METASwin: A Meta Information-Aware Network For Fine-Grained Visual Classification

Siqi Xiao[1], Jiajing Zhu[1], Zhuomin Zhou[1], Yu Jiang[1]

Abstract

Fine-Grained Visual Classification (FGVC) is a very important research area in the field of computer vision. FGVC focuses on the small differences in the same super-category to distinguish between multiple subordinate categories. Recent state-of-the-art methods contain different kinds of transformers and have achieved a relatively high accuracy. However, only analyzing the visual information is not sufficient if we want a higher accuracy. Actually nowadays, besides visual information, images are always equipped with some meta-information, including the longitude, latitude and time when shooting a picture or some simple descriptions that someone else added. Thus, after processing this information in a particular way, this kind of meta-information can help us increase the accuracy or efficiency of the task of classification. Here we choose Swin Transformer and add processed meta-information to it. Our model is equipped with a plug-in module from [3] which can output pixel-level feature maps and fuse filtered features so that the performance of Swin Transformer can be improved. Then, we add meta-information to it and try to further improve the performance of this model. Adding meta information, we found that the accuracy of classification can be improved by about 1.7% compared with just training with visual information.

Keywords: FGVC, Swin Transformer, Meta Information

1. Introduction

Vision classification are ubiquitous in our lives as it can be applied to almost all parts of the life, such as face recognition and object classification. Vision classification tasks can be divided into two categories by level of details: coarse-grain visual classification and fine-grained visual classification (FGVC). Unlike traditional one, FGVC needs to classify two similar species with few differences like the shape of the ears and the color of the fur. Because of the little inter-class variations, big intra-class variation and the high expense for experts annotating, it is considered to be a challenging job.

There are some current methods to deal with FGVC. One way is to take in additional bounding-box and use supervised learning[18]. However this method need a large amount of human resource for the marking jobs. Another potential risk of this method is the false attention marked by specialists, which may lead to lower accuracy and higher cost. Other ways also exists, including exploiting reinforced-learning, multi-attention, and etc[19][14]. However, these models are of high complexity and inquire high computing competence, which indicates the impossibility to get a compelling result in a relative short time or lower cost.

Comparing to coarse-grain image classification tasks, FGVC emphasizes more on the fine distinctions under the same class or species. If it is hard to only learn from vision, why not combine various meta information along with the image? Here comes our motivation. Geography location and text information has been proved effective [4], which inspires us to utilize them by combination.

We combine the meta information from the image with the vision information as the concatenate input of our designed vision transformer to improve the classification task. Meta information includes the date, geography information and text description. Vision Transformer is a

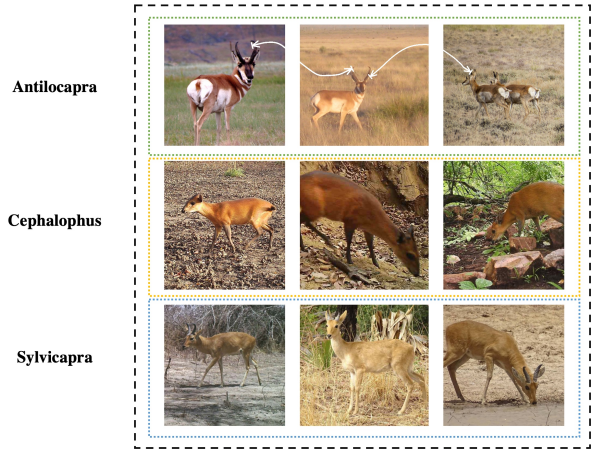

Figure 1: Objects in a row are the same species, while objects in different rows are not the same species though have similar appearance. For human, in order to tell what species they are we need to extract distinct features of each species, and the same to models, for example, the horn/ear of these animals.

method to segment the original image and flatten it into a sequence, which is efficient and less image-specific.

FGVC has been a difficult job not only for machine, but also for experts. With our work, we can save a lot manual work to annotate the images. Also, machine can discover the subtle features which is hard for human eyes, and thus human fault can be reduced. Furthermore, with the help of text information as meta information, the model size and complexity can be reduced.

2. Related Works

In this section, we will briefly introduce xxx and some existing methods for fine-grained visual classification, including different kinds of transformers used.

