The Pennsylvania State University

The Graduate School

# AN ANALYSIS AND FRAMEWORK FOR MULTI-LABEL IMAGE CLASSIFICATION OF INSECT TAXONOMY WITH CONVOLUTIONAL NEURAL NETWORKS

A Thesis in

Computer Science and Engineering

by

Viet Pham



Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Master of Science

May 2022

The thesis of Viet Pham was reviewed and approved by the following:

Vijaykrishnan Narayanan
Distinguished Professor of Computer Science and Engineering
Thesis Advisor

John Morgan Sampson
Assistant Professor of Computer Science and Engineering

Chitaranjan Das
Distinguished Professor of Computer Science and Engineering
Head of the Department of Computer Science and Engineering

# ABSTRACT

Multi-label image classification is the task of assigning multiple labels to an image. As of current, there are only a few bodies of work in this specific task and its application to classifying insects and their taxonomy, and therefore, this is the task that this body of work explored and contributed to. Furthermore, for many use cases, the multiple labels of an object are not unrelated, but they have a dependent relation with each other, such as the hierarchy of insect taxonomy. Since others that have worked on this task have not explored in-depth the possible relations among the labels of an object and how a model might make use of them, this body of work also sought to provide insight into this inquiry.

The approach was to first build a framework with a pipeline that will facilitate this research process, of which involves data collection and preparation, model training, and model evaluation. Hence, this work was able to develop a framework that contains code to support all three steps, in which images of desired insects can be automatically downloaded, structured, and correctly labeled for model training and evaluation with the desired model and test image set. This framework was then used for evaluating multiple models, including a single-label image classifier as a baseline, with an f-beta metric, and with comparisons on classifications of image frames from video footage that were manually labeled by expertise. Conclusively, this work found that the multi-label image classifiers were able to achieve f-beta scores greater than 95%, and that for some of the footage, the model was able to obtain the correct results with a percentage of 24.4% in

the order level and of 23.1% in the family level. Lastly, in addition to the order percentage accuracy score on the footage being higher than the family percentage score, the model was able to classify most insect families that were in the same order that it has trained on but in a different family that it has not seen correctly. With these results, there is a lot of potential to use the developed framework and generated insights to run more experiments with better models and larger datasets, obtaining stronger answers for the research questions proposed.

# TABLE OF CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

# ACKNOWLEDGEMENTS

This work was supported in part by DARPA/SRC JUMP program. The findings and conclusions are only of the researchers and are not necessarily those of the sponsors. I would like express my utmost thanks and respect to my advisor, Dr. Vijaykrishnan Narayanan, for being a great advisor, mentor, and leader of the lab as well as the insect monitoring project this work stems from. Likewise, I would also like to thank my general graduate advisor, Dr. John Sampson, for all of his support through my career here at the university, whether it was general class advice or explanations regarding complex technical concepts. Additionally, I would also like to thank Ph.D. students, Eric Homan, Codey Mathis, Nelson Daniel Troncoso, and Chonghan Lee for being great team members, lab members, and friends. Overall, I would like to the University in general, as well as the computer science and engineering department, for being a great educational and holistic environment that profoundly enabled and encouraged my development as a student and person.

# Chapter 1

# Introduction

## 1.1 Motivation

Image classification is a machine learning task that is useful for many applications. It is popularly done in applications where images need to be classified with a single label, as in classifying images of animals with their respective common name (e.g., an image that contains a cat is labeled as a cat). However, some applications require multiple labels for a single image, rather than a single label, since the image can contain multiple objects (e.g., an image that contains a cat and a dog has labels cat and dog). This task is generally known as multi-label image classification, and as shown in Figure 1.1, it has been done with the combination of many techniques, such as object detection and instance segmentation, in order to localize and distinguish the different objects, and an image classifier to finally label each object. However, not only can an image have multiple objects that need to be classified, but each object can have multiple labels to be assigned to them. This task of assigning multiple labels to each object in a single image will be denoted in this body of work as multi-label/per-object image classification. Nonetheless, multi-label/per-object image classification can simply be done by performing general multi-label image classification repeatedly on individual cropped bounding boxes for each object in an image. Hence, this work focuses on performing

multi-label image classification on images that contain a singular object, since improving this underlying mechanism will directly contribute to the more extensive ability of applying multiple labels to multiple objects in an image.

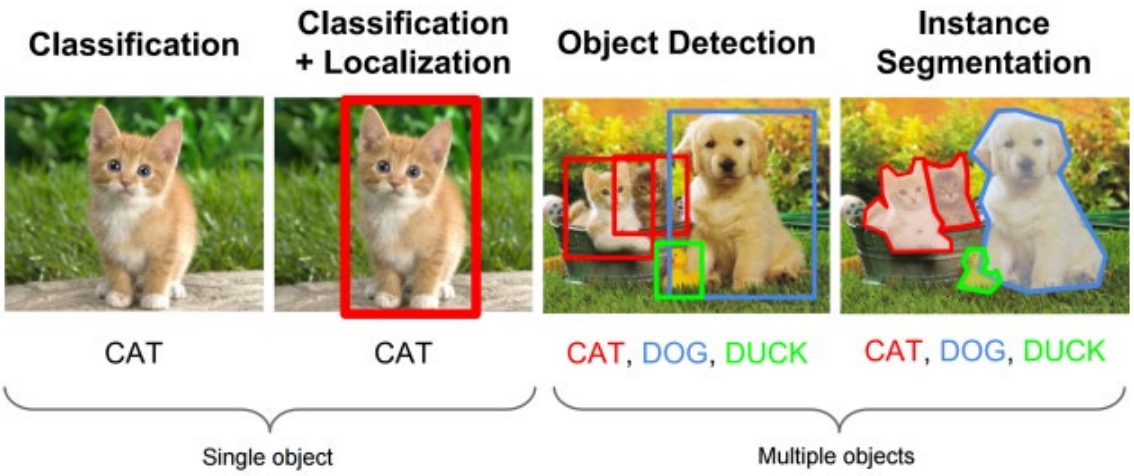


Figure 1.1: Single-label vs Multi-label Image Classification [12]

The significance of multi-label/per-object image classification comes from the notion that an object can have multiple labels because different pieces and types of information can be associated to define it. For example, a cat can have information regarding not only its descriptive features, such as “orange,” “spotted,” or “big,” but also of what it eats, such as “fish” or “cat food.” More importantly, these different types and pieces of information can have some sort of relation with one another, in which one piece of information will affect other pieces of information associated with it. One example of this notion is when information about an object is structured in a hierarchy, as in a car’s make and model, since the model of a car is dependent on what make it is. Another example in which objects can have this structured hierarchical relation, is insect taxonomy, in which a species, such as Danaus Plexippus, is a subgroup of the genus,

Danaus (i.e., Monarch butterfly), and the genus is a subgroup of the family, Nymphalidae. This is shown more generally in Figure 2.

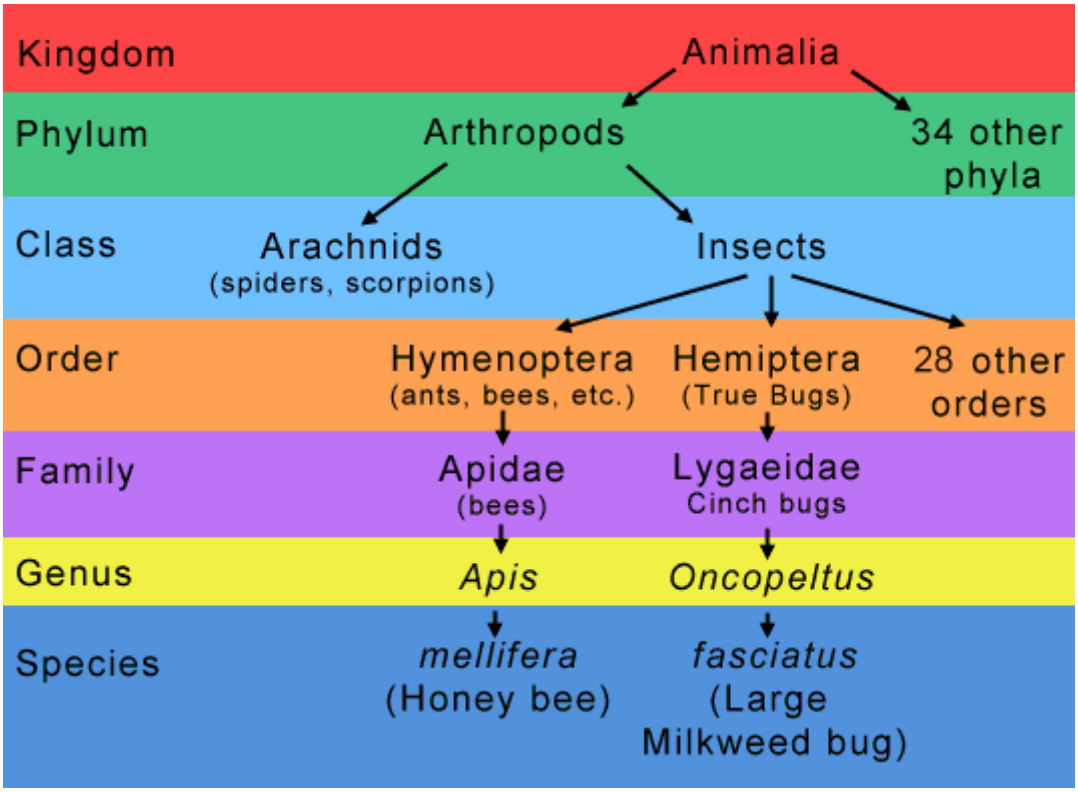


Figure 1.2: Chart showing taxonomy of insects and where they fit in the kingdom Animalia [13]

With this inherent hierarchy within insect taxonomy, taxonomic insect image classification is an excellent example application to explore the task of multi-label image classification, and thereby was the application this body of work directly contributed to. The efforts unraveling this inquiry provided answers, or at least great insight, for many meaningful research questions, such as what models perform best for this task and what types and amount of data in a dataset is optimal, and much more. Additionally,

researching this task discovered other important and deeper notions, such as if a model is able to learn or manifest any structural relations between the multiple labels of a classification. For example, does it understand, to some degree, that an insect with a label of the bee family cannot have labels for any genus of the ant family. This very notion of a structural relation among the labels, such as a hierarchy, can be extrapolated beyond insect taxonomy, and thereby has potential to be a useful feature to be integrated into machine learning models for improving performance. Furthermore, this task applied to the specific application of insect taxonomy is not worked on as much as it could be, as there are no general large dataset, evaluation benchmark, or framework for it that could be of great value for researchers and enthusiasts who can greatly leverage this technology for many applications such as insect monitoring or similar. Lastly, aside from the technical knowledge and contribution that is gathered from the results of this research endeavor, there is also an impact and contribution on a high-level to the general problem of insect monitoring. Ultimately, this work not only sought to advance and contribute to multi-label image classification for insect taxonomy, but also to build and improve upon the previous related works of others within the same domain.

## 1.2 Related Works

Regarding general insect classification with computer vision, there are many papers on this topic. One piece of work that is relevant to this work and more recent is from [1], in which the authors were able to classify beetles at the species level with a convolutional neural network (CNN). They obtained decent results, as they classified 51.9% of their 19,165 test images correctly at the species level and 74.9% at the genus level. However, their results at the genus level were extracted from the classifications that only contained a single label denoting the species, rather than having multiple labels assigned to each classification denoting both the genus and species. Another more recent and notable piece of work is from [2]. In this work, the authors were able to use many machine learning models and algorithms, such as support vector machines, k-nearest neighbors, and a CNN to classify insect pests. As a result, they were successfully able to obtain a rate of 91.5% and 90% for 9 and 24 insect classes, respectively. However, similar to the previous work, each insect was classified with a single label.

As of current, the only known notable work that attempted to explore in-depth the classification of insect taxonomy specifically is from [3]. The authors of this work also used a CNN, and in particular the VGG16 CNN architecture, with transfer learning for two goals. The first of which is to see whether a model can learn to classify an insect of an unknown subgroup (e.g., a particular fly species) to a higher group (e.g., the fly family), given that the model trained on other subgroups (e.g., other fly species). The second goal was to test the model's performance on classifying insects that are closely related taxonomically and hence, very similar in appearance. For both goals, the authors

were able to obtain over 92% and 97% accuracy, respectively, but these results were obtained from datasets that were relatively small and the images were very controlled. For example, the largest dataset had 3845 images, and each dataset had images that were closeups of a particular body part or were insects placed on an artificial background, rather than entire insects with varying positions, angles, and natural backgrounds. Hence, it is not certain how this successful this model would perform for general insect monitoring tasks. Additionally, although this work provided insight into the model's learning and understanding of insect taxonomy, it is still not multi-label insect classification.

With regards to multi-label classification in general, there are many works in this domain. To start off, the two bodies of work that are more relevant and recent are [4] and [5], both of which performed multi-label image classification on satellite images to describe the scenery. Both used CNNs and were able to successfully obtain high accuracy scores. However, although these two bodies of work performed multi-label image classification, they did not necessarily do the task of multi-label/per-object image classification since the classification of a satellite image had multiple labels describing the whole image holistically. Nonetheless, their technical approach of using a CNN and fine-tuning it for multiple labels was quite useful and leveraged for the work done in this research.

## 1.3 Thesis Organization

As introduced in the previous sections of this chapter, this body of work explored and contributed to the task of multi-label image classification on insect taxonomy. The following chapters and section will describe the process, concepts, and technical details that were involved in this research inquiry. In particular, chapter 2 contains the background information, laying out a more detailed explanation for each of the concepts that one needs to be familiar with to best receive and understand this body of work. Chapter 3 contains and describes the methods used in the process to obtain the results and analysis. it describes the developed framework that automated and facilitated much of the steps involved, including the data processes, model selection and training, and also prediction. Additionally, it also describes the evaluation methods, such as the f-beta score and the comparison with the manually labeled video footage, and how these methods indicate the performance of the model for multi-label image classification on insect taxonomy. Chapter 4 shows the results obtained from the evaluations of the models in both methods. Moreover, the results also include visualizations that will supplement the results, such as the training graphs. After showcasing the results, chapter 5 is a discussion that goes in-depth on what the results imply to answer the research questions. Finally, chapter 6 will conclude this body of work and discuss the contributions and their impact, as well as any future work that could potentially be explored based on the results and insight found.

## 1.4 Contributions

Overall, there are three main contributions this body of work provided. The first of which is a framework for multi-label image classification of insect taxonomy. This contribution is a more concrete and useful product from this body of work, as it is a codebase that automates and facilitates much of the processes involved to perform the task. Secondly, this body of work also provided results and insight showing and comparing which how well certain models perform for classifying insect taxonomy with multiple labels. Lastly, this body of work analyzed the model and its multi-label classification more in-depth, providing insight on model's understanding and learning of the hierarchical structure of the multiple labels classifying insect taxonomy.

# Chapter 2

# Background

## 2.1 Insect Monitoring

### 2.1.1 Insect Biodiversity

Insect biodiversity is on a decline [6] due to a variety of sources, including habitat lost, pesticides, and also climate change. As a result, this trend is impacting many insect groups that are beneficial to the environment. Therefore, many entomologists and environmental researchers are establishing great efforts to combat this issue. One of the great methods that has helped is insect monitoring, which allows researchers to analyze and understand insect populations and biodiversity along with their influence on the environment.

### 2.1.2 InsectEye Project

One of the groups applying insect monitoring is the collaboration between a team of researchers in the computer science department and team of researchers in the entomology department at Penn State. This effort is seeking to develop a non-lethal insect trap for biodiversity monitoring using cameras. With the non-lethal aspect as one of the core features for this device, software utilizing AI to perform computer vision is a

significant task to complete. Part of completing this involves performing multi-label image classification on insect taxonomy. By doing so, the goal is to enable the cameras to monitor and identify insects crawling through the trap. This work is the embodiment of that portion for the InsectEye project and it documents the current status and contributions so far.

### 2.1.3 Insect Taxonomy

To identify insects, entomologists and researchers alike classify each type into a hierarchy of classifications that are part of the animal kingdom. This hierarchy consists of multiple levels, starting from the Animal kingdom, then phylum, class, order, family, genus, and finally species. Each level in the taxonomy is often referred technically as a taxon. Moreover, each level deeper in the hierarchy represents a more specific classification of the animal, of which are a subgroup of the levels above, and are mutually exclusive and differing to other classification groups within the same level. That is, for example, an animal with a genus classification has all the distinguishing properties of that classification that separates it from other groups within the same genus level, but makes it part of and have the properties of the parent levels above it such as the family, order, and so forth. With regards to the InsectEye project and this body of work, the multiple labels denote the classifications of an insect at each level. For example, the monarch butterfly has classification labels, Lepidoptera in the order taxon, and Nymphalidae in the family taxon.

## 2.2 Multi-label Image Classification

To classify an image of an insect with multiple labels in computer vision, the problem can be formatted as a multi-label image classification task. Multi-label image classification has been done for numerous applications, such as describing satellite images and simply labeling an image containing objects. It works by training a neural network model, which is usually a CNN, with large amounts of image data, during which the model learns the features of the objects. By learning the features of the objects, and with an image dataset containing images that are labeled, a model can perform algorithms such as stochastic gradient descent and backpropagation to optimize its parameters for the task of classification. This is generally how all image classification tasks are done with neural networks. The difference with multi-label image classification, as the name implies, is that each image in the dataset have multiple labels. Regarding implementation, this difference in the dataset does not differ the mechanisms of training a model by that much. In fact, one can simply change the metric used to evaluate the model's performance to weigh the scoring factors differently, and also modify some of the output layers with transfer learning. All of the implementation details specific to this work will be explained in the Methods section.

## 2.3 Convolutional Neural Networks

Convolutional neural networks is a special type of neural network models that perform convolutions many times, along with other processes such as pooling, to find distinguishing features of a interested object within an image. For example, one layer can be used to detect edges, while other layers can be used to detect even more granular and specific details, such as a distinguishing alignment of pixels for a particular object. There are many works done on the architecture of this model to improve its performance, such as VGG16, MobileNetV2, and ResNet. Most of these models can also come pretrained on a very large image dataset, for which they can be conveniently imported and modified for a particular task such as multi-label image classification.

# Chapter 3

# Methods

## 3.1 Data Processes

The first and one of the most important steps to perform any machine learning task is to collect, organize, and prepare the data for training and evaluation. However, as of current, there is not any ideal dataset that has been established and organized for the task of multi-label image classification on insect taxonomy. As described in the previous section on the related works, the datasets involved had either single-labeled data or were relatively small, both in the total amount of samples for a species and the total amount of different species available. Additionally, the images themselves did not have a sufficient variety of angles, position of insects, or backgrounds, as most of the insect images were from a lab environment rather than natural environments. Nonetheless, this is understandable, as there are millions of species of insects, all in different environments, making it difficult to establish a sufficient dataset for several species of insects, much less a general dataset of all insects. Regardless, for many applications, such as insect monitoring, the number of species necessary can be narrowed down based on what species are expected within a particular region. Therefore, one approach to circumvent the issue is to enable potential researchers and users to obtain the desired insects needed for the application accurately and conveniently. To implement this approach, this body of

work developed a module that allows users to obtain data from highly credible and abundant sources of insect images and automated many of the processes to reformat the data for training conveniently.

### 3.1.1 Datasets

| | Taxon | | | | |
|---|---|---|---|---|---|
| | **Kingdom** | **Phylum** | **Class** | **Order** | **Family** |
| **Classification** | Animalia | Arthropoda | Insecta | Coleoptera | Coccinellidae |
| | | | | | Lampyridae |
| | | | | Diptera | Chironomidae |
| | | | | | Syrphidae |
| | | | | Hemiptera | Cicadellidae |
| | | | | Hymenoptera | Ichneumonidae |
| | | | | | Halictidae |
| | | | | | Vespidae |
| **Count** | 1 | 1 | 1 | 4 | 8 |

Table 3.1: Taxonomy of Labels in Dataset

As shown in the table, the dataset used for model training contains 8 families of insects from 4 different orders, all of which are within the same classification for each of the higher taxon. Hence, in total, there will be 15 labels for classification, 8 of which were families of insect that were chosen based on the ground-truth data later discussed in

the section on evaluation. Note that the family taxon is the lowest level of taxonomy used in this dataset. This ensured that there were enough data for each insect, since going to any taxon lower in the hierarchy meant more specific insects and less data available per insect. The images are collected from two sources, both of which are indexed by an international governmental platform that records all data about all types of life for open access, known as the Global Biodiversity Information Facility (GBIF) [7]. The main data source that makes up the majority of images is from iNaturalist Research-grade Observations [8], which contains images that are cross-checked thoroughly and contributed by an online community of experts and enthusiasts across the world. The second source is from the International Barcode of Life (iBOL) project, which has a similar and credible infrastructure as iNaturalist. Nonetheless, this source was used only for the particular family of insects, Chironomidae, since the data from iNaturalist was insufficient. Ultimately, based on general knowledge of machine learning and prior experience, the chosen amount for training was 9,000 images for each family of insect, totaling to a dataset of 72,000 images.

### 3.1.2 Data Collection

Collecting the data revolved around using GBIF's infrastructure. With the images from either source being indexed, a user can select the criteria to download the desired insects with particular constraints, including the source, taxon, and even geographical boundaries. For example, a user can search for the occurrences of the Lampyridae family, and constrain the results to only images from iNaturalist and within square coordinates in

the United States. Furthermore, GBIF provides a multimedia text file, containing download links to each of the images that resulted in the search. To obtain those images conveniently, data collection scripts were developed to semi-automate this process. The script enables a user with the multimedia text files of their desired insects to automatically download all of the images for each insect, and organize them based on their lowest taxon classification. Additionally, after downloading and organizing the images, the corresponding file containing the respective labels for each image and the training directory containing all of the images are created. For example, after getting the multimedia text file for each of the 8 families in the dataset, the scripts were used to download 9,000 images for each family into their respective folders, after which the file containing the 72,000 labels for all images and the training directory containing all 72,000 images were created.

### 3.1.3 Data Preparation

After collecting and organizing the data into a training directory and a corresponding label file, the two components still needed to be assembled into a dataset format. To implement this, a script was developed to first create mappings of each unique label found (e.g., Diptera, Lampyridae) by numerically encoding each label as a unique integer index (e.g., 1, 2). An example is shown in Table 3.2. Afterwards, with the integer mapping of each unique label, the target vector for each image file can be created by one-hot encoding the integer mapping of all labels associated with each image. More specifically, each image will have a target vector of length equal to the total number of

labels, where an index represents a particular label and a value of 0 or 1 denotes if the image is classified with that label. Finally, the image files can be converted to pixel arrays and be assigned to their respective target vectors. After shuffling the pairs, all are compressed and synthesized into a dataset file that can be loaded for model training.

| **Label** | **Index** |
|---|---|
| Animalia | 0 |
| Arhropoda | 1 |
| Chironomidae | 2 |
| ⋮ | |
| Lampyridae | 13 |
| Syrphidae | 14 |
| Vespidae | 15 |

Table 3.2: Example label-to-integer mapping of labels within dataset

## 3.2 Optimization

After the data has been collected and prepared for use in model training, the next step is to optimize the model with the dataset. The optimization stage of the framework consists of two main steps- model selection and model training.

### 3.2.1 Model Selection

Model selection involves selecting and importing the model, then modifying it for training. For selection, the framework allows the user to import a model available within Keras from the TensorFlow library [9] for transfer learning. Next, to modify a CNN model to perform transfer learning, the main convolutional and pooling blocks have to be set as trainable to enable fine-tuning. Afterwards, additional layers have to be appended to the end of the model's network to modify the model for classification with new labels. This can include a layer for flattening, a Relu layer, and the final output layer, which has the same number of nodes as there are labels from the dataset. For example, one of the models used for this work was the VGG16 model. The VGG16 model had its final block's convolutional layers and pooling layer to be trainable, along with adding one flattening layer, one Relu layer, and a final output layer with 15 nodes for 15 labels. Additionally, during this stage, the optimizer and performance metric for training can be set as well, in which case the VGG16 model used stochastic gradient descent and the F-beta score, respectively.

### 3.2.2 Model Training

After selecting and modifying the model for training, the next step is to train it. This step first loads the dataset with the dataset file created in the previous stage, and splits it into a training and test set. Additionally, the images in the dataset are rescaled and augmented to maximize the training performance. After selecting the number of epochs, the model will finally be prepared for training. During model training, the model is utilizing its chosen optimizer, such as stochastic gradient descent, and the F-beta metric to optimize its parameters. Since a chosen optimizer, such as stochastic gradient descent or others, is popularly used for training and already explained within many of the previous works in this domain, the new addition to be discussed in the next section is the F-beta score used as the performance evaluation metric for the specific task of multi-label image classification. This framework also includes a Tensorboard and logs to keep track of the computed score after each epoch and also saves the checkpoint of each epoch. Lastly, after the model has finished training, the model is saved into a file that can be loaded for further testing and evaluation.

## 3.3 Evaluation

### 3.3.1 F-beta

The metric used to evaluate the state of performance by the model, during training and when training is completed, is the F-beta score. Mathematically, the F-beta score is very similar to the F1 score commonly used by many other works and machine learning tasks, except that it has a beta parameter that is used to add weight to the ratio between precision and recall. More specifically, the F-beta score is defined as:

$$F_{beta} = (1 + beta^2) * \frac{(precision * recall)}{(beta^2 * precision) + recall}$$

where

$$precision = \frac{TP}{TP + FP} \text{ and } recall = \frac{TP}{TP + FN}$$

On a high level, precision can be thought of as a model's ability to correctly predict a positive label among all positive predictions, while recall can be thought of as a model's ability to correctly predict the positive label among all actual positives. This is illustrated in Figure 3.1. In other words, maximizing precision minimizes false positives and maximizing recall minimizes false negatives. With regards to multi-label classification, precision is the percentage of labels assigned to the image that are actually true out of all the labels assigned to it while recall is the percentage of actual true labels assigned to the

image out of all the actual true labels. For example, having a beta value of 2 will put more attention on recall and minimizing false negatives, meaning the task at hand prefers to have more correct labels in total for the image than it does to have the labels of an image to not be wrong. In general, a model with a beta value of 2 cares more about detecting and discovering the correct labels for an image rather than being precise among the correctness of the labels. Overall, not only will this metric be used for optimization during model training, but as a performance score for the model's ability to do multi-label image classification on the insect image dataset sourced from iNaturalist and iBOL. Although there is no other external benchmark currently specifically for multi-label classification of insect taxonomy, the final F-beta score of the model is still a good indication of its classifying ability since the dataset has a sufficient variety of images.

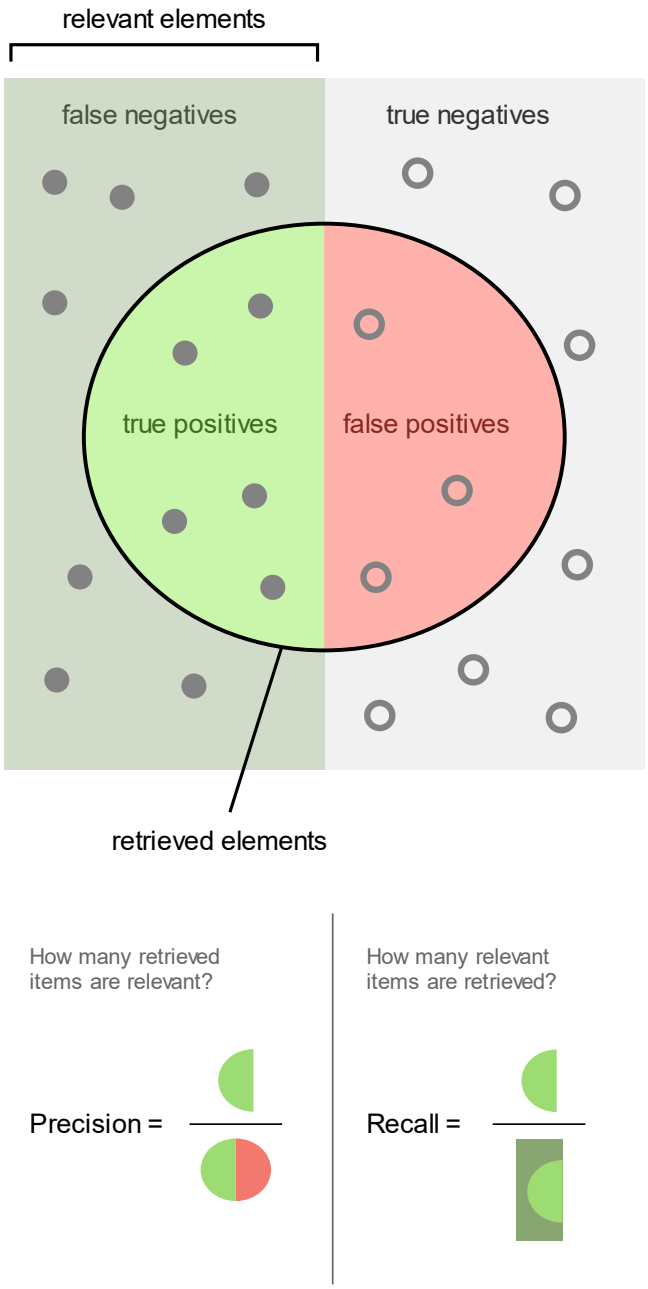


Figure 3.1: Illustration of precision and recall [10]

### 3.3.2 Manually Labeled Classifications from Expertise

After using the F-beta metric to evaluate the performance of a model for training and optimization, this work also used the trained model to evaluate it for the specific application of insect monitoring. To implement this, this framework not only includes code to use a trained model to predict the taxon classifications of a single insect image, but also a script that scales this task to detect insects and predict their classification in all frames from video footage that were captured during an insect monitoring experiment with camera-based insect traps. An example image frame is show in Figure 3.2 below.


Figure 3.2: Example image frame from video footage captured during insect monitoring

A portion of the footage were manually labeled by expert entomology researchers in the experiment. More specifically, as shown Table 3.3, videos of a few hours for three days of monitoring were thoroughly examined and timestamps containing insects were labeled. Those timestamps are then the ground-truth labels matched with the corresponding image frames used in the script to enable comparison between the manually labeled image frames with the labels predicted by a model. With this ground-truth data, the model can be evaluated more specifically for an example application of insect monitoring.

| Day | Insect Number | Time entered | Time left | Insect Order | Insect Family | Shadow or Insect? |
|---|---|---|---|---|---|---|
| 9/19/2021 | 2021_073 | 10:31:45 | 10:31:47 | Hymenoptera | Vespidae | Insect |
| 9/19/2021 | 2021_073 | 10:32:14 | 10:32:18 | Hymenoptera | Vespidae | Insect |
| 9/19/2021 | 2021_073 | 10:32:26 | 10:32:32 | Hymenoptera | Vespidae | Insect |
| 9/19/2021 | 2021_073 | 10:34:49 | 10:35:02 | Hymenoptera | Vespidae | Insect |
| 9/19/2021 | 2021_073 | 10:35:12 | 10:35:13 | Hymenoptera | Vespidae | Insect |

Table 3.3: Example excerpt of manually labeled timestamps from expert

# Chapter 4

# Results

## 4.1 F-beta Scores

### 4.1.1 VGG16

As a baseline for F-beta scores, VGG16 was the model selected, fine-tuned with transfer learning, and trained for multi-label classification. It has a CNN architecture and had state-of-the art performance in image classification for some time while also being conveniently integrated into Keras for easy import and transfer learning, making it a good choice for a starting baseline. In total, the model has approximately 15.7 million parameters. However, with transfer learning, only the final block's convolutional and pooling layers were made trainable for fine-tuning. Moreover, a flattening layer, Relu layer, and a final dense output layer of 15 nodes were appended and were trainable as well. Hence, training only involved approximately 8.1 million parameters. After training for 50 epochs with a beta value of 2, the model was able to obtain a final F-beta score of 95.95% on the dataset, with a final loss of 5.928%.

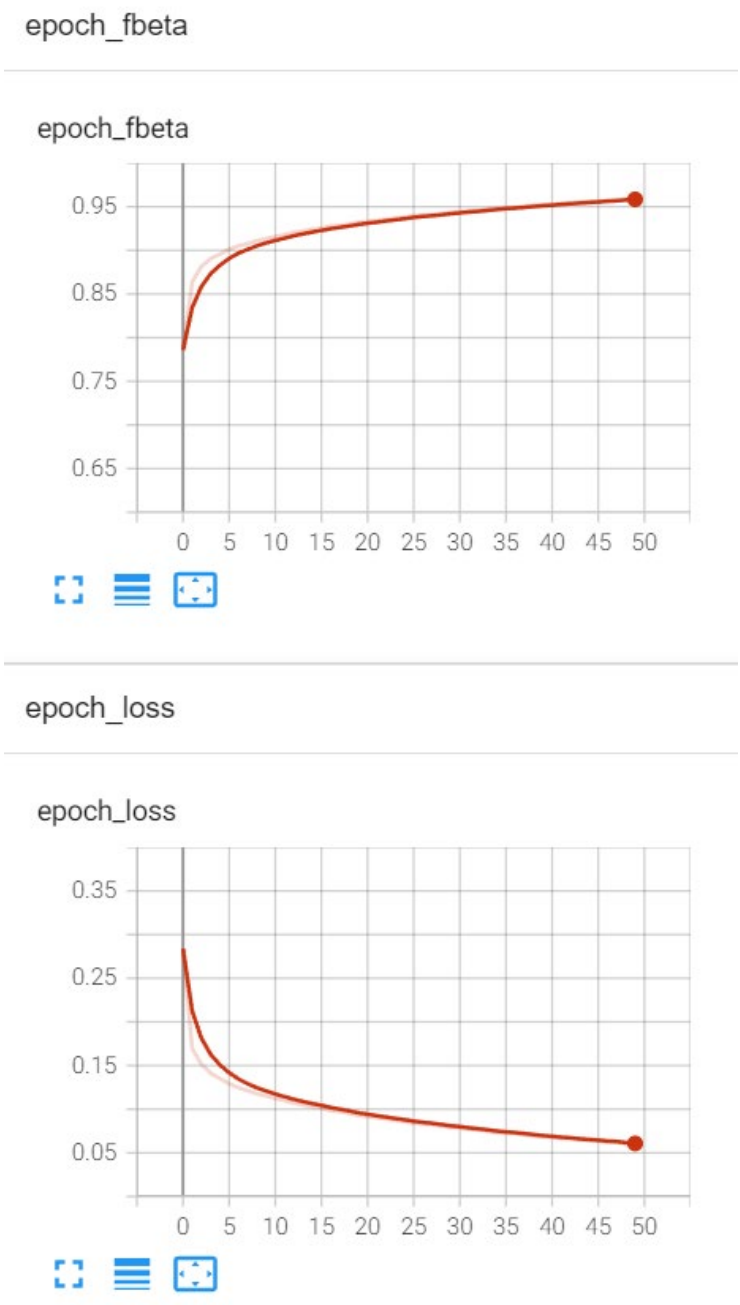


Figure 4.1: F-beta score of VGG16 after 50 epochs w/ beta as 2

The model was then trained again with a beta value of 0.5. This was to see if weighing the precision of the model more than recall would have any affect. As a result, the final f-beta score for this implementation was 96.4% with a loss of 6.636%.

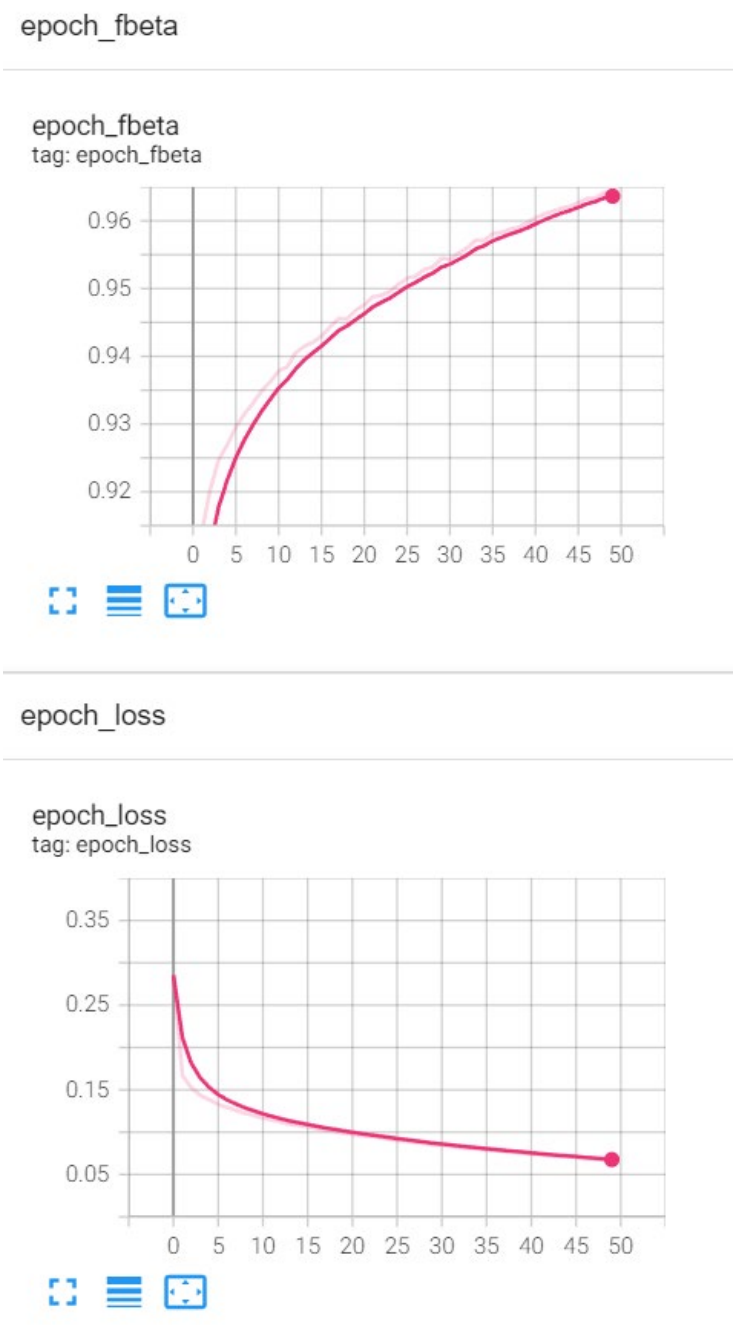


Figure 4.2: F-beta score of VGG16 after 50 epochs w/ beta as 0.5

## 4.2 Accuracy on Insect Monitoring Footage

### 4.2.1 Baseline Model (MobileNetV2)

The baseline model, which was a MobileNetV2 CNN architecture that was trained with only 4 labels for single-label classifications, was able to get 59 classifications out of 123 classifications in the order taxon correct. This meant it obtained an accuracy score of 47.97%. Note that its labels were the common colloquial names for the order, such as Hymenoptera would be mapped to Bee and Coleoptera would be mapped to Beetle.

### 4.2.2 VGG16

Compared with the manually labeled ground-truth classifications, the model was able to make 30 classifications out of 123 classifications in the order taxon that was the same as the ground-truth. This meant it obtained an accuracy score of 24.4% for the order taxon. In the family taxon, the model was able to only make 15 classifications out of 65 classifications in the family taxon for a score of 23.1%. Combining classifications for both levels of taxon, the model was able to have 23.9% of its classification to be the same as the ground-truth's classification.

# Chapter 5

# Discussion

## 5.1 Analysis on F-beta Scores

### 5.1.1 Effect of Beta Term on F-beta Results

As shown in the previous section, the model was able to achieve a score of 95.95% with a beta value of 2 while a model with a beta value of 0.5 was able to receive a score 96.4%. This implies that preferring the model to weigh precision more does increase the overall accuracy, but not by a significant amount. Intuitively, this makes sense since the accuracy or correctness of the model depends more on if it can identify the taxon labels correctly, rather than having more taxon labels identified. For example, if the model receives an image containing a wasp with the two lowermost taxon labels as Hymenoptera and Vespidae for the order and family taxon, respectively, then it would be better for the model to correctly identify each taxon that it's able to identify. In this case, if the model is able to have a high probable classification, such as Hymenoptera in the order taxon, then the model would prefer that to actually be correct rather than if it was able to obtain labels for more levels of taxon, such as having Hymenoptera and Halictidae, but both or at least one end up being wrong.

### 5.1.2 Accuracy and Performance

The final F-beta scores from both implementations of the VGG16 model had really high accuracy scores. Since the optimization code split up the dataset to have a training set and a testing set, it is still a valid indicator of performance for multi-label image classification on insect taxonomy. However, since there is no official external benchmark for this task that has been peer reviewed within the machine learning community, the only valid conclusion to be drawn is that the models are able to be accurate with the types of images that are within the dataset, which are relatively higher quality images of the insects compared to the insect monitoring experiment.

## 5.2 Analysis on Results from Insect Monitoring Footage

### 5.2.1 Comparison Between Scores of Different Levels of Taxon

As seen in the results for the VGG16 model's performance on the insect monitoring footage, there were more correct classifications in the order taxon than the family taxon. However, the difference is not that much so it is uncertain for any implications that the model has learned any hierarchical relation between the labels based on this result.

### 5.2.2 Comparison Between Scores of Different Models

The baseline model with the MobileNetV2 CNN architecture performed much better than the multi-label model with the VGG16 architecture. There are several reasons that can be inferred for this result. The first is that the MobileNetV2 model was trained on a dataset of over 400,000 images, compared to the 72,000 images for the VGG16 model. This means that it had a lot more images to train on and thereby, would surely do better on correctly classifying the images, especially on images like the frames from the insect monitoring footage that varied greatly from iNaturalist quality images. Another suspected reason for this is that it was trained on less labels and the labels were in more general and higher levels of taxon. More specifically, the approximate 400,000 images it had to train on only consisted of flies, beetles, bees, and butterflies, which meant 100,000 images for each label on average. This is way more than the 9,000 images on average that each of the 15 labels for the VGG16 model would have. Additionally, since it only had 4 labels, each of the labels were within the order level of taxon, which is higher and more general so the dataset had more variability and coverage to improve performance. To make it concrete, a model can learn to classify with the shape and physical features of a bee in general much better than it can learn to classify a more specific family of bees that require more explicit and granular details to distinguish it. Lastly, the MobileNetV2 CNN architecture is a more recent and advanced CNN architecture than VGG16. This could have also impacted the performance difference since the MobileNetV2 model has higher benchmark scores for general image classification.

## 5.3 Classifying Unseen Families to Higher Orders

To further evaluate the model, the model was tested on images of insects that are within families that it has not been trained on, but are within order that is has been trained on. The goal was to see if the model is able to identify the unseen family to the correct higher order based on the training of another family in that order. More specifically, a family from each order in the trained model was tested. The selected families for evaluation and their results are shown in the table below. Evidently, only the family from the Diptera order was incorrectly classified.

| Order | Family | Order Prediction |
|---|---|---|
| Coleoptera | Chrysomelidae | Coleoptera |
| Diptera | Calliphoridae | Hemiptera |
| Hemiptera | Membracidae | Hemiptera |
| Hymenoptera | Formicidae | Hymenoptera |

Table 5.1: Unseen insect families and their orders

To illustrate the experiment more clearly, shown below in Figure 5.1 are the pictures of the new unseen insect families versus a sample of pictures of the families of insects in the same order that the model trained on. It is clearer now to see a possible reason the model might be correct about certain orders versus others. In particular, the images from the insect orders of Coleoptera, Hemiptera, and Hymenoptera, in which the unseen insect families were correctly labeled, the insects are more similar in appearance. This is

opposed to the insect families in the Diptera order, in which the unseen insect family of Calliphoridae looked really different from the seen insect families within the same order. Overall, a more extensive experiment can be done to have more robust proof and evidence, but this does indicate that the model might be able to learn the hierarchy of the labels since it learned the physical features of the insect families it trained on and can extrapolate from that to generalize to unseen insect families.

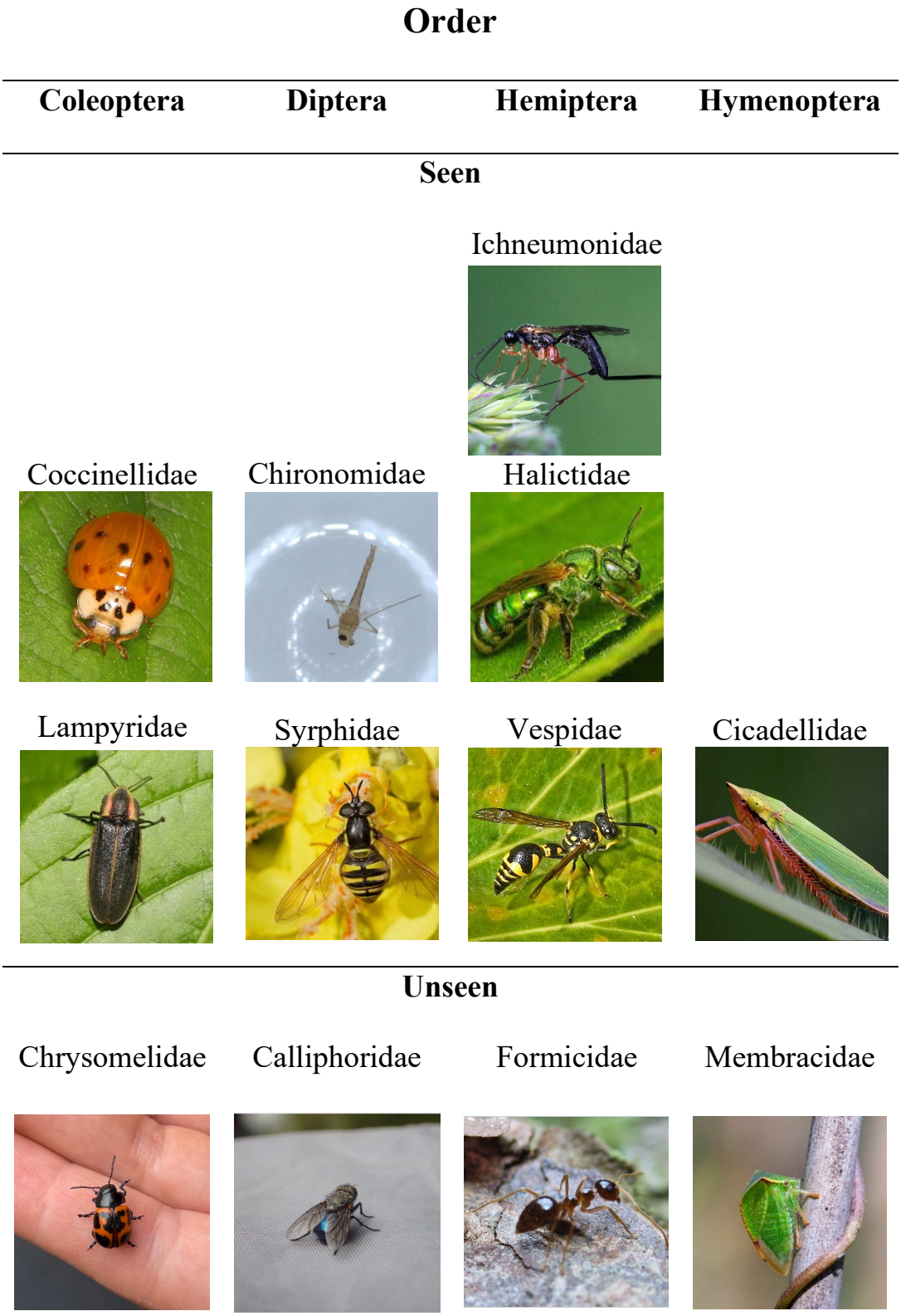


Figure 5.1: Images of Seen vs Unseen Insect Families

# Chapter 6

# Conclusion

## 6.1 Summary

Multi-label image classification on insect taxonomy is a useful task to explore and improve upon. Not only does research for this topic help out with the technical progress of multi-label image classification and even general image classification, but it has contributions to applications such as insect monitoring and beyond. This work explored and contributed to the specific application of insect monitoring and insect classification in general because it can have potentially significant real-world impacts. Furthermore, the processes involved within this type of work are not as technologically up-to-date as they can and should be to optimally leverage machine learning tasks such as multi-label image classification. Applications aside, the inherent hierarchical structure of insect taxonomy made it a unique and interesting inquiry that can provide great insight onto how machine learning models can learn or not learn about the relations of the multiple labels that objects can have. Overall, this work sought to showcase each of those reasons in many ways. First is by developing a framework that greatly automates and facilitates many of the processes that are required for multi-label image classification on insect taxonomy while also providing example model results and their analyses from using the framework. As seen in the results, it is possible to train models conveniently with desired insect labels and get high scores, as the models achieved higher than 95% F-beta scores. Additionally,

even though the results for insect monitoring can be greatly improved upon, these models can be applied to insect monitoring, especially with larger datasets and better architectures as indicated by the results. Lastly, although there are no strong conclusive evidence that a multi-label model can learn the hierarchy of the taxonomic labels, the results still provided insight that future work can use in experiments for further research.

## 6.2 Future Work

As previously mentioned, this work established a framework that can be used and improved upon for multi-label image classification on insect taxonomy. Some of the more straightforward potential ideas that can help contribute is to implement this framework to train better CNN models on larger datasets. This will explore the potential that models have for many applications involving multi-label image classification such as insect monitoring. Regarding the insect monitoring experiment done to evaluate the models, better results can be obtained with better images and video footage, as well as including an object detection step that can crop out the insect for the model to focus on without the background noise. Taking it a step further, not only can different and better CNN models be explored, but newer and potentially greater model architectures can be explored as well. For example, [6] performed multi-label image classification on satellite images, but with a vision transformer network. This idea can be extrapolated and applied to insect taxonomy to potentially improve insect classification performance. Furthermore, not only can performance and the models affecting it be explored, but the question of whether models can showcase or learn the hierarchical structure of insect taxonomy be

inquired. An example of this can be found in [11]. This work was able to build a visual attention network based off the transformer architecture that could classify images, and was able to generate a visualization of the model's attention in each image, as shown in Figure 6.1. This technique can be used and built upon to help generate insight into a attention model's understanding of the hierarchy of the multiple labels by visualizing its attention for the classification of each taxonomic level.

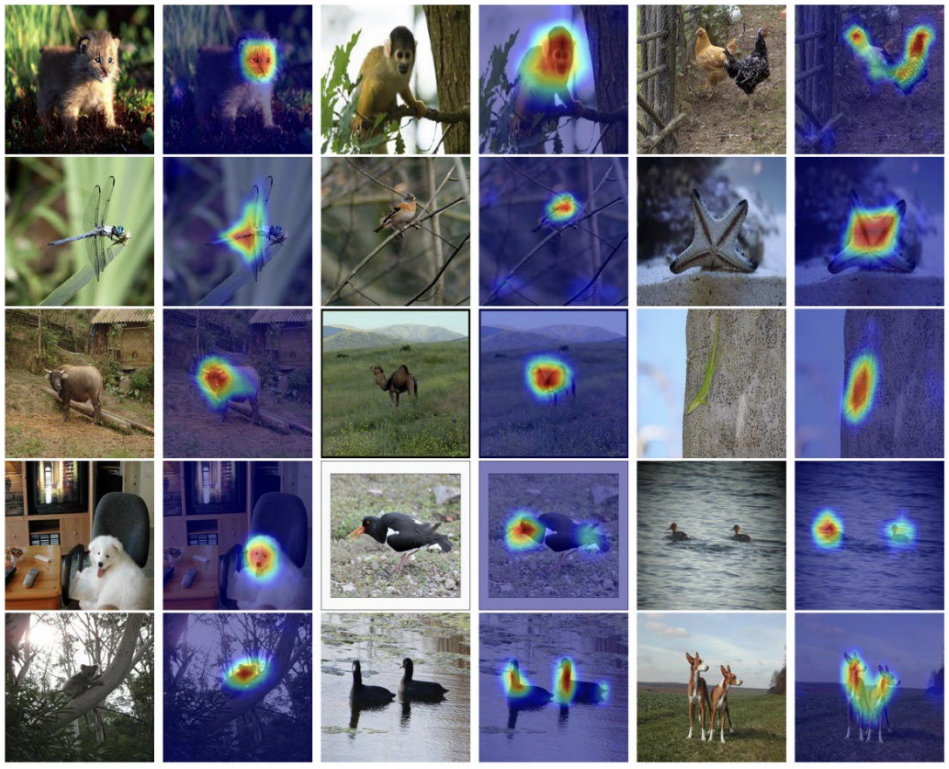

Figure 6.1: Visualization results. All images come from different categories in ImageNet [11]

Ultimately, there are an many ideas that can stem from this work and many potential research questions that can be very impactful when explored. Multi-label image classification, and in particular, its application to insect taxonomy and similar domains, still have a lot of room to be improved upon and be impactful.

# References


[1] O. L. P. Hansen, J.-C. Svenning, K. Olsen, S. Dupont, B. H. Garner, A. Iosifidis, B. W. Price and T. T. Høye, "Species-level image classification with convolutional neural network enables insect identification from habitus images," *Ecology and Evolution,* vol. 10, no. 2, pp. 737-747, 2020.

[2] T. Kasinathan, D. Singaraju and S. Reddy Uyyala, "Insect classification and detection in field crops," *Information Processing in Agriculture,* vol. 8, no. 3, pp. 446-457, 2021.

[3] M. Valan, K. Makonyi, A. Maki, D. Vondráček and F. Ronquist, "Automated Taxonomic Identification of Insects with Expert-Level Accuracy Using Effective Feature Transfer from Convolutional Networks," *Systematic Biology,* vol. 68, no. 6, pp. 876-895, 2019.

[4] D. Gardner and D. Nichols, "Multi-label Classification of Satellite Images with Deep Learning," Stanford, 2017.

[5] N. Khan, U. Chaudhuri, B. Banerjee and S. Chaudhuri, "Graph convolutional network for multi-label VHR remote sensing scene recognition," *Neurocomputing,,* vol. 357, pp. 36-46, 2019.

[6] F. Sánchez-Bayo and K. A. G. Wyckhuys, "Further evidence for a global decline of the entomofauna," *Austral Entomology,* vol. 60, no. 1, pp. 9-26, 2020.

[7] Global Biodiversity Information Facility, "GBIF: The Global Biodiversity Information Facility," 13 January 2020. [Online]. Available: https://www.gbif.org/what-is-gbif. [Accessed 2022].

[8] iNaturalist, "iNaturalist Research-grade Observations," iNaturalist, 2022.

[9] F. Chollet and others, "Keras," GitHub, 2015. [Online]. Available: https://github.com/fchollet/keras.

[10] Wikipedia contributors, "Precision and recall," Wikipedia, 2022. [Online]. Available: https://en.wikipedia.org/wiki/Precision_and_recall. [Accessed 12 3 2022].

[11] M.-H. Guo, C.-Z. Lu, Z.-N. Liu, M.-M. Cheng and S.-M. Hu, "Visual Attention Network," arXiv, 2022.

[12] M. Maj, "Object detection and image classification with Yolo," 2018. [Online]. Available: https://www.kdnuggets.com/2018/09/object-detection-image-classification-yolo.html. [Accessed 8 March 2022].

[13] A. Dolezal and P. Baluch, "True Bugs," Arizona State University, 6 October 2010. [Online]. Available: https://askabiologist.asu.edu/explore/true-bugs. [Accessed 8 March 2022].

[14] M. Kaselimi, A. Voulodimos, I. Daskalopoulos, N. Doulamis and A. Doulamis, "Forestvit: A vision transformer network for convolution-free multi-label image classification in deforestation analysis," in *International Conference on Machine Learning*, 2021.