.
- [5] D. Xia, P. Chen, B. Wang, J. Zhang, and C. Xie, “Insect Detection and Classification Based on an Improved Convolutional Neural Network,” *Sensors (Basel, Switzerland)*, vol. 18, no. 12, p. 4169, Nov. 2018, ISSN:

- 1424-8220. DOI: [10.3390/s18124169](https://doi.org/10.3390/s18124169). [Online]. Available: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6308804/>.
- [6] S. Kaloudis, D. Anastopoulos, C. P. Yialouris, N. A. Lorentzos, and A. B. Sideridis, “Insect identification expert system for forest protection,” en, *Expert Systems with Applications*, vol. 28, no. 3, pp. 445–452, Apr. 2005, ISSN: 0957-4174. DOI: [10.1016/j.eswa.2004.12.005](https://doi.org/10.1016/j.eswa.2004.12.005). [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0957417404001496>.
- [7] K. Thenmozhi and U. Srinivasulu Reddy, “Crop pest classification based on deep convolutional neural network and transfer learning,” en, *Computers and Electronics in Agriculture*, vol. 164, p. 104906, Sep. 2019, ISSN: 0168-1699. DOI: [10.1016/j.compag.2019.104906](https://doi.org/10.1016/j.compag.2019.104906). [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0168169919310695>.
- [8] M. Martineau, D. Conte, R. Raveaux, I. Arnault, D. Munier, and G. Venturini, “A survey on image-based insect classification,” *Pattern Recognit.*, vol. 65, pp. 273–284, 2017.
- [9] C. E. Sosiak and P. Barden, “Multidimensional trait morphology predicts ecology across ant lineages,” en, *Functional Ecology*, vol. 35, no. 1, T. Higham, Ed., pp. 139–152, Jan. 2021, ISSN: 0269-8463, 1365-2435. DOI: [10.1111/1365-2435.13697](https://doi.org/10.1111/1365-2435.13697). [Online]. Available: <https://onlinelibrary.wiley.com/doi/10.1111/1365-2435.13697>.
- [10] 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), ISSN: 1051-4651, Aug. 2018, pp. 1530–1535. DOI: [10.1109/ICPR.2018.8545798](https://doi.org/10.1109/ICPR.2018.8545798).

- [11] V. Perrichot and B. Fisher, “AntWeb: Digitizing Recent and fossil insects for an online database of the ants of the world,” en, in *Digital Fossil International Conference*, Sep. 2012. [Online]. Available: <https://hal-insu.archives-ouvertes.fr/insu-00805243>.
- [12] M. J. Pongsiri and J. Roman, “Examining the Links between Biodiversity and Human Health: An Interdisciplinary Research Initiative at the U.S. Environmental Protection Agency,” *Ecohealth*, vol. 4, no. 1, pp. 82–85, 2007, ISSN: 1612-9202. DOI: [10.1007/s10393-007-0087-3](https://doi.org/10.1007/s10393-007-0087-3). [Online]. Available: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7087626/>.
- [13] S. P. Cooper, J. A. Finlay, G. Cone, M. E. Callow, J. A. Callow, and A. B. Brennan, “Engineered antifouling microtopographies: Kinetic analysis of the attachment of zoospores of the green alga Ulva to silicone elastomers,” *Biofouling*, vol. 27, no. 8, pp. 881–892, Sep. 2011, ISSN: 0892-7014. DOI: [10.1080/08927014.2011.611305](https://doi.org/10.1080/08927014.2011.611305). [Online]. Available: <https://doi.org/10.1080/08927014.2011.611305>.
- [14] E. E. Mann, D. Manna, M. R. Mettetal, R. M. May, E. M. Dannemiller, K. K. Chung, A. B. Brennan, and S. T. Reddy, “Surface micropattern limits bacterial contamination,” eng, *Antimicrobial Resistance and Infection Control*, vol. 3, p. 28, 2014, ISSN: 2047-2994. DOI: [10.1186/2047-2994-3-28](https://doi.org/10.1186/2047-2994-3-28).
- [15] James T. Buxton, Kylie A. Robert, Alan T. Marshall, Travis L. Dutka, and Heloise Gibb, “A cross-species test of the function of cuticular traits in

