Taxonomic Classification of Objects with Convolutional Neural Networks

SungRyeol Yang
Dept. of Computer Engineering
Korea Polytechnic University
Siheungsi, Korea
sryang@kpu.ac.kr

Geoffrey C Fox
School of Informatics, Computing, and Engineering
Indiana University
Bloomington, USA
gcf@indiana.edu

Bokyoon Na
Dept. of Computer Engineering
Korea Polytechnic University
Siheungsi, Korea
bkna@kpu.ac.kr

***Abstract*— it is difficult to build a CNN model that can classify many classes at once. Therefore, this study does not want to make many classes recognizable at once using only one model but by taxonomic classification. This study suggests a method of dividing the large number of classes into different steps of each step using Taxonomic classification. We propose a method of classifying a large number of classes by dividing them into models for each step using taxonomic classification. Our method uses taxonomic classification to distribute the weights required for training and test step by step. This will save a lot of time than creating a one-level model. In addition, to detect objects in never trained categories, the result may come up to a certain step without retraining the model. This shows that part of the model can be recycled. In this study, we presented a way to distinguish large numbers of classes using taxonomic classification by using multiple datasets, such as PASCAL VOC2012, ILSVRC 2013 image data from ImageNet, and 102 Category Flower Dataset.**

Keywords—Object classification, Taxonomic classification, Convolutional neural networks, Machine learning

I. INTRODUCTION

The convolutional neural networks (CNN) [1] is a partial connection structure, which is a technique to reduce the complexity of the model. By applying convolutional arithmetic, model complexity can be lowered and good features can be extracted. Because CNN learns features directly, there is no need to manually extract features. Due to its small learning parameters, CNN also has a higher recognition rate than other algorithms, with an accuracy of approximately 70 to 90 percent.

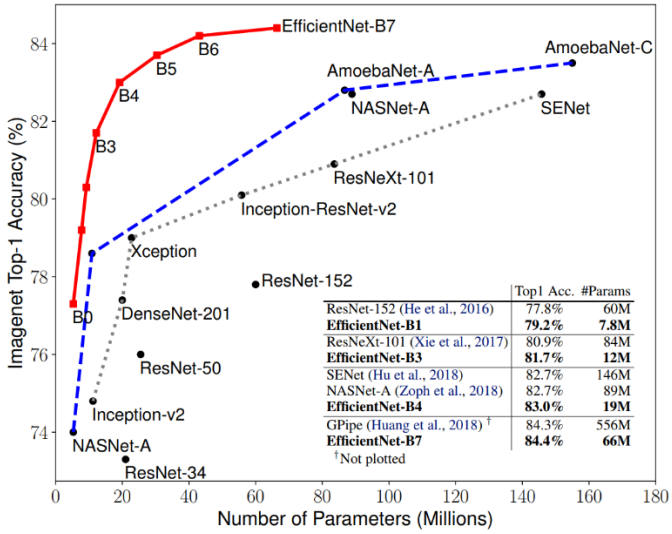

Figure 1: Performance table performed by Google with imagenet [2]

However, as the number of categories increases, the number of layers such as convolutional with ReLU and pooling increases. As the number of layers increases, the amount of computations increase, which take a long time to extract features. To demonstrate this, let's take a paper [3] that took time to process 10,000 categories and 9 million images when using a 2.66GHz Intel Xeon CPU. When using LIBLINEAR belonging to linear SVM [4][5][6], 1 v.s. all (training 10,000 such classifiers) method took 1 year of training time and 16 hours of testing. In the SPM+SMV method, the intersection kernel SVM is 100 times slower than the linear SVM, so it takes more than 100 years. This means that it is difficult to distinguish many types of categories at once. To prove this, in [7] where the time complexity of CNN was obtained, the time complexity of the entire convolutional layer is represented by $O\left(\sum_{i=1}^{d} n_{i-1} \cdot s_i^2 \cdot n_i \cdot m_i^2\right)$. According to this formula, i is the index of a convolution layer, d is the depth (number of convolution layers), n_i is the number of filters (also known as "width") of the i-th layer, n_{i-1} is the number of input channel, s_i is the spatial size (length) of the filter, and m_i is the spatial size of the output feature map. Considering the size of the output feature map without considering the rest of the factors, it can be seen that it takes a lot of time to take $O\left(\sum_{i=1}^{d} m_i^2\right)$.

As the number of classes (or categories) increases in CNN, the size of weights to be trained and tested generally increases, so the time to train and test increases. For example, consider a CNN that can distinguish 1000 classes [8]. Looking at Table 1, Conv is a convolution layer, Pool is a pooling layer, and FC is an Fully-Connected layer. Suppose that it is possible to distinguish 1,000,000 classes, not 1000 classes. Even with this structure, 4,096,000,000 weights are generated in the FC:8 layer only. Even with the simple structure of 8 layers except the pooling layer, the weight increases as the number of classes increases. As a result, it can be seen that learning and execution time is long. Therefore, to make a single model that can distinguish a large number of classes, it takes a lot of time per epoch to train and the total training time is excessively long, so it is practically impossible. For this reason, the object detection paper using CNN generally sets the classes to about 20 classes or reduces the number of layers if there are many class types.

Here are examples of setting the number of classes to around 20. An example [9] uses YOLO v3 to identify objects in satellite images. This is divided into a parent level detector and a child level detector. The parent level detector separates 10 such as Truck, Passenger-Vehicle and Building. Child

level detector sifts between 60 classes derived from parent level detector. For example, child level detector can classify it into subclasses, such as Pickup truck and Cargo truck derived from the Truck class. In addition, [11] used 3D-SSD can detect 3D objects in indoor images, including beds, chairs and boxes. Up to 19 classes can be divided, such as toilet sins and trash-cans. In [14], a mix of YOLO and ResNet used to identify multiple objects in complex natural scenes, but 20 classes distinguished.

TABLE I. EXAMPLE OF CNN'S STRUCTURE AND LEARNING TARGET WEIGHTS THAT CAN DISTINGUISH 1000 CLASSES [8]

layer	Filter/ block size	Num. of Filters	stride	padding	Node (output size)	Learning target weights
Input					224×224×3	
Conv:1	11×11×3	96	4	3	55×55×96	(11×11×3+1)×96
Pool:1	3×3		2		27×27×256	
Conv:2	5×5×96	256	1	2	27×27×256	(5×5×96+1)×256
Pool:2	3×3		2		13×13×256	
Conv:3	3×3×256	384	1	1	13×13×384	(3×3×256+1)×384
Conv:4	3×3×384	384	1	1	13×13×384	(3×3×384+1)×384
Conv:5	3×3×384	256	1	1	13×13×256	(3×3×384+1)×256
Pool:5	3×3	256	2		6×6×256	
FC:6					4096	6×6×256×4096
FC:7					4096	4096×4096
FC:8					1000	4096×1000

Here are some examples of how to reduce the number of layers to distinguish many classes. Let's take an example to compare based on YOLOv2 [15]. In [16] that classifies 80 classes, total of 31 layers were used, including 5 max-pooling layers and 23 convolution layers. 71.4 fps comes out when using GPU and CUDNN [17]. YOLO9000 [15] using YOLOv2 suggested a way to distinguish between 9000 categories. The YOLOv2 separated the 20 class of the VOC2007 dataset, with 67fps at 76.8mp and 40fps at 78.6mp. However, the Darknet-19 used to speed up than YOLOv2 used only 19 convolution layers and 5 max-pooling layers.

Besides the reason that increasing the number of detectable categories takes a long time, there is another problem. At making changes to a model that has already been trained to distinguish between objects in a new class and trained classes, there is a burden of relearning the model from the beginning. In general, the larger the category number, the larger the size of the entire dataset to be learned, which takes a long time. In real, new objects are constantly created and destroyed, so it is difficult to make quick changes to the model as needed using the existing method.

II. RELATED RESEARCH

CNN usually applies activation function to the convolution arithmetic result, and then applies the pooling operation to the result. At this time, if the feature map becomes smaller, it also helps speed up and memory efficiently. Therefore, stride more than 2 get the effect of reducing the size of the feature map.

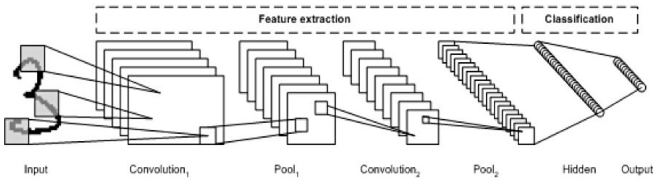

Figure 2: CNN architecture[18]

However, to separate a large number of objects at once, the number of layers increases usually. This will take a long time to analyze even if there are reducing the size of the feature map by doing more than two strides. It made us to consider of a new way to use CNN's structure, but to reduce slow-down so that we can distinguish between different kinds of objects.

From a method of biological science classification when classifying organisms, we propose a taxonomic classification of objects with convolutional neural networks, Biological science classification is necessary to distinguish the myriad creatures on the earth. It is a taxonomy that binds creatures of the same visible characteristics according to standards, and divides creatures of different characteristics. As the organism is divided according to the biological forms of visible appearances, it is possible to know the relationships between the various organisms, characteristics, evolutionary processes. And it is possible to find the desired organism efficiently. Using this, if the species found in the existing can be easily looked up through the characteristics of the existing. If a new species is discovered, it is possible to deduce the relationship between existing organisms using existing taxonomy. The criteria for classifying inanimate objects are not yet a systematic taxonomy that is officially common in use as a biological taxonomy in the society. Thus, there is a need to define a new taxonomy.

Biological science classification was developed based on a systematized and defined taxonomy by Carl von Linne. Currently, it is known that the main use of the biosorting corporation 3-domains and 6-kindoms taxonomy commonly used in popular journals. Of course, some people use their own taxonomy as needed, or modifies and complements the existing method. Therefore, biological science taxonomy [19] is developing.

III. PROPOSED METHODS

The 'Taxonomic classification of objects' approach we propose is as follows; During detecting with each step-by-step model, each step model crops the bbox where the object was detected and hands it over to the next step. Each step was organized as the next paragraph. Note that, to accurately classify objects in biological sciences, it must be analyzed with RNA, but it is enough to classify objects from the image because there are partially different visual features for each organism. This is called Morphological taxonomic. We focus only on classifying objects using imagery features. For example, dog is hairy, crocodiles have wet skin even though they go on all fours.

1. Root – as step 1, distinguishes whether it is a creature or an inanimate object. If the inanimate object is selected, then a sub-classification method according to a classification such as electronic devices / non-electronic devices, et

c. follows for suitable inanimate objects. If the result is a creature, then as the next step another CNN will follow a ccording to the biological science classification that is ap propriate for the organism. Since inanimate objects are a mbiguous in taxonomic because there are no systematic c riteria to distinguish into sub-steps, we have built a way t o explain them.

2. Domain - Step 2 operates based on the results classified i n the above step, Root. Step 2 separates the domain of cr eatures, such as Domain Bacteria/Eukaryota. For exampl e, if it is determined to be Domain Bacteria in step 2, the next step proceeds to the sub-classification method for D omain Bacteria. If it is determined to be Eukaryota in ste p 2, the next step will proceed with the sub-classification method for Eukaryota. It will save a lot of computations on nodes of hidden layers in CNN.
3. Kingdom - Step 3 operates based on the results classified in the above steps. Step 3 distinguishes that the Plantae/ Animalia corresponds to a Kindom. For example, if the r esult of step 3 is determined to be a Plantae of Eukaryote s, the next step proceeds to the sub-classification method for the Plantae. If the result of step 3 is determined to be Animalia, the next step proceeds to the sub-classificatio n method for the Animalia.
4. Phylum - Step 4 operates based on the results classified i n the above steps. Step 4 distinguishes the phylum. For e xample, if the result of step 3 is determined by Animalia, it is divided into the Chordata, Nematoda in the step 4. I f the result of step 3 is determined by Plantae, it is divide d into Gnetophyta, Cycadophyta, etc. When the results of this step are released, the next step is to proceed with the sub-classification according to the results separated at th is step.
5. Class - Step 5 operates based on the results classified in t he above steps. Step 5 distinguishes Class. For example, if the result of step 4 is Chordata, the result of step 5 can be Mammalia, Aves, etc. When the results of this step ar e released, the next step is to proceed with the sub-classif ication according to the results separated at this step.
6. Order - Step 6 operates based on the results classified in t he above steps. Step 6 distinguishes Order. For example, the Order corresponding to Mammalia is Primates, Lago morpha. The Order corresponding to Aves includes Colu mbiformes, etc. When the results of this step are released, the next step is to proceed with the sub-classification acc ording to the results separated at this step.
7. Family - Step 7 operates based on the results classified in the above steps. Step 7 distinguishes Family. For examp le, The Family has Hominidae, Columbidae, etc. When t he results of this step are released, the next step is to proc eed with the sub-classification according to the results se parated at this step.
8. Genus - Step 8 operates based on the results classified in the above steps. Step 8 distinguishes Genus. For exampl e, there is a Homo, Pan, etc. When the results of this step are released, the next step is to proceed with the sub-clas sification according to the results separated at this step.

Species - Step 9 operates based on the results classified in the above step. Step 9 distinguishes Species. For example, there are Homo sapiens, Homo habils. When the results of this step are released, the next step is to proceed with the sub-classification according to the results separated at this step.

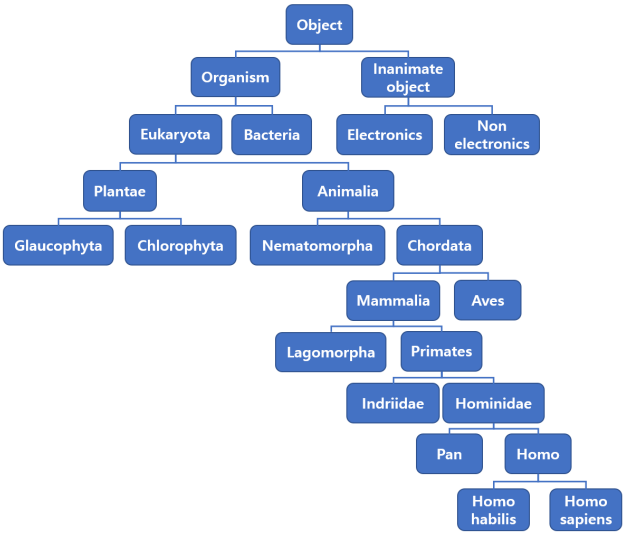

Figure 3: If we draw a step-by-step picture of the method described above, it is as follows: Many substeps are omitted and only flow is visible.

In a flow similar to

Figure *3*, we implemented Faster R-CNN in each step. However, we consider as future work that YOLO, DSSD will have more optimized results in parallel detections. Among the various algorithms, we choose to capture the features most effectively at each step of the way.

This method we proposed is to follow the taxonomy by checking the criteria of the taxonomy and the common characteristics of the object in a given image. In our proposal, we tried to divide object detections into several steps, reducing the time required to distinguish between many classes. Because our method follows steps, each step cuts only the area detected by each step and pass it to the next sub-step in the entire area of the image. It is possible to compare many objects in a short time because it is possible to reduce the amount of computation by reducing the number of areas and categories to be detected at once. In addition, when the detection area is reduced, the time required for the sliding window is reduced. For example, species of organisms can be found in the NCBI database [20][21][22] are about 466,327 species in now. If you try to classify these 466,327 species at once, it is not possible in the current way for the aforementioned reasons. However, using the taxonomic classification method can be divided into several phases to classify a large amount of category. Since there is no uniform Biological taxonomy, we followed examples based on NCBI databases [20][21][22], various papers [19][23] and K-12 education in Korea. Because we've mixed information, we are revealing that the taxonomy of the paper [19][23] we referenced and the taxonomy we used may be a little different. For example, Domain [23] can be divided into three categories: Bacteria/Archaea/Eukarya. By the three categories divided by Domain, they are divided into the next sub-step, Kingdom. Domain Bacteria's Kingdom is divided into Bacteria. Domain Archaea's Kingdom is divided into Archaea. Domain Eukarya's Kingdom is divided into Protozoa/Plantae/Fungi/Animalia/Chromista. Even though the entire category of Kingdom is 7 for this domain [19], it is broken down into a one Domain by the previous step.

Let's explain the next sub-step, Phylum (or Division). Animalia (Kingdom) in Eukaya (Domain) has 34 Phylum,

including Chordata / Hemichordata / Nematomorpha. Plantae (Kingdom) in Eukaya (Domain) has 8 Phylum, including Glaucophyta / Chlorophyta / Rhodophyta. Fungi (Kingdom) in Eukaya (Domain) has 5 Phylum, including Ascomycota / Basidiomycota / Zygomycota. Protozoa (Kingdom) in Eukaya (Domain) has 8 Phylum, including Amoebozoa / Choanozoa / Sulcozoa. Chromista (Kingdom) in Eukaya (Domain) has 10 Phylum, including Cryptista / Miozoa / bigyra. Bacteria (Kingdom) in The Domain has 29 Phylum, including Chlorobi / Dictyoglomi / Nitrospira. Bacteria (Kingdom) in Bacteria(Domain) has 29 Phylum, including Chlorobi / Dictyoglomi / Nitrospira. Archaea (Kindom) in Archaea (Domain) has two Phylum[19], including crenarchaeota / Euryarchaeota. Phylum (or Division) is a total of 96. However, it was also categorized in domain and kingdom, which are higher than Phylum. Therefore, there are actually up to 34 types that need to be classified at this step, less than 96. Therefore, the number of categories is reduced than the way all category is separated at once. The sub-step also increases the total number in the same way as above, but the actual classification category is reduced.

In addition to these advantages, our taxonomic classification allows object detection to identify or infer new object which is not trained until a certain level. When an object in a new class needs to be distinguished, the existing model may be reused rather than retraining the entire model, and the method of re-training only for the indistinguishable step will save resources of re-training and enable faster response to the object of the new class. For example, let's say you did not train a mouse on the model you created, and you need to insert a mouse image as an input data to detect that it is a rat. Even if the model can distinguish a large amount of category according to the conventional method, when one more category is added, it takes a long training time to re-learn the entire model. For example, if you have model learning with 9 million images, you will need to relearn the image of the existing 9 million + α (learning image of the added category) when a new category is added. However, in our method, existing model can distinguish that this is Animalia(Kingdom), but it can't distinguish that this is mouse. Then you only need to perform training new model for the lower level.

IV. PERFORMANCE EVALUATION

To prove if the taxonomic method is likely to be implemented, we experimented with three steps of the method described above. Step 1 separates the living_things /non_living_things. Step 2 separates Plantae/Animalia. Step 3 separates Aves/Mammalia. Each step classified into two categories. The reason for this is that we only used more than 40,000 images per class in our dataset for training. Algorithms for every step used Faster R-CNN [13].

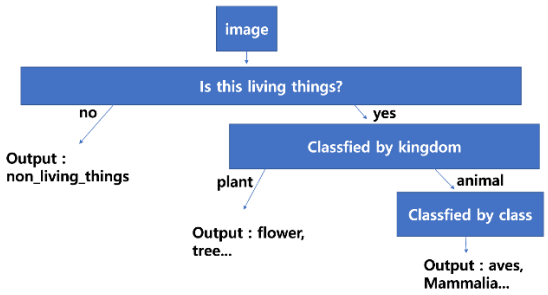

Figure 4: Experimental procedure performed in this paper. Among the methods proposed in proposed methods, image in each step was automatically cropped if needed and passed to the next step.

The basic data used for training was used in the 'PASCAL VOC [24] 2012 dataset' and some of the data downloaded from imagenet's [25] '2013 ILSVRC [26] image data'. We did not have enough images for plants to use for training. To solve this problem, we trained more than 20,000 copies in total, including the entire '102 Category Flower Dataset [27]', 600 images searched in trees on Google, and the plants of the aforementioned 'basic data'. The tree image search on Google and the '102 Category Flower Dataset' were labeled for the entire image area. The annotations are flower and tree, respectively. We added 100 images that are searched by google as 'rock' to recognize the rocks as creatures. Like labeling of the plant, the labeling of the rocks was to the entire area of the image. Compared to dataset labeled by others such as PASCAL VOC or Microsoft COCOS, our labeled images seems to be a little bit fewer than those dataset. However, when we consider the total number of images per category used in the test, it is enough to exceed 20,000. In addition, there are fewer than 100 sub-category images in the Dataset we used. Therefore, the number of images we have added arbitrarily is never small quantities.

The total number of living things used for learning is 61, and the number of inanimate objects is 144. The full list of categories used for learning is in Appendix Table 17 and Table 18. The green-colored letters correspond to the plant, and the red-colored letters indicate labeling, which is labeled the entire area of the image. In fact, the red color of "Living_things" corresponds to the plant.

Since multiple datasets were used as one, the criteria for classifying categories in the categories table such as appendix Table 17 and Table 18 is necessary. The criteria for classification is as follows. Basically it was based on what was labeled in each dataset. However, the dataset downloaded from ImageNet [25], with multiple categories in the same item, was only listed as the <name> category, such as Figure *5*.

Therefore, in appendix A total of 144 "non_living_things" category lists. Potted plants are classified as non-living because of their greater potency than plants in the label area.

, A ball of basketball and a golf_ball were classified respectively without tying to the ball category. However, birds were classified only as bird without classifying each by species.

From the Category list of appendix Table 17 and Table 18, we have picked out only the items that fit each step and have trained them. For example, in step 1 we used all of the category in the "Living_things" given in Table 17 and "Non_living_things" in Table 18. In step 2 we trained by plant

and Animal in the “Living_things” given in Table 17. Step 3 trained by Aves and Mammalia from Animal images in step 2.

```
<annotation>
        <folder>n07697313</folder>
        <filename>n07697313_235</filename>
        <source>
                <database>ILSVRC_2013</database>
        </source>
        <size>
                <width>500</width>
                <height>375</height>
        </size>
        <object>
                <name>n07697100</name>
                <subcategory>n07697313</subcategory>
                <bndbox>
                        <xmin>106</xmin>
                        <xmax>404</xmax>
                        <ymin>42</ymin>
                        <ymax>311</ymax>
                </bndbox>
        </object>
</annotation>
```

Figure 5: some xml content of the data set downloaded from imagenet. In this case, n07697100 in <name> means ‘hamburger’, n07697313 in <subcategory> means ‘cheeseburger’, so only the value of the <name> tag corresponding to the upper category was classified. Other database’s xml has only one name tag, so we were needed criteria to count categories.

TABLE II. THE NUMBER OF IMAGES TRAINED IN STEP 1.

	Living_things	Non_liivng_things	sum
Total	143,036	143,028	286,064

TABLE III. THE NUMBER OF IMAGES TRAINED IN STEP 2.

	Plant	animal	Sum
Total	21,154	21,106	42,260

TABLE IV. THE NUMBER OF IMAGES TRAINED IN STEP 3.

	Aves	Mammalia	sum
Total	37,025	37,072	74,097

The number of images for learning in step 2 is less than the one in step 3. The reason for this is that the number of Plant is too small compared to the number of Animal in the living_things. To solve this problem, we reduced the number of Animal to match the number of Plant, so the total amount of learning data was reduced.

This test used part of the ILSVRC [26] 2014 dataset. The ILSVRC 2014 dataset is different from the ILSVRC 2013 dataset. Thus, it is fine for testing.

Step 1 is the step of separating living and nonliving things. There were many errors in the test because of labeling of objects in dataset. Therefore, we reduced the threshold for the test from 0.8 to 0.4 to detect objects. In other words, it was tested by lowering the accuracy than before. When determining the test results, the criteria are as follows. Even if there are multiple objects in a single Bbox, one object is detected. At this time, it is based on the judgment that more areas are included in the Bbox. If the same object is detected as duplicate, it was determined that the other object was detected. There are at least 6 objects of the same category in a photograph, or the same object has been caught in a bbox too finely more than 6 times. These were only counted to 5. When we tested 9999 images based on this determination and confirmed 300 of them, the results table is the same as Table 5.

TABLE V. TABLE OF THE STEP 1 RESULTS OF THE FIRST 300 TESTS. ROUNDED OFF TO THE THIRD DECIMAL PLACE.

	TRUE	FALSE	detect_fail	sum
total	622	274	54	917
ratio(%)	67.83	29.88	2.29	100

The whole of 9999 images were tested in 8477.55 seconds rounded up from the third decimal spot. Some of the test results are Figure 6.

What is difficult to determine in step 1 is that a person is dressed, as shown in Figure 7. Therefore, in the case of Figure 7, a person with the big proportion of clothing in the bbox area was judged to be non_living_things because it detected the clothing. However, since the person was not detected, it was added to the number of undetected (detect_fail in the table). The reason for this problem is that the clothing such as swimsuits trained to non_living_things objects, and trained person into living_things. To solve this problem, person should be trained only the body parts such as the hands, feet, and face of the person except clothes or clothing should be removed from a category.

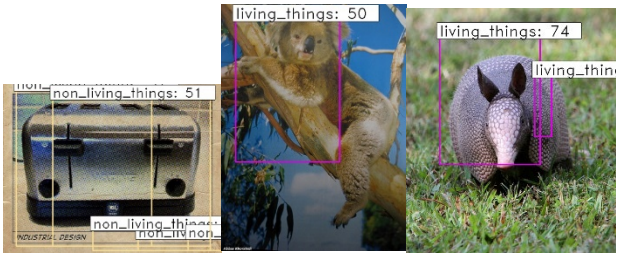

Figure 6: Image of the results of step 1. It can be seen that the accuracy is very low, such as duplicate detection and bbox detection including only a part.

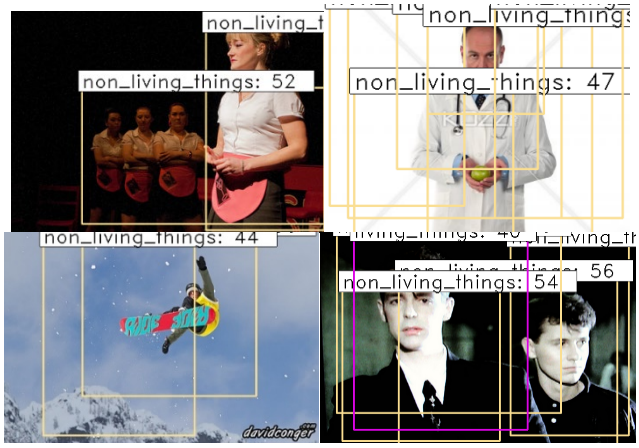

Figure 7: people wearing a cloth in step 1

Step 2 is the distinction between Plant and Animal. For the same reason as step 1, we lowered the threshold to 0.4 to test. Using the same judgement as step 1, but the detection of objects that are not applicable to plants and animals was excluded from the number of counts. Originally, this method intended to deliver the detected bbox in step 1 into step 2. However, the results of the step 1 were not clear. For example, in step 1 an object was detected as a plant but in step 2 the bounding box which was detected as a plant was detected as a material. After 9999 images were tested using these criteria, we took 300 images of them to show the result in .

TABLE VI. 300 OF THE RESULT OF STEP 2 IMAGES CORRESPONDING TO STEP 2 ARE TEST CONFIRMATION TABLES. ROUNDING OFF FROM THE THIRD DECIMAL PLACE

	TRUE	FALSE	detect fail	sum
total	621	86	134	841
ratio(%)	73.84	10.23	15.93	100

9999 images were tested in 8346.35 seconds, rounded up from the third decimal spot. Some of test results are in Figure 8.

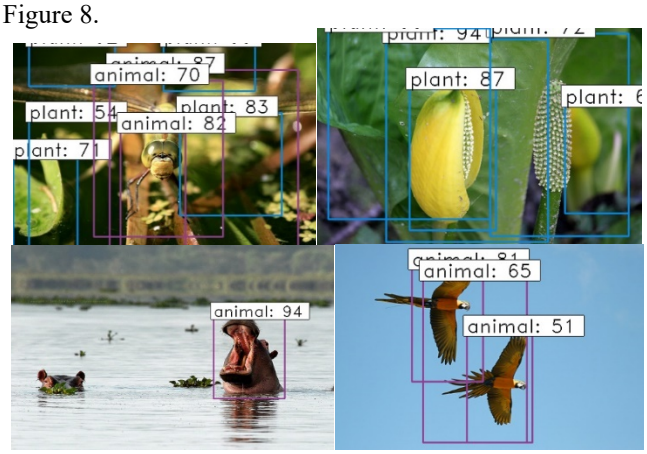

Figure 8: Test results from stage 2.

The top-right image in Figure 8 shows that a blurred fruit was detected as a plant. This is caused by labeling the entire area of the image, not just the plant, while creating the plant dataset.

Step 3 is the distinction between Mammalia and Aves. When 300 images are tested using the same judgement as step 2, the result table is shown in Table 7.

TABLE VII. 300 OF THE IMAGES CORRESPONDING TO STAGE 3 ARE TEST CONFIRMATION TABLES. ROUNDING OFF FROM THE THIRD DECIMAL PLACE

	TRUE	FALSE	detect fail	sum
total	554	97	108	759
ratio(%)	72.99	12.78	14.23	100

Some of the test results are shown in Figure 9. Compared to steps 1 and 2, a clear picture produces good results, but it is still generally less accurate if it is blurry or unclear, such as a tiger photo in Figure 9. Occluded objects, such as a picture in row 2 in Figure 9, have also been detected correctly. However, there are still many miss-detectable cases. In particular, the more complex the background, such as the photos in the second and third rows in Figure 9, the more miss-detection the part. However, the category-specific features are not diverse, such as step 1 or 2, it seems to show better results than the previous step to show similar features.

As another experiment, we took care of images that were not trained or a category similar to the learned category, but not included in the training data. This experiment was tested with 90 images downloaded from Pixabay. This experiment was also used as the same as the above experiments to determine the results. The criterion is that “even if there are multiple objects in a single Bbox, one object is detected. It is considered as many areas in the Bbox are included as a judgment criterion. If the same object is detected as duplicate, it is determined that the other object is detected. If there are more than six objects of the same category in a photograph, or the same object is caught more than six times, it will be counted only up to five.".

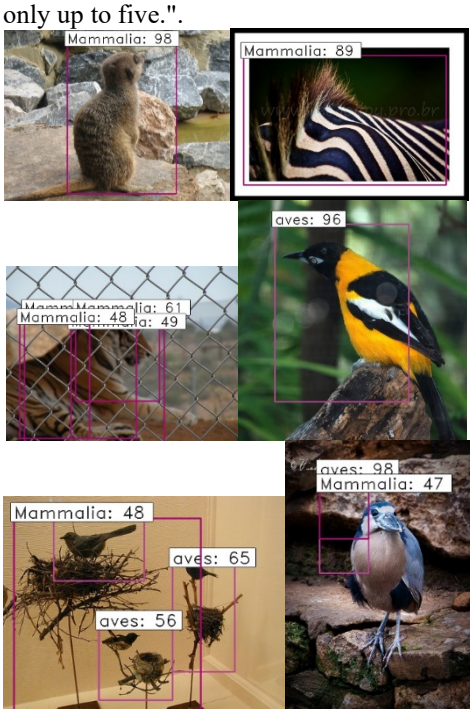

Figure 9: Test Results from stage 3.

This experiment was also used as the same as the above experiments to determine the results. The criterion is that “even if there are multiple objects in a single Bbox, one object is detected. It is considered as many areas in the Bbox are included as a judgment criterion. If the same object is detected as duplicate, it is determined that the other object is detected. If there are more than six objects of the same category in a photograph, or the same object is caught more than six times, it will be counted only up to five." Table 8 is for the step-1 result table, Table 9 for the step-2 result table, and Table 10 for the step-3 result table.

TABLE VIII. THE RESULT TABLE OF THE ORGANISM DURING THE STAGE 1 TEST WITH AN IMAGE THAT DOES NOT LEARN OR CONTAINS SIMILAR CATEGORIES. OF THE ABOVE CATEGORIES, THE CATEGORY BELONGING TO THE TRAINING DATA LIST IS NOT INCLUDED IN OR SIMILAR TO THE TRAINING DATASET. ROUNDED TO 3 DECIMAL PLACES. T IS TRUE, F IS FALSE, M IS MISS IN TABLE COLUMN NAME.

	T	F	M		T	F	M
bat	3	3	0	ibis	4	7	0
beetle	34	35	5	lettuce	5	0	0
Cherry blossom	3	5	0	Long Tailed tit	0	3	0
cheetah	7	1	0	narcissus	0	2	0
Chicken	5	0	0	parrot	9	5	0
cicada	11	0	6	pumpkin	5	6	0
corn	7	0	0	raccoon	10	1	0
daffodil	4	2	0	rat	9	0	0
dragon fruit	14	3	0	sparrow	3	8	0
giraffe	2	0	0	tomato	2	6	0
total	137	87	11	ratio(%)	58.30	37.02	4.68

TABLE IX. TABLE OF INANIMATE RESULTS DURING THE STAGE 1 TEST WITH IMAGES THAT DO NOT LEARN OR CONTAIN SIMILAR CATEGORIES. ROUNDED OFF TO THE THIRD DECIMAL PLACE. INANIMATE OBJECTS BELONGING TO THE TRAINING DATASET ARE EXTENSIVE, SO THE NUMBER

OF CATEGORIES TESTED IS MINIMAL TO MINIMIZE OVERLAP. T IS TRUE, F IS FALSE, M IS MISS IN TABLE COLUMN NAME

	T	F	M		T	F	M
butter	9	4	0	statue	7	7	0
Honeycomb	10	1	0	stone	5	0	0
popcorn	11	1	0				
total	42	13	0	ratio(%)	76.36	23.64	0

TABLE X. A RESULT TABLE INCORPORATING THE RESULTS OF LIVING AND INANIMATE OBJECTS IN STAGE 1. ROUNDING OFF FROM THE THIRD DECIMAL PLACE.

	TRUE	FALSE	MISS	SUM
total	179	100	11	290
ratio(%)	61.72	34.48	3.79	100

The total time tested for 90 images in step 1 was 547.32 seconds rounded up from the third decimal place. Part of the test results is shown in Figure 10. As shown in Figure 10, 'Human-shaped statues' are sometimes detected as creatures, but they are more often perceived as inanimate objects. Of course, it is necessary to solve the problem of the first step of experiments which are shown in Figure 6 with ILSVRC 2014 dataset as it is.

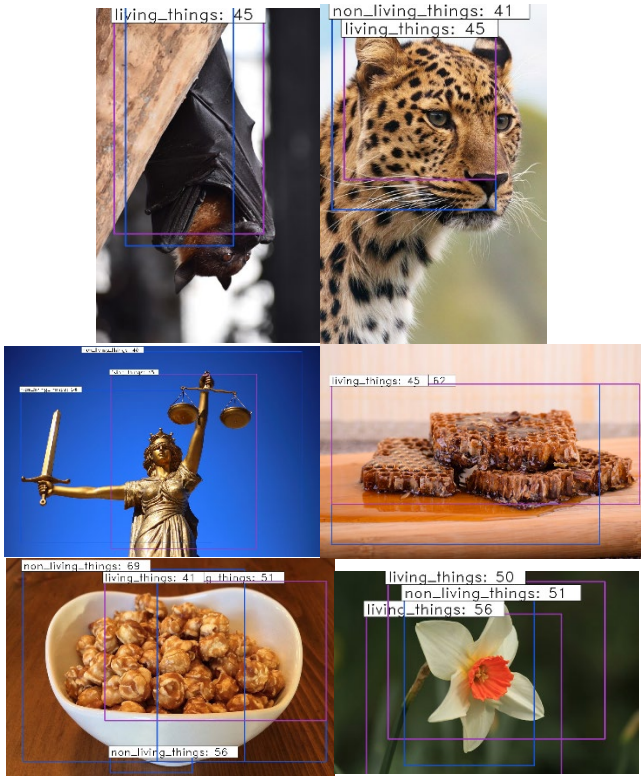

Figure 10: The stage 1 result with unlearned or similar categories. Objects in the background are often recognized, so there are many boxes, but in that case, only the class corresponding to the table above was determined. Blue bbox is inanimate, purple bbox is animate.

TABLE XI. A RESULT TABLE INCORPORATING THE RESULTS OF ANIMALIA OBJECTS IN STAGE 2. ROUNDING OFF FROM THE THIRD DECIMAL PLACE. T IS TRUE, F IS FALSE, M IS MISS IN TABLE COLUMN NAME.

	T	F	M		T	F	M
bat	11	0	0	ibis	11	1	0
beetle	49	0	5	Long Tailed tit	5	0	0
cheetah	4	0	1	parrot	11	6	0
chicken	5	0	4	raccoon	13	0	0
cicada	8	0	6	rat	5	0	0
giraffe	2	0	0	sparrow	7	0	0
total	131	7	16	ratio(%)	85.06	4.55	10.39

TABLE XII. A RESULT TABLE INCORPORATING THE RESULTS OF PLANTAE OBJECTS IN STAGE 2. ROUNDING OFF FROM THE THIRD DECIMAL PLACE. T IS TRUE, F IS FALSE, M IS MISS IN TABLE COLUMN NAME

	T	F	M		T	F	M
Cherry blossom	0	7	0	lettuce	5	0	0
corn	6	4	0	narcissus	1	1	0
daffodil	1	0	1	pumpkin	8	3	0
dragon fruit	10	5	0	tomato	3	3	0
total	34	23	1	ratio(%)	58.62	39.66	1.72

TABLE XIII. A RESULT TABLE INCORPORATING THE RESULTS OF ANIMALIA AND PLANTAE OBJECTS IN STAGE 2. ROUNDING OFF FROM THE THIRD DECIMAL PLACE.

	TRUE	FALSE	MISS	sum
total	165	30	17	212
ratio(%)	77.83	14.15	8.02	100

In step 2, we tested with data that we experimented in step 1 with 76 images, excluding inanimate objects.

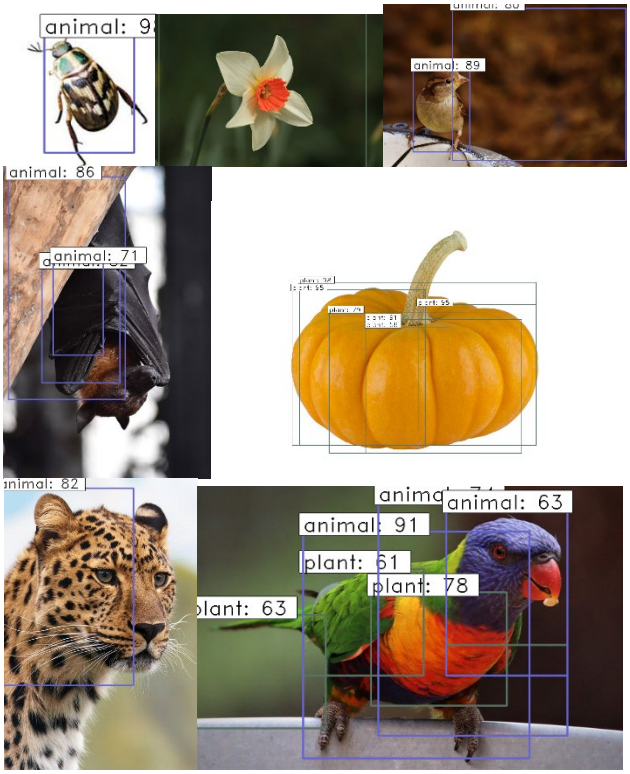

Figure 11: Results in step-2 including alike category which is not trained. Bounding box with green color means plant and blue for animal.

The results of the step 2 experiment are on Table 11, Table 12, and Table 13. In some cases, background detection in a step 2 such as Figure 11, has a problem that should be solved by deleting the background as proposed.

In step 3, we selected only birds and mammals from the data tested in step 2, and test with 25 images. The results of

the experiment are in, Table 14, Table 15, Table 16, and in Figure 12.

TABLE XIV. A RESULT TABLE OF MAMMALIA DURING A STAGE 3 TEST WITH IMAGES THAT DO NOT LEARN OR CONTAIN A SIMILAR CATEGORY. ROUNDING UP FROM THE THIRD DECIMAL PLACE.

	TRUE	FALSE	MISS
bat	1	4	0
cheetah	2	0	2
giraffe	1	2	0
raccoon	3	0	0
rat	4	0	0
total	11	6	2
ratio(%)	57.89	31.58	10.53

TABLE XV. *A RESULT TABLE OF AVES DURING A STAGE 3 TEST WITH IMAGES THAT DO NOT LEARN OR CONTAIN A SIMILAR CATEGORY. ROUNDING UP FROM THE THIRD DECIMAL PLACE.*

	TRUE	FALSE	MISS
chicken(live)	1	2	0
ibis	5	1	0
long-tailed_tit	4	1	0
parrot	8	2	0
sparrow	5	0	0
total	23	6	0
ratio(%)	79.31	20.69	0

TABLE XVI. *A TABLE OF STAGE 3 RESULTS INCORPORATING THE RESULTS OF MAMMALIA AND AVES. ROUNDING UP FROM THE THIRD DECIMAL PLACE.*

	TRUE	FALSE	MISS	sum
total	34	12	2	48
ratio(%)	70.83	25.00	4.17	100

As shown in Figure 12, if there is an object camouflaged or disguised as a cheetah image, it is often undetectable. However, this problem can be solved by using the method suggested by the proposed methods, which are 'cropping bbox step-by-step and passing it on to the next step'.

V. CONCLUSION

Through the two kinds of proposed experiments, it is also possible to classify through the model for each layer. However, in order to accurately carry out our proposal, the following things must be solved. it is required more high-quality learning image dataset, specific and systematic detection criteria (e.g., whether a person's clothing should be judged as an object when detecting a person).

If the amount of learning data is sufficient, we use the entire classification layer of 466,327 species [20][21][22] as learning data. This will allow us to distinguish between most creatures on earth. By establishing a standard of taxonomy classify that can distinguish the characteristics of inanimate objects well, and learning in a similar way as above, you will be able to distinguish between inanimate objects too.

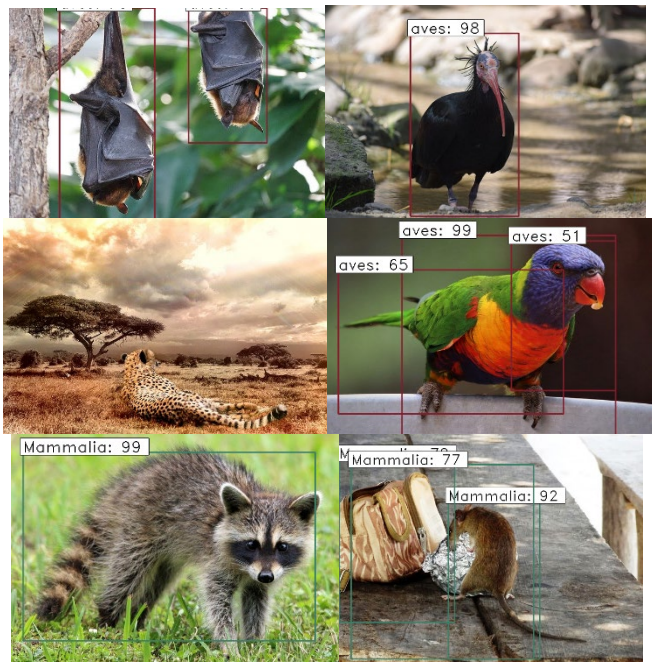

Figure 12: A stage 3 result that includes unlearned categories, etc. Flying creatures like bats are often recognized as birds, or when there are many objects around them, such as rats, the error is large, or protected creatures such as cheetahs are often not recognized. Red bboxes are Aves, green bboxes are Mammalias.

In this experiment, the bbox was not accurately caught in each step. In particular, the first step of the first experiment was also low in accuracy of 67.83% (Table 5) could not be properly classified because we annotated images in rough.

REFERENCES

[1] Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton. "ImageNet Classification with Deep Convolution Neural Networks". In Advances in Neural Information Processing Systems (NIPS), (1097-1105). (2012).

[2] Tan, M., & Le, Q. V. (2019). Efficientnet: Rethinking model scaling for convolutional neural networks. *arXiv preprint arXiv:1905.11946*.

[3] Deng J., Berg A.C., Li K., Fei-Fei L. (2010) "What Does Classifying More Than 10,000 Image Categories Tell Us?". In: Daniilidis K., Maragos P., Paragios N. (eds) Computer Vision – ECCV 2010. ECCV 2010. Lecture Notes in Computer Science, vol 6315. Springer, Berlin, Heidelberg

[4] Hastie, T., Tibshirani, R., & Friedman, J. (2017). *The elements of statistical learning: data mining, inference, and prediction*. Springer Science & Business Media.

[5] Bishop, C. M. (2006). *Pattern recognition and machine learning*. springer.

[6] Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine learning*, *20*(3), 273-297.

[7] K. He and J. Sun, "Convolutional neural networks at constrained time cost," *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, 2015, pp. 5353-5360, doi: 10.1109/CVPR.2015.7299173.

[8] Keon Myung Lee, 『Artificial intelligence: from Turing tests to deep learning (Korean edition)』, Saengneung publisher(2018), p271-274.

[9] Kumar Ayush, Burak Uzkent, Marshall Burke, David Lobell, Stefano Ermon. "Generating Interpretable Poverty Maps using Object Detection in Satellite Images". arXiv:2002.01612v2 [cs.CV] 18 Feb 2020

[10] Redmon, J., & Farhadi, A. (2018). Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*.

[11] Luo, Q., Ma, H., Tang, L., Wang, Y., & Xiong, R. (2020). 3d-ssd: Learning hierarchical features from rgb-d images for amodal 3d object detection. *Neurocomputing*, *378*, 364-374.

[12] X. Liu, H. Wang, H. Jing, A. Shao and L. Wang, "Research on Intelligent Identification of Rock Types Based on Faster R-CNN Method," in IEEE Access, vol. 8, pp. 21804-21812, 2020, doi: 10.1109/ACCESS.2020.2968515.

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

[14] Z. Lu, J. Lu, Q. Ge and T. Zhan, "Multi-object Detection Method based on YOLO and ResNet Hybrid Networks," 2019 IEEE 4th International Conference on Advanced Robotics and Mechatronics (ICARM), Toyonaka, Japan, 2019, pp. 827-832, doi: 10.1109/ICARM.2019.8833671.

[15] J. Redmon and A. Farhadi. "YOLO9000: Better, faster, stronger". In CVPR, 2017.

[16] C. Kim *et al*., "Implementation of Yolo-v2 Image Recognition and Other Testbenches for a CNN Accelerator," *2019 IEEE 9th International Conference on Consumer Electronics (ICCE-Berlin)*, Berlin, Germany, 2019, pp. 242-247.

[17] Chetlur, S., Woolley, C., Vandermersch, P., Cohen, J., Tran, J., Catanzaro, B., & Shelhamer, E. (2014). cudnn: Efficient primitives for deep learning. *arXiv preprint arXiv:1410.0759*.

[18] B. Wang, Y. Sun, B. Xue and M. Zhang, "Evolving Deep Convolutional Neural Networks by Variable-Length Particle Swarm Optimization for Image Classification," 2018 IEEE Congress on Evolutionary Computation (CEC), Rio de Janeiro, 2018, pp. 1-8, doi: 10.1109/CEC.2018.8477735.

[19] Ruggiero MA, Gordon DP, Orrell TM, Bailly N, Bourgoin T, et al. (2015) "A Higher Level Classification of All Living Organisms". PLOS ONE 10(4): e0119248.

[20] NCBI. https://www.ncbi.nlm.nih.gov/Taxonomy/taxonomyhome.html/index.cgi?chapter=statistics&?&uncultured=hide&unspecified=hide&m=0. "Taxonomy Nodes(all dates)". Check the exclude uncultured and exclude informal names in the Filters at the bottom, and then use the values of All taxa and species from All dates in the results.2020/Feb/15

[21] Sayers EW, Cavanaugh M, Clark K, Ostell J, Pruitt KD, Karsch-Mizrachi I. GenBank. Nucleic Acids Res. 2019 Jan 8;47(D1):D94-D99.

[22] Federhen S. The NCBI Taxonomy database. Nucleic Acids Res. 2012;40(Database issue):D136-D143.

[23] CARL R. WOESE, OTTO KANDLER, Mark L. WHELLIS. "Towards a natural system of organisms: Proposal for the domains Archaea, Bacteria, and Eucarya" Proc. Nati. Acad. Sci. USA Vol. 87, pp. 4576-4579, June 1990 Evolution

[24] Everingham, M., Van Gool, L., Williams, C. K., Winn, J., & Zisserman, A. (2010). The pascal visual object classes (voc) challenge. *International journal of computer vision*, *88*(2), 303-338.

[25] Yang, K., Qinami, K., Fei-Fei, L., Deng, J., & Russakovsky, O. (2020, January). Towards fairer datasets: Filtering and balancing the distribution of the people subtree in the imagenet hierarchy. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency* (pp. 547-558).

[26] Russakovsky, O., Deng, J., Su, H. *et al.* ImageNet Large Scale Visual Recognition Challenge. *Int J Comput Vis* **115,** 211–252 (2015).

[27] M. Nilsback and A. Zisserman, "Automated Flower Classification over a Large Number of Classes," *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*, Bhubaneswar, 2008, pp. 722-729, doi: 10.1109/ICVGIP.2008.47.

VI. Appendix

TABLE XVII. List of 61 "Living_things" categories in total

ant	Dog	lemon	seal
antelope	dragonfly	lion	sheep
apple	elephant	lizard	skunk
armadillo	Fig	lobster	snail
artichoke	flower	monkey	snake
banana	Fox	orange	squirrel
bear	Frog	otter	starfish
bee	goldfish	panda	strawberry
bell pepper	hamster	person	swine
bird	head cabbage	pineapple	tick
butterfly	hippo	pomegranate	tiger
camel	horse	porcupine	tree
cat	isopod	rabbit	turtle
centipede	jellyfish	ray	whale
cow	koala	scorpion	zebra
cucumber	ladybug		

TABLE XVIII. A total of 144 "non_living_things" category lists. Potted plants are classified as non-living because of their greater potency than plants in the label area.

accordion	dumbbell	Potted plant
aeroplane	Electric fan	Power drill
axe	Ewer	pretzel
Baby bed	Face powder	printer
backpack	File	puck
bagel	flute	Punching bag
Balance beam	Frying pan	Purse
Band aid	Golf ball	Racket
banjo	Golf cart	Remote control
baseball	guacamole	Rubber eraser
basketball	guitar	Rugby ball
Bathing cap	Hair spray	ruler
beaker	hamburger	saltshaker
Bell	hammer	saxophone
bench	harmonica	screwdriver
bicycle	harp	ski
binder	helmet	snowmobile
Blow dryer	Horizontal bar	snowplow
boat	Horn	Soap dispenser
bookcase	Hotdog	Soccer ball
bottle	Icebox	sofa
bow	iPod	spatula
Bow tie	ladle	stethoscope
bowl	lamp	stove
brassiere	laptop	strainer
burrito	lipstick	stretcher
bus	(cooked)Lobster	sunglasses
bicycle	lolly	swimsuit
Can opener	maillot	syringe
car	maraca	table
cart	microphone	Tape player
cello	microwave	Tennis ball
chainsaw	Milk can	toaster
chair	miniskirt	Traffic light

Cocktail shaker	motorbike	train
Coffee maker	mug	trombone
Computer keyboard	nail	tvmonitor
Computer mouse	napkin	unicycle
corkscrew	Neck brace	vacuum
Cowboy hat	oboe	vessel
Cream	Pencil case	violin
Croquet ball	Pencil sharpener	volleyball

Crutch	perfume	Waffle iron
digital clock	piano	washer
dining table	pizza	Water bottle
dishwasher	Plastic bag	Windsor tie
display	Plate rack	Wine bottle
drum	Pot	rock