
Image Classification On Super And Sub Classes With Unseen Classes

Evan Ting-I Lu, Lawrence Li
Columbia University
COMS 4995 Neural Network Deep Learning
tl3098@columbia.edu, ll3598@columbia.edu

Abstract

1 Image classification is one of the most important applications in computer vision,
2 where deep learning frameworks categorize and label groups of pixels and vectors
3 within an image based on specific patterns and extracted features. This project
4 proposes classification models for three image superclasses and corresponding sub-
5 classes. To approach the problem, a super-class guided network approach (SGNet)
6 is proposed to integrate superclass features into the prediction of subclasses, which
7 can be implemented using pre-trained models such as VGG and ResNet. In addition
8 to standard training techniques like data augmentation and weight decay, our
9 approach will also combine the image super-resolution network ESRGAN and the
10 SGNet to improve the classification accuracy for low resolution images.

11 1 Introduction

12 In this project, we will build an image classification system that can predict 3 super classes: bird,
13 dog, and reptile, and their corresponding sub classes. The competition provides a dataset consisting
14 of 6472 training images and 9127 test images. We will be doing training based on the training set
15 and testing on the test sets. Each image in the dataset is a standard RGB colored image with 8×8
16 low resolution. Our main evaluation metrics are the classification accuracies on superclasses and
17 subclasses, where novel class samples are included in subclasses.

18 2 Related Works

19 There are numerous pre-trained convolutional neural models that can be used to classify images into
20 classes, such as the ResNet series, VGG series, and MobileNet series. These models have been trained
21 using the ImageNet database and achieved state-of-the-art top-1 and top-5 classification accuracy.
22 The ImageNet database is very similar to the database used in competition where it requires models
23 to extract features and predict the primary object in an image. These convolutional neural models
24 have performed well in the classification of image datasets including CIFAR-100 and ImageNet.

25 Regarding the classification of superclasses and subclasses, one related work is the use of super-class
26 guided networks (SGNet). Inspired by how human beings learn to identify an object, SGNet proposes
27 a superclass branch (SCB) that can be directly plugged into any existing or pre-trained model. This
28 branch is trained together with the base model to guide the finer class training while extracting and
29 learning the superclass features. The SCB branch should be shorter than the FCB branch because
30 of the relative ease of super-class classification. The SCB method for SGNet is a superclass related
31 fine-tuned network that can be used for any pre-trained models mentioned above. Figure 1 shows the
32 basic architecture of super-class guided network where the superclass feature is concatenated into
33 subclass via skip connection before reaching the output layer.

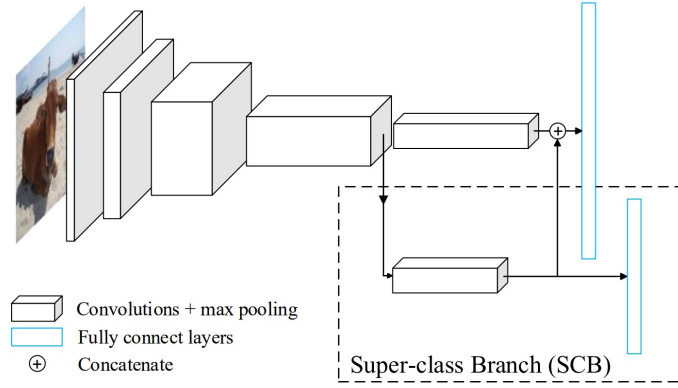


Figure 1: Architecture of super-class guided convolutional neural network. [2]

Besides the main idea of superclass guided network architecture, the paper also purposed two methods of inference for the SGNet model. The first method is a two-step inference (TSI) where the model first predicts superclass, and then finds the highest confidence score in the corresponding subclasses. The second method is a direct inference (DI) where the model directly generates classification results from the subclass branch. Comparing both methods of inference, the paper found out that from all test results at the end of each epoch, the direct inference method consistently outperforms the two-step inference method. One important aspect from the test results is that if the superclass accuracy is not sufficiently high, then direct inference method will achieve better performance overall.

Researchers have adopted this superclass guided method on a pre-trained VGG16 model and improved the classification accuracy by a minimum of 0.69% and maximum of 1.50% using the same ImageNet database. [2]

Since our dataset consists of 8×8 low resolution images, we planned to apply super-resolution technique. Super-resolution is a concept that adds details to an image and thus increases the resolution. It has long been adopted in the computer vision field. Lim et. al. proposed the Enhanced Deep Residual Networks (EDSR) that out performed then-state-of-the-art SR methods for single image super-resolution. They contribute the significant performance boost to three major factors: 1. removal of unnecessary modules in the residual networks, 2. increment of model size 3. new multi-scale deep SR system, MDSR, which is capable of reconstructing high definition images. [1] In our work, we incorporated the EDSR model to upsample our images before feeding into the model, in hope of the higher resolution may help model achieve better performance.