2.1. Fine-Grained Visual Classification

Fine-Grained Visual Classification (FGVC) mainly focuses on the small differences in the same super-category to distinguish between multiple subordinate categories. Recent state-of-the-art methods contain different kinds of transformers and have achieved a relatively high accuracy. Firstly, there are some CNN-based methods focusing on extracting effective information from multi-level features [12] [10], or locating some distinguishing feature points [16] [9]. Also, there are some kinds of transformer-based methods focusing on different multi-level Transformer layers:

Transformer For Vision Transformer [8], it splits the whole picture into fixed-size patches, and then linearly embedded each of them as well as position embeddings, which later served as the role of sequence in NLP problems. For MetaFormer [17], it states that the inside token mixer module, which in previous researches is the attention-based module in order to achieve a competitive result, can be replaced by a simply module such as pooling. For Swin Transformer [11], it proposes a self-attention model with shift windows, called Shifted Window based Self-Attention (SW-MSA), which can largely reduce the computational cost, and thus getting the training cheap in terms of computation.

In our research, we draw inspiration from these different kinds of transformer and created our own transformer called METASwin, which can deal with meta information and has a relatively high accuracy.

2.2. Self-supervised Learning

Self-supervise learning is proved to be a good alternative to supervised learning in some cases.[] In our case, we consider it a rational choice for our pretrain method since labeled data for Fine-Grained Visual Classification tasks are scarce and it takes the advantage of eliminating manual annotation and low data threshold.

Contrastive learning This is one method of SSL, which achieves state-of-the-art currently [2][6][13][1], performs well in learning features and is proved to have the ability to make models extract latent representations. It works based on discriminating different augmentations of images. By intuition, we need different augmented views of the same image to be similar, while in other cases, distances between augmented views are long. Recent SSL works show the usefulness of contrastive learning through augmentation[15], momentum encoders [2][7], contrastive losses[1] and so on. We feel interested in momentum encoder and exploit MoCo-like method to do our pretrain, hoping our model learn a good embedding.

3. Method

In this paper, we aim at finding the region with high discrimination and combines these features with meta information to better classify the images. The overall framework is shown below in Fig 2.

3.1. Problem Formulation

Here, we consider the problem of fine-grained visual object recognition, assuming a given dataset with n samples, *e.g.*, $\mathcal{D} = \{(x_1, m_1, y_1), \ldots, (x_n, m_n, y_n)\}$, where $x_i \in \mathbb{R}^{H\times W\times 3}$ refers to an image, m_i denotes the meta information, $y_i \in \mathbb{R}^C$ refers to the semantic class that the object belongs to. Specifically, we aim to train a model that jointly considers the visual information and meta information for fine-grained visual representation learning:

$$y_i = \Phi_{\text{FGVC}}(\Phi_{\text{ENC}}(x_i), \Phi_{\text{M}}(m_i)) \tag{1}$$

3.2. Visual Encoder

Contrastive learning pretrained model: Currently, many works on fine-grained vision classification task rely on the supervised pretrained models on ImageNet. However, the performance of application on the classification on the fine-grained targets is less satisfying since ImageNet mainly focus on the coarse-grained image classification. Therefore, we adopt a contrastive learning model to pretrain the parameter. As Chen suggested[2], the performance of the pretrained model gets better accuracy on the classification tasks. The general structure of the pretrained model is shown in the fig.(?) We maintained a query q as the objects that need to be detected and updated a queue k where the items in the queue are randomly cropped, flipped, clustered and colored. To save the memory and enhance efficiency, we adopt a momentum update for the keys in k. A positive batch was gained when q and k get the same image with different preprocess, otherwise, a negative batch is gained. In order to support the positive batch and punish the negative batch, the loss function was calculated as $\mathcal{L} = -\log \frac{\exp(q\cdot k_+/\tau)}{\sum_{i=0}^{K}\exp(q\cdot k_i/\tau)}$, where k_+ denotes a positive batch and k_i denotes a negative batch.

Selector: We choose swin transformer to be our backbone. The hierarchical structure of swin transformer is similar to feature pyramid network, which can extract visual features at different levels. According to the work by Chou at el. [3], we use a weakly supervised selector to filter the useful feature. $f_i \in R^{C\times H\times W}$ refers to the feature map output of the i^{th} swin block, where C refers to the size of the feature dimension, H and W refers to the height and width of the feature map. The feature map f is fed into a weakly supervised selector, where we will select first few feature points with high confidence score. Each feature point will be fed into a linear classifier. If the highest probability of the predicted result y_l is bigger than a set value, the feature point is considered to be a useful one. It will be remained for further combination of features. In contrast, the predicted result is smaller than the set value, then the feature point will be ignored.

The implementation of selector is from Chou at el [3]. This selector takes the advantage of the pyramid structure of swin transformer to learn the features in more dimensions, not only the height and width, but also the stages.

$$y_l = \text{Softmax}(\text{MLP}(x))$$

3.3. Meta information

It is not easy to classify some fine-grained species even for the experts because of high similarity. Some extra information are needed to help make the final conclusion of the classification. Fig.**??** clarify an example style of meta-information we can use. Here we mainly focus on the date time and geological information. Since different species have different habits and different places to live, some geographical information may be helpful when conducting the fine-grained tasks. We first normalized all the latitude and longitude according to $[lat', lon'] = [\frac{lat+90}{180}, \frac{lon+180}{360}]$. Then, we convert the geographic coordinate to a rectangular coordinate system [5], *i.e.*, $[lat', lon'] \rightarrow [x, y, z]$. Similarly, to ensure the continuity of the date, we perform $\left[sin\left(\frac{2\pi\times month}{12}\right), cos\left(\frac{2\pi\times month}{12}\right)\right]$ as the month information. by training a text encoder? or something ? use the same formulation as in Sect. 3.1.

3.4. Fusion

In this stage, we concatenate the meta information and the vision information as the input of the combiner.

We develop a layer to preprocess the meta tokens in order to fit with the vision tokens. As shown in Eq.2, it is a linear layer with Layernorm, where m_i is the meta token sequence, LN represents the operation of Layernorm.

$$m_{i+1} = \text{MLP}(\text{LN}(m_i)) \tag{2}$$

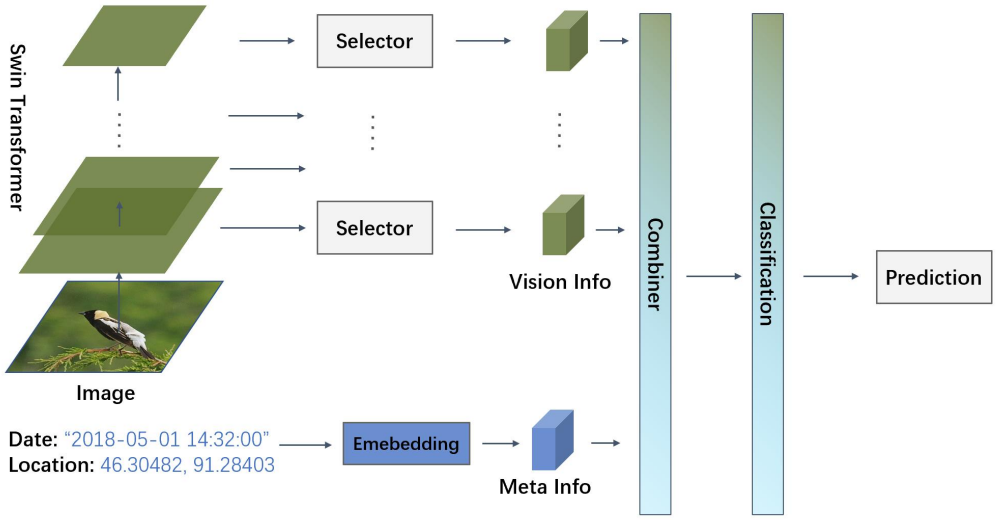

Figure 2: The overall structure of our project

Now, we can concatenate the tokens. In Eq.3, x_{meta} refers to the meta token, x_{vision} refers to the visual token, and $\mathbf{z}_0$ is the token sequence.

$$\mathbf{z}_0 = \left[\mathbf{x}^1_{\text{meta}}, \cdots, \mathbf{x}^{n-1}_{meta}; \mathbf{x}^1_{\text{vision}}, \cdots, \mathbf{x}^m_{\text{vision}}\right] \quad (3)$$

Our model supports **five** different types of combiners.