- ants (Hymenoptera: Formicidae)," 2021. DOI: [10.25849/MYRMECOL.NEWS_031:031](https://doi.org/10.25849/MYRMECOL.NEWS_031:031). [Online]. Available: https://myrmecologicalnews.org/cms/index.php?option=com_content&view=category&id=1577&Itemid=439.
- [16] 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*, ISSN: 1063-6919, Jun. 2014, pp. 3606–3613. DOI: [10.1109/CVPR.2014.461](https://doi.org/10.1109/CVPR.2014.461).
- [17] T. Reed and J. Dubuf, "A Review of Recent Texture Segmentation and Feature Extraction Techniques," en, *CVGIP: Image Understanding*, vol. 57, no. 3, pp. 359–372, May 1993, ISSN: 10499660. DOI: [10.1006/ciun.1993.1024](https://doi.org/10.1006/ciun.1993.1024). [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S1049966083710247>.
- [18] P. Maillard, "Comparing texture analysis methods through classification," *Photogrammetric Engineering and Remote Sensing*, vol. 69, pp. 357–367, 2003.
- [19] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern classification*, 2nd ed. New York: Wiley, 2001, ISBN: 9780471056690.
- [20] I. Arel, D. C. Rose, and T. P. Karnowski, "Deep Machine Learning - A New Frontier in Artificial Intelligence Research [Research Frontier]," *IEEE Computational Intelligence Magazine*, vol. 5, no. 4, pp. 13–18, Nov. 2010, ISSN: 1556-6048. DOI: [10.1109/MCI.2010.938364](https://doi.org/10.1109/MCI.2010.938364).

- [21] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998, ISSN: 1558-2256. DOI: [10.1109/5.726791](https://doi.org/10.1109/5.726791).
- [22] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, “Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization,” en, *International Journal of Computer Vision*, vol. 128, no. 2, pp. 336–359, Feb. 2020, ISSN: 0920-5691, 1573-1405. DOI: [10.1007/s11263-019-01228-7](https://doi.org/10.1007/s11263-019-01228-7). [Online]. Available: <http://link.springer.com/10.1007/s11263-019-01228-7>.
- [23] Q.-s. Zhang and S.-c. Zhu, “Visual interpretability for deep learning: A survey,” en, *Frontiers of Information Technology & Electronic Engineering*, vol. 19, no. 1, pp. 27–39, Jan. 2018, ISSN: 2095-9230. DOI: [10.1631/FITEE.1700808](https://doi.org/10.1631/FITEE.1700808). [Online]. Available: <https://doi.org/10.1631/FITEE.1700808>.
- [24] L. van der Maaten and G. E. Hinton, “Visualizing data using t-sne,” *Journal of Machine Learning Research*, vol. 9, pp. 2579–2605, 2008.
- [25] L. Deng, “The MNIST Database of Handwritten Digit Images for Machine Learning Research [Best of the Web],” *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 141–142, Nov. 2012, ISSN: 1558-0792. DOI: [10.1109/MSP.2012.2211477](https://doi.org/10.1109/MSP.2012.2211477).
- [26] 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*, vol. 7,

pp. 45 301–45 312, 2019, ISSN: 2169-3536. DOI:

[10.1109/ACCESS.2019.2909522](https://doi.org/10.1109/ACCESS.2019.2909522).

- [27] T. Kasinathan and S. R. Uyyala, “Machine learning ensemble with image processing for pest identification and classification in field crops,” en, *Neural Computing and Applications*, vol. 33, no. 13, pp. 7491–7504, Jul. 2021, ISSN: 1433-3058. DOI: [10.1007/s00521-020-05497-z](https://doi.org/10.1007/s00521-020-05497-z). [Online]. Available: <https://doi.org/10.1007/s00521-020-05497-z>.
- [28] 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*, ISSN: 2381-8549, Sep. 2013, pp. 2577–2581. DOI: [10.1109/ICIP.2013.6738531](https://doi.org/10.1109/ICIP.2013.6738531).
- [29] C. C. Gotlieb and H. E. Kreyszig, “Texture descriptors based on co-occurrence matrices,” en, *Computer Vision, Graphics, and Image Processing*, vol. 51, no. 1, pp. 70–86, Jul. 1990, ISSN: 0734189X. DOI: [10.1016/S0734-189X\(05\)80063-5](https://doi.org/10.1016/S0734-189X(05)80063-5). [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S0734189X05800635>.
- [30] D. G. Lowe, “Distinctive Image Features from Scale-Invariant Keypoints,” en, *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, Nov. 2004, ISSN: 0920-5691. DOI: [10.1023/B:VISI.0000029664.99615.94](https://doi.org/10.1023/B:VISI.0000029664.99615.94). [Online]. Available: <http://link.springer.com/10.1023/B:VISI.0000029664.99615.94>.
- [31] 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), Feb. 2016, pp. 125–129. DOI: [10.1109/CONIELECOMP.2016.7438563](https://doi.org/10.1109/CONIELECOMP.2016.7438563).
- [32] J. A. Glick and K. Miller, *Insect Classification With Heirarchical Deep Convolutional Neural Networks* *Convolutional Neural Networks for Visual Recognition (CS 231 N)*, en, 2016. [Online]. Available: <https://www.semanticscholar.org/paper/Insect-Classification-With-Heirarchical-Deep-Neural-Glick-Miller/bb8316bb841bf6667431bf755f6fe01ec013b8d5>.
- [33] 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)*, Nov. 2017, pp. 210–215. DOI: [10.1109/ICCSCE.2017.8284406](https://doi.org/10.1109/ICCSCE.2017.8284406).
- [34] 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*, ISSN: 1063-6919, Jun. 2009, pp. 248–255. DOI: [10.1109/CVPR.2009.5206848](https://doi.org/10.1109/CVPR.2009.5206848).
- [35] C. Guestrin and E. Fox, *Text: Imagenet 1000 class idx to human readable labels*, en. [Online]. Available: <https://gist.github.com/yrevar/942d3a0ac09ec9e5eb3a>.
- [36] V. Chandra and D. J. C. Kronauer, “Foraging and feeding are independently regulated by social and personal hunger in the clonal raider ant,” en, *Behavioral Ecology and Sociobiology*, vol. 75, no. 2, p. 41, Jan. 2021, ISSN: 1432-0762. DOI: [10.1007/s00265-021-02985-7](https://doi.org/10.1007/s00265-021-02985-7). [Online]. Available: <https://doi.org/10.1007/s00265-021-02985-7>.

- [37] I. Fetter-Pruneda, T. Hart, Y. Ulrich, A. Gal, P. R. Oxley, L. Olivos-Cisneros, M. S. Ebert, M. A. Kazmi, J. L. Garrison, C. I. Bargmann, and D. J. C. Kronauer, “An oxytocin/vasopressin-related neuropeptide modulates social foraging behavior in the clonal raider ant,” en, *PLOS Biology*, vol. 19, no. 6, e3001305, Jun. 2021, ISSN: 1545-7885. DOI: [10.1371/journal.pbio.3001305](https://doi.org/10.1371/journal.pbio.3001305). [Online]. Available: <https://journals.plos.org/plosbiology/article?id=10.1371/journal.pbio.3001305>.
- [38] A. Sclocco, S. J. Y. Ong, S. Y. P. Aung, and S. Teseo, “Integrating real-time data analysis into automatic tracking of social insect behavior,” en, bioRxiv, Tech. Rep., Nov. 2020, Type: article, p. 2020.11.03.366195. [Online]. Available: <https://www.biorxiv.org/content/10.1101/2020.11.03.366195v1>.
- [39] X. Cao, S. Guo, J. Lin, W. Zhang, and M. Liao, “Online tracking of ants based on deep association metrics: Method, dataset and evaluation,” en, *Pattern Recognition*, vol. 103, p. 107233, Jul. 2020, ISSN: 00313203. DOI: [10.1016/j.patcog.2020.107233](https://doi.org/10.1016/j.patcog.2020.107233). [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S003132032030039X>.
- [40] A. Gal, J. Saragosti, and D. J. Kronauer, “anTraX, a software package for high-throughput video tracking of color-tagged insects,” eng, *eLife*, vol. 9, e58145, Nov. 2020, ISSN: 2050-084X. DOI: [10.7554/eLife.58145](https://doi.org/10.7554/eLife.58145).
- [41] K. Simonyan and A. Zisserman, “Very Deep Convolutional Networks for Large-Scale Image Recognition,” *arXiv:1409.1556 [cs]*, Apr. 2015, arXiv: 1409.1556. [Online]. Available: [http://arxiv.org/abs/1409.1556](https://arxiv.org/abs/1409.1556).

- [42] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” *arXiv:1512.03385 [cs]*, Dec. 2015, arXiv: 1512.03385. [Online]. Available: <http://arxiv.org/abs/1512.03385>.
- [43] *Anaconda software distribution*, version Vers. 2-2.4.0, 2020. [Online]. Available: <https://docs.anaconda.com/>.
- [44] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T. E. Oliphant, “Array programming with NumPy,” *Nature*, vol. 585, no. 7825, pp. 357–362, Sep. 2020. doi: [10.1038/s41586-020-2649-2](https://doi.org/10.1038/s41586-020-2649-2). [Online]. Available: <https://doi.org/10.1038/s41586-020-2649-2>.
- [45] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, “Pytorch: An imperative style, high-performance deep learning library,” in *Advances in Neural Information Processing Systems 32*, Curran Associates, Inc., 2019, pp. 8024–8035. [Online]. Available: <http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf>.

- [46] J. D. Hunter, “Matplotlib: A 2d graphics environment,” *Computing in Science & Engineering*, vol. 9, no. 3, pp. 90–95, 2007. doi: [10.1109/MCSE.2007.55](https://doi.org/10.1109/MCSE.2007.55).