3 Conjecture

Based on the existing researches, we propose the following hypothesis:

When combining the methods of superclass guided network architecture and super-resolution image resampling, we will achieve better performance overall than the base model.

4 Methods

4.1 Overview

We implemented a fine-tuned version of SGNet as mentioned above and used super-resolution image method to improve the overall model performance.

To begin with, we trained a simple combination of CNN + MLP models as a baseline, i.e., one for super class and the other for subclass prediction. The baseline model consists of a few convolution layers followed by dense layers and a softmax classification layer. We utilized super-resolution models, specifically, the EDSR (a GAN-based) model to enhance input image resolution.

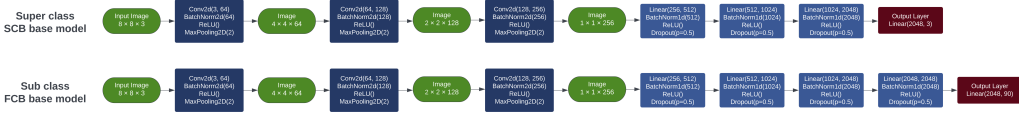


Figure 2: Base model architecture for superclass and subclass image classification. The left side is for superclass (SCB) and the right side is for subclass (FCB)

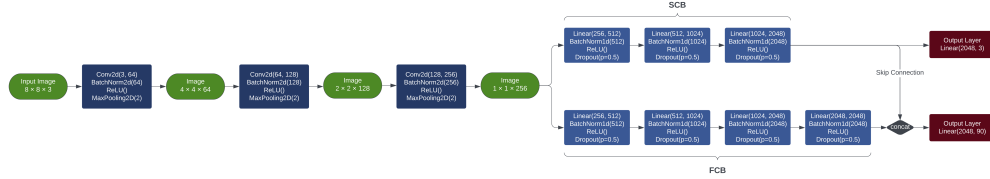


Figure 3: Super-class guided model architecture for image classification. The SCB is the super-class branch used to predict the super class label and the FCB is the finer-class branch used to predict the sub class label.

As mentioned in the related works, we used fine-tuned based superclass guided convolutional networks to classify images into corresponding superclasses and subclasses. This is basically a fine-tune method but we are adding a parallel convolutional block to classify the superclasses followed by a separate fully-connected layer. To train the model, we will first fine-tune the pre-existing model and add a SCB to train the weights for superclasses. Then, we freeze the bottom convolutional blocks including SCB to train the remaining weights for subclasses. Here, the features from superclass will be concatenated to the prediction of subclasses. Finally, we unfreeze all layers and train the whole network.

4.2 Base Model

Initially we attempted to follow the SGNet paper’s approach and apply transfer learning and fine-tuning using VGG16 and ResNet18 architecture. However, we found out that such model performed poorly on our dataset since the input image resolution is very low. The image is a 8×8 size with 3 color channels. This means that the image is basically a flattened feature array of length 192 ($8 \times 8 \times 3 = 192$). In general, large convolutional neural networks are not suitable for low-resolution images since they can be easily overfitted given a very limited amount of information from the feature vector.

As a result, we will be building our own base model from scratch. Our base model is a convolutional neural network architecture with 3 convolutional layer blocks, 3 fully-connected layer blocks for superclass prediction and 4 fully-connected layer blocks for subclass prediction, and 1 output layer. The input feature will be a 3-channel RGB image in .jpg format. Because of the very-low resolution of input images, we decided to go with kernel size of 5, stride=1, and padding=2 for every 2D convolutional layers in our model. ReLU activation was used in every convolutional layers. Figure 2 shows the base model architecture for predicting superclass and subclass labels.

4.3 SGNet: A super-class guided model

As mentioned in the base model section, we have 3 fully-connected layer blocks for superclass prediction and 4 fully-connected layer blocks for subclass prediction. Following our base model architecture, we combined the SCB and FCB components from base models into a super-class guided model in order to improve the performance on subclass label. As we suspect that our superclass accuracy may not be sufficiently high, we decided to apply direct inference method (DI) where we will predict the subclass label without predicting its corresponding superclass label first. Figure 3 shows the super-class guided model architecture expanded from the 2 base models.

96 We train the super-class guided network by following the conventional approach in fine-tuning.
 97 The algorithm box 1 explains how we train and fine-tune the super-class guided network for better
 98 prediction for subclass labels.

Algorithm 1 Training super-class guided network

Require: A SGNet with SCB and FCB component initialized.

Train the super-class SCB base model from scratch, save the model with weights.

Copy all the weights from SCB base model to corresponding SCB component in SGNet.

Freeze all the weights in the SGNet except the FCB component and the output layer for subclass predictions.

Train the SGNet on subclass labels, only the weights from FCB component and output layer are updated.

Unfreeze all layers including the SCB component.

Train the SGNet again and update all weights for better results.

99 **4.4 Super-resolution image re-sampling**

100 Because of the extremely-low image resolution in this task, we decided to adopt super-resolution
 101 method to up-sample the images. Specifically, we used the EDSR model, proposed by Lim et. al.,
 102 which achieved robust results in various benchmark datasets. The PyTorch implementation we used
 103 is TorchSR, contained in the Python Package Index (PyPI). Other than EDSR, the package includes a
 104 few SR methods implementations, such as RCAN, NinaSR, etc. After some pilot experiments, we
 105 found the EDSR model seemed to deliver better results, potentially due to its significantly larger size
 106 as advertised by the original authors. Thus, we decided to use EDSR as our SR method.

107 Initially, we implemented super-resolution during data augmentation, i.e., writing self-defined class
 108 such that it can be called as if other transform functions in the torchvision package. However, this
 109 approach significantly increases the model training time. Therefore, we later decided to generate
 110 all upsampled images beforehand, thereby saving much time during training and evaluation. We
 111 upsampled the image using the EDSR model with a upscaling factor of 4, making the image resolution
 112 from 8×8 to 32×32 . Figure 4 shows the comparison between original low-resolution images and 4
 113 times scaled-resolution images from training set. In this figure, 4 images are randomly chosen as
 114 examples for comparison.

115 For the base models, we tweaked the architecture to better extract the features from the newly 32×32
 116 super resolution image. We added two additional convolutional block and increased the number of
 117 features to learn for the fully-connected layer blocks. Figure 5 and 6 shows the tweaked architecture
 118 for the super-resolution image method.

119 **4.5 Data preprocessing and training**

120 We used data augmentation method by randomly performing horizontal flip of an image with a
 121 probability of 0.5. In addition, we followed the data augmentation approach from the Trivial
 122 Augment paper [3] where we randomly change the contrast and lightness of the image to better
 123 improve the robustness of the model. The trivial augmentation method is mainly a contrast and
 124 lightness augmentation method applied with a randomized strength factor from 0 to 30. This
 125 method was experimented in the training of CIFAR-100 dataset and outperformed most other
 126 data augmentation methods in test accuracy, recall, and precision. Since the training images all
 127 have low-resolution, any data augmentation methods including blurring, cropping, padding, and
 128 perspective-related transformations may not be effective in improving model generalizability. In fact,
 129 these data augmentation techniques had caused our model to underfit the data and still unable to
 130 improve the validation and evaluation loss after training for long epochs. After the data augmentation,
 131 we normalize all images using the mean and standard deviation of ImageNet dataset since our dataset
 132 consists of ordinary photos of natural scenes, specifically animals, which corresponds to the nature of
 133 ImageNet database.

134 For hyper-parameters, we choose batch size as 64 and Adam as our optimizer. We choose the learning
 135 rate to be 1×10^{-4} with no weight decay since we found that dropout and batch normalization layers
 136 in our model already effectively regularizes both training and validation loss.









Original 8 × 8	Super Resolution Scaling factor = 4 32 × 32	Label
		1: Dog 63: Afghan hound, Afghan
		0: Bird 30: Vulture
		1: Dog 7: Pekinese, Pekingese, Peke
		2: Reptile 59: African crocodile, Nile crocodile, Crocodylus niloticus

Figure 4: Comparison between original low-resolution image and 4 times scaled-resolution image with corresponding labels



Figure 5: Base model architecture for superclass and subclass image classification for super-resolution method. The left side is for superclass (SCB) and the right side is for subclass (FCB)

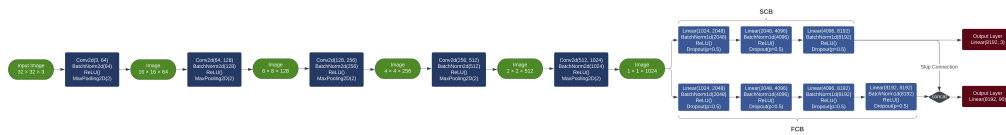


Figure 6: Super-class guided model architecture for super-resolution image classification. The SCB is the super-class branch used to predict the super class label and the FCB is the finer-class branch used to predict the sub class label.

We applied K-Fold validation training process where we first separate the 6472 training images into 5954 image training set and 518 image evaluation set. Within 5954 training sets we divide them into 13-fold blocks, so each fold block has a length of 458 images. We trained a total of 13 models by each choosing a fold block as validation set and the rest of them as training set. We will evaluate the performance from 13 models on the 518 images evaluation set and obtain the local test accuracy from the training set. Finally, we will submit the model with the best local test accuracy to evaluate the final performance on online testing set.

For the novel class prediction under subclass, we decided to apply a manually-set threshold to the subclass predictions based on the softmax probability outputs. If the maximum probability of predictions on subclasses is under a certain threshold, then we classify that label as novel class. In other words, we classify unassured results predicted by our super-class guided model to better improve the performance for classifying novel classes.

5 Results

Here we will report the local evaluation accuracy from the training set (518 image evaluation set). We include the performance for two base models plus the super-class guided model with and without super-resolution resampling. We choose the highest accuracy, the lowest accuracy, and the mean accuracy from 13 models we trained from KFold cross validation.

5.1 Model performance without super resolution

The first table shows the evaluation accuracy on local test set from training set (518 images). Note that the SG model pertains only to subclass classification.

Class Label	Highest	Lowest	Mean
Base model superclass	80.23%	73.12%	76.12%
Base model subclass	20.29%	14.69%	17.21%
SG model subclass	21.37%	15.73%	18.82%

The second table shows the final test accuracy on online test set. All subclass models are evaluated using a confidence threshold of 0.70 for novel class prediction.

Class Label	Accuracy	F-1 score	Precision	Recall
Base model superclass	70.33%	70.00%	70.09%	71.13%
Base model subclass	63.43%	3.66%	11.02%	2.77%
SG model subclass	67.51%	3.27%	11.03%	2.49%

5.2 Model performance with super resolution

The first table shows the evaluation accuracy on local test set from training set (518 images).

Class Label	Highest	Lowest	Mean
Base model superclass	85.14%	79.92%	82.86%
Base model subclass	22.20%	18.73%	20.85%
SG model subclass	23.36%	19.69%	21.51%

The second table shows the final test accuracy on online test set. All subclass models are evaluated using a confidence threshold of 0.70 for novel class prediction.

Class Label	Accuracy	F-1 score	Precision	Recall
Base model superclass	75.31%	75.02%	74.65%	78.04%
Base model subclass	69.02%	2.82%	9.04%	2.22%
SG model subclass	70.26%	2.51%	10.36%	2.09%

5.3 Sub class model performance with novel class threshold

The following table shows how the performance of the model varied across different threshold value for novel class prediction. All models are based on the super-class guided model from super-resolution resampling.

Threshold	Accuracy	F-1 score	Precision	Recall
0.70	70.26%	2.51%	10.36%	2.09%
0.50	52.45%	5.04%	8.07%	4.14%
0.30	37.25%	5.49%	7.39%	5.11%
0.10	3.26%	4.98%	6.18%	6.39%
No threshold	3.24%	4.98%	5.07%	6.39%

6 Discussion

6.1 Overall performance

Our baseline model achieved 80.2% super class accuracy on local test set and 70.3% on competition test set, which is not to surprising considering there are only three super classes, making the classification task relatively simpler compared to the 90-subclass task. Our CNN-based model proved to be useful even for such ultra-low resolution images, showcasing CNN’s ability to capture patterns and structures. While we do not have access to the online (competition) data set and thus could not infer with total confidence, the significant performance drop from local to online tests could be due to the more varied and complex images in the latter setting.

Both SGNet and Super-resolution proved to be effective, with the former contributed to around 1% improvement on accuracy and the latter 3%~ 5%. More to be discussed in the coming sections.

6.2 SGNet

SG model, our improvement over the baseline model, turned out to be effective, with around 1% improvement over its counterpart. This meets our expectation because the concept of SGNet is using the potential useful feature learnt from super classes to guide the subclass classification, as opposed to adding some miraculous novel information about sub classes. Compared to the improvement claimed by the original paper which is around 0.69% (over their SGNet models based on VGG16), we consider our adoption of this idea successful. Since the performance of super-class on online test set is not accurate enough, we suspect that replacing our current direct inference method to a two-step inference method may not achieve better performance on subclasses.

6.3 Super resolution

If our SGnet approach achieved some success, the super resolution approach brought even bigger improvement on our local test set. Comparing models with or without resolution, we found significant performance boost with the aid of super-resolution across all performance metrics. Refer to Figure 4, where we showed a few images before/after super-resolution, one can tell even with bare eyes that the upsampled versions deliver a much better sense of the object. Though such enhancement is somewhat qualitative and can’t be measured directly, it seems the model was able to interpret and capture the underlying structures much better from the 32 x 32 images than the raw 8 x 8 counterparts. On our local test set, models trained with super-resolution images brought around 5% accuracy improvement to super classes and around 2% for sub classes.

197 For online (competition) set, super-resolution brought significant improvement to super classes across
198 all metrics, while the sub classes also improvement over accuracy, with some sacrifice on the other
199 metrics, such as F1, precision, and recall rates. Again, we couldn't infer directly what happened
200 as the lack of access to the dataset. However, combined with our other observations that we will
201 address later, we believe the competition dataset is somewhat imbalanced. This leads us to believe
202 the performance drop was due to the imbalance nature rather than the adoption of super-resolution.
203 Overall, we'd consider super-resolution a good approach for our specific task with some more rooms
204 for discussion around the imbalance issue, which in itself is a big and complex topic.

205 **6.4 Threshold vs. Sub class performance**

206 First of all, we saw that the accuracy metric was not telling the whole story when taking into account
207 other metrics. This is where we were convinced of the imbalance nature of the online test dataset.
208 Refer to section 5.3, we saw that changing threshold has dramatic impacts on model's performance.
209 Without any threshold, the model performed poorly on almost every metric, which is reasonable
210 because it won't correctly classify any novel class instances. We started to observe significant
211 accuracy improvements with threshold >0.30 , meaning if the probability that our model considers an
212 instance to belong to a class is smaller than 30%, the instance would be classified as novel class.

213 As we kept on push the threshold, we saw very interesting effects that accuracy enhanced dramatically
214 as threshold is raised, while the other metrics fluctuated. Threshold = 0.5 might be a good balanced
215 point as it provides decent accuracy (around 52%) while maintained a relatively good performance
216 for imbalance metrics, i.e., F-1 score, precision, and recall rates.

217 We suspect there are a considerable amount of novel sub class instances in the competition dataset,
218 thereby causing accuracy to increase significantly with the raise of threshold. Beyond that, if our
219 suspicion is correct that the test set is imbalanced. Then, it is essentially a trade off between different
220 metrics and which model to go for is largely domain and context dependent that will require more
221 prior knowledge or domain expertise. Since the imbalanced novel class distribution does not appear in
222 the training set (We don't have novel class labels in the training set), dealing with such problem may
223 require additional novel class dataset to be included for training, or use other methods like zero-shot
224 learning. Nonetheless, we believe our experiments provides a solid starting point for future works.

225 **7 Conclusion**

226 In this paper, we experimented with various ways to achieve the best possible performance for the
227 particular image classification competition task. Specifically, we applied the SGNet architecture
228 to improve the subclass performance, and we experimented with super-resolution method, i.e.,
229 the EDSR model to enhance overall model performance. Although our methods can improve the
230 overall performance of the model on both local evaluation set and online test set, we were unable
231 to fully overcome the data imbalance issue on subclass labels, especially for the imbalanced novel
232 subclass distribution in online test set that is not presented in training set. Based on the results
233 from local evaluation set and online test set, our conjecture that super-resolution and super-class
234 guided architecture techniques can improve the model generalizability and overall performance is
235 demonstrated.

236 **References**

- 237 [1] B. Lim, S. Son, H. Kim, S. Nah, K. Lee. (2017). Enhanced Deep Residual Networks for Single Image
238 Super-Resolution. *CVPR 2017*. <https://arxiv.org/abs/1707.02921>
- 239 [2] Li, K., Wang, N. Y., Yang, Y., & Wang, G. (2021). SGNet: A super-class guided network
240 for Image Classification and Object Detection. *2021 18th Conference on Robots and Vision (CRV)*.
241 <https://doi.org/10.1109/crv52889.2021.00025>
- 242 [3] Muller, S. G., & Hutter, F. (2021). Trivialaugment: Tuning-free yet state-of-the-art data augmentation. *2021*
243 *IEEE/CVF International Conference on Computer Vision (ICCV)*. <https://doi.org/10.1109/iccv48922.2021.00081>