1. **Full-Connected** combiner: This is the basic one which is only a fully connected layer.
2. **MLP** combiner. This structure is implemented by a simple linear combiner.
3. **Convolutional** combiner: We aggregate the features by convolution through the pooling. $z_{i+1} = \text{Maxpool}(Relu(Conv(z_i)))$
4. **GCN** combiner: We use graph convolution in this combiner, which regards all the feature points as a graph structure. We use the graph to aggregate the features. This method also take the locality of different stages into consideration. GCN is much simpler than transformers, which takes less time to compute. This can improve the efficiency of the model, with little loss of accuracy.
5. **LSTM** combiner: Considering the efficiency of Vision Transformer and to take the advantage of the sequence input, we also tried LSTM model.

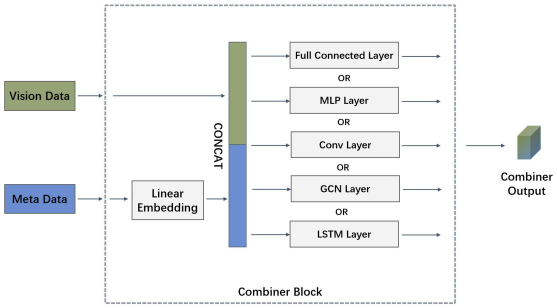

Figure 3: The structure of combiner.

3.5. Loss Function

To calculate the loss, we first take the average of the prediction of all features. The class loss of the entire block is calculated through Cross Entropy, which we denotes as L_b. We use a mask to denote the selected and dropped region, where *1* represents the selected feature points and *0* represents the dropped feature points. We sum the features of the selected and dropped regions and take Cross Entropy of the sum respectively as the loss within each block. We denote the loss of the selected region as L_s and the loss of the dropped regions as L_n. The combiner

category loss is calculated through Cross Entropy, which is represented as L_c. Therefore, the total loss function is defined as Eq.4 as shown below.

$$L = \lambda_b L_b + \lambda_s L_s + \lambda_n L_n + \lambda_c L_c \quad (4)$$

where the λ_b, λ_s, λ_n and λ_c are the weights of L_b, L_s, L_n and L_c. We set $Lb = 1$, $Ls = 0$, $Ln = 5$, $Ls = 1$.

4. Experiments

In this section, we will introduce the datasets, the evaluation method, and the setting of experimental hyperparameters. Then we will discuss our experimental results and compare it with some current state-of-the-art methods when they are solving similar problems. Finally, we will discuss some factors considered critically that may affect the classification accuracy and our method's performance.

4.1. Data

Mammalia-iNaturalist2018 The datasets we use are revised based on iNaturalist2018, a big dataset including 8142 species of plants and animals. All 8142 species are categorized into 14 supercategories, supported by 437513 train images and 24426 validation images in total. Consider both the size of the dataset and the problem to solve, we revised the dataset, curtailing the size and limiting the supercategory. Here we chose Mammalia (indicating this dataset as Mammalia-iNaturalist2018 in the following passsage) from all 14 supercategories, since we found that some of the supercategories are classified by different kingdoms and some are classified by class. By the definition of FGVC, choosing supercategories classified based on class are much more reasonable, since animals/plants in one class are much more similar in appearance and features. Mammalia-iNaturalist2018 is formed by 234 categories, 20104 train images and 702 validation images.

Plantae-iNaturalist2018 We also used other data to verify our hypothesis of our method on similar classification problems. To be more specific, we want to find out the power of the method on classifying objects that are not fine-grained. Based on the upper discussion, the different levels of classifying the supercategories exist in iNaturalist2018 and we found that Plantae is one of the supercategories categorized based on kingdom. Obviously, objects in one kingdom is more coarsely grained than in one class. We further modified the Plantae data by choosing 156 species under 156 genus to guarantee the coarseness of classification. This dataset, part of the Plantae supercategories in iNaturalist2018 (referred as Plantae-iNaturalist2018), consists of 28615 train images and 468 validation images.

We adopt Swin-T as our backbone network, and the input image is a 384×384 color image. We also adopt meta-information in iNaturalist2018 as our auxiliary train information. Each image has its own record of categorization, location, and date photographed. We exploit these information and map them one-to-one with the images so that it can be helpful in increasing classification accuracy.

All in all, datasets we used have their inputs as images of similar plants/mammalia, and their output as numbers where each number represents a species. What we need to do is to conceives images and tell what species the objects in these images belong to.

4.2. Evaluation method

Accuracy is our most important evaluation metric. It is the percentage of predictions our model got right. The accuracy can be calculated by $Accuracy = TP + TN/TP + TN + FP + FN$, where TP means true positive, TN means true negative, FP means false positive and FN means false negative. Another possible evaluation metric can be the time used to get the result of certain datasets if we can get the related data. We are expecting a higher accuracy or a shorter time compared to other methods.

4.3. Experimental details

The method of data augmentation is the same as Swin-T. First, scale the image to 510×510, and then, by performing Random Crop, Random HorizontalFlip, Random GaussianBlur, RandomAdjustSharpness, and normalization during training. The kernel size of the previous GaussianBlur is 5×5, and the sharpness_factor of previous RandomAdjustSharpness is 1.5. In validation procedure, the data is augmented by Resize, CenterCrop and normalization. We use SGD as o optimizer, with momentum equals

to 0.9, learning rate equals to 0.0005. Besides, the cosine decay is also used and we set the weight decay to 0.0005. We also set batch size to 12 and epoch number to 50.

It takes about 15 hours to complete a training on either Mammalia-iNaturalist2018 or Plantae-iNaturalist2018.

4.4. Results

4.4.1. Compare with baseline and SotA approaches

Since our baseline is Swin-T model, We designed experiments to test the performance of Swin-T on this classification task. Swin-T model includes sliding Windows and has a hierarchical design. The sliding window operation includes non-overlapping local window and overlapping cross-window. Limiting attention computation to a window can introduce locality of CNN convolution operation on the one hand and save computation on the other hand.

Based on the ablation experiments, we choose the best parameters in each experiment and fixed these parts to compare the performance of the baseline and our approach. The number of selected area of four layer in FPN module is [256,128,64,32] respectively, and the dimension of the meta-information is 14.

The comparison between our approach and baseline is shown in Tab.1. Based on the growth of accuracy, we are convinced that the meta-information is helpful when conducting FGVC-classification task.

We also compare our result with state-of-the-art in similar task[5]. We apply Sotas both with and without meta-information on our Mammalia-iNaturalist2018. The comparisons are shown in Tab.1. As our approach with meta-information and without meta-information both outperform the corresponding counterpart, it is convincing that: 1. Our backbone is suitable on this FGVC classification task. 2. Swin-T works better with meta-information than the combination of meta-information with ViT. So we may conclude that our approach performs well on FGVC task.

4.4.2. Classification Result of Plantae-iNaturalist2018

We apply our method on Plantae-iNaturalist to test the power of our approach on dealing with medium-grained classification problem, which is to classify objects that are of one kingdom, but not so confusing as objects in FGVC. The best accuracy of our approach on Plant-iNaturalist is 97.436% with meta-information and 96.211% without meta-information, a satisfactory result, since though objects are not fine-grained, they are still similar in some sense. The comparison with classification accuracy on Mammalia-iNaturalist2018 is shown in Fig.4. We observe that the performance without met-information on Plantae-iNaturalist is already very well and the improvement of adding meta-information is less significant than that of Mammalia-iNaturalist. Therefore,this offers an insight that our additional meta-information may be much useful when dealing with FGVC tasks, since it's harder to perform well on than on medium-grained images, not to say regular classification tasks.

Table 1: Comparison among performance of our approach (METASwin) and Sotas on Mammalia-iNaturalist2018

Method	Backbone	Input	Accuracy(%)
Metaformer[5]	ViT	image	79.981
Metaformer[5]	ViT	image+text	82.074
PIM[3]	Swin-T	image	81.624
METASwin	Swin-T	image+text	83.333

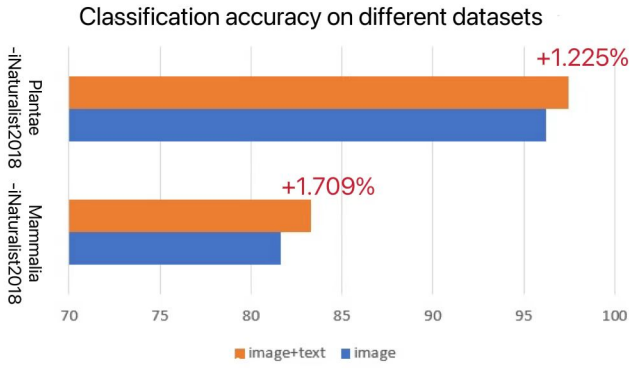

Figure 4: Comparison of performance between METASwin on different datasets, we may find our approach performance good on both cases, but excellent in medium-grained case. It's good performance indicates its generality and great power on classification, even on hard task as FGVC. We may also find that meta-information are more important in FGVC case because of the task's difficulty.

4.4.3. Ablation study

1) Meta-information dimension In this part of the experiment, we designed a number of different ways to deal with the meta information. In the dataset we use, the pictures are equipped with meta information including the longitude, latitude and the time, especially the hour the pictures were taken. We use methods of trigonometric function and one-hot encoding to deal with the month. For longitude and latitude, we also use trigonometric function and normalization to deal with it. Using different combinations, we found that the best accuracy is got when we deal with the month using one-hot encoding and deal with the longitude and latitude using spherical coordinates. When other conditions are the basic ones and are totally the same, the accuracy of the above experiment can be 81.446%. In comparison, if we use both trigonometric function, the accuracy can be 79.345%; if we use one-hot encoding for months and normalization for longitude and latitude, the accuracy can be 81.071%; if we use trigonometric function for months and normalization for longitude and latitude, the accuracy can be 80.627%. Thus, we may come up with the conclusion that dealing with months using one-hot encoding and longitude and latitude using spherical coordinates is a better way to deal with the meta information.

2) Number of Selections This part of experiment is designed to test performance of different combination of selection numbers. The list in the selections number represents the number of selected areas of the four blocks of Swin-T (the first number in the list maps to the first block in Swin-T and so on). We choose 6 combinations, shown in Tab.2, where the right list of the table records the best accuracy of each combination.

Based on the experimental results, we observed that the combination of [256,128,64,32] is the one with the best performance. However, these accuracy are not diverging much, which indicates the choices of combination of selection numbers may not be the most critical part when pursuing better classification performance on FGVC task using this approach.

3) With/Without Meta-information This part of experiment aims to find out the usefulness of exploiting meta-information. Based on the comparison between Swin-T and METASwin in Tab.1, we find the latter outperform the former by increasing the accuracy by 1.709%. With this result, we may conclude meta-information is a good helper when doing FGVC task, since they may offer auxiliary information about specific species and some rules are nested in these meta-information.

Table 2: Compare different combination of Selection numbers

Selections Number	Accuracy(%)
[32,32,32,32]	82.051
[256,128,64,32]	83.333
[512,256,128,64]	81.766
[1024,512,128,64]	82.479
[1024,512,128,128]	82.336

Table 3: Comparison between different combiner structures

Combiner Type	Number of Layers	Accuracy(%)
FC	/	81.624
MLP-Linear	1	81.197
	2	82.613
GCN	1	81.479
	2	81.921
MLP-CONV	1	81.325
	2	81.932
LSTM	1	81.909
	2	82.051

4) Different combiners and number of layers This part of experiment is designed to test performance of different structure of combiners and different layers on them. We designed five different combiners with two layers. The results are shown in the Tab.3 Based on the experimental results, we observed that the two layer of MLP achieves best performance.

5) Different backbones This part of experiment aims to find out the best backbone on this task. The result is shown in Tab.4 We fixed other parameters and experimented with 3 different backbones, which are swin_large_patch4_window12_384_in22k, swin_tiny_patch4_window12_384_in22k, and

Table 4: Comparison between backbones

Backbone	Accuracy(%)
swin_large_patch4_window12_224_in22k	82.237
swin_tiny_patch4_window12_384_in22k	80.439
swin_large_patch4_window12_384_in22k	83.333

swin_large_patch4_window12_224_in22k (referred as large384,tiny384 and large224 in following passage).

By doing comparison between the performance based on different backbones, we find the one with large384 is the best when pursuing higher accuracy, because it extracts more features and has larger size. However, tiny384 has its own advantage, which is the higher time-efficiency. It can run much faster than larger backbone, while just compromise small percentage of accuracy. Training with tiny backbone only cost about 10 hours. In this case, tiny384 is also a good choice when in time-limited case. Since our evaluation method is about accuracy, we finally choose large384 as our backbone.

5. Conclusions and Future Work

In this work, we complete the FGVC task with the Swin Transformer and add meta-information on it. METASwin uses a plug-in-module to generate pixel-level feature map and fuse the filtered features. Meanwhile, METASwin also provides a MetaLayer to preprocess the meta-information to fit the visions. We designed five approaches to process the meta-information as combiners. METASwin achieve better performance in iNaturalist dataset. We believe that meta-information is essential for fine-grained recognition tasks in the future and METASwin can provide a way to deal with the auxiliary information.

Acknowledgments. This submission is the project proposal for course ECE4880J in 2022SU. This course is held by UM-SJTU Joint Institute, Shanghai Jiao Tong University. The mentor of this course is Dr. Siheng Chen.

References

[1] Chen, T., Kornblith, S., Norouzi, M., Hinton, G., 2020a. A simple framework for contrastive learning of visual representations, in: International conference on machine learning, PMLR. pp. 1597–1607.

[2] Chen, X., Fan, H., Girshick, R., He, K., 2020b. Improved baselines with momentum contrastive learning. arXiv preprint arXiv:2003.04297 .

[3] Chou, P.Y., Lin, C.H., Kao, W.C., 2022. A novel plug-in module for fine-grained visual classification. URL: https://github.com/chou141253/fgvc-pim, arXiv:2202.03822.

[4] Chu, G., Potetz, B., Wang, W., Howard, A., Song, Y., Brucher, F., Leung, T., Adam, H., 2019. Geo-aware networks for fine-grained recognition, in: 2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW), pp. 247–254. doi:10.1109/ICCVW.2019.00033.

[5] Diao, Q., Jiang, Y., Wen, B., Sun, J., Yuan, Z., 2022. Metaformer: A unified meta framework for fine-grained recognition.

[6] Grill, J.B., Strub, F., Altché, F., Tallec, C., Richemond, P., Buchatskaya, E., Doersch, C., Avila Pires, B., Guo, Z., Gheshlaghi Azar, M., et al., 2020. Bootstrap your own latent-a new approach to self-supervised learning. Advances in neural information processing systems 33, 21271–21284.

[7] He, K., Fan, H., Wu, Y., Xie, S., Girshick, R., 2020. Momentum contrast for unsupervised visual representation learning, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 9729–9738.

[8] Heigold, G., Dehghani, M., Houlsby, N., Gelly, S., Weissenborn, D., Zhai, X., Beyer, L., Kolesnikov, A., Uszkoreit, J., Dosovitskiy, A., Minderer, M., Unterthiner, T., 2020. An image is worth 16x16 words: Transformers for image recognition at scale. Learning .

[9] Heliang Zheng, Jianlong Fu, Z.J.Z., Luo., J., 2019. Looking for the devil in the details: Learning trilinear attention sampling network for fine-grained image recognition., in: CVPR.

[10] Lianbo Zhang, Shaoli Huang, W.L., Tao., D., 2019. Learning a mixture of granularity-specific experts for fine-grained categorization., in: ICCV.

[11] Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B., 2021. Swin transformer: Hierarchical vision transformer using shifted windows, in: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), pp. 10012–10022.

[12] Ruoyi Du, Dongliang Chang, A.K.B.J.X.Z.M.Y.Z.S., Guo., J., 2020. Fine-grained visual classification via progressive multi-granularity training of jigsaw patches., in: ECCV.

[13] Tian, Y., Krishnan, D., Isola, P., 2020. Contrastive multiview coding, in: European conference on computer vision, Springer. pp. 776–794.

[14] Wang, F., Jiang, M., Qian, C., Yang, S., Li, C., Zhang, H., Wang, X., Tang, X., 2017. Residual attention network for image classification, in: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 3156–3164.

[15] Wang, X., Qi, G.J., 2021. Contrastive learning with stronger augmentations. arXiv preprint arXiv:2104.07713 .

[16] Yao Ding, Yanzhao Zhou, Y.Z.Q.Y., Jiao., J., 2019. Selective sparse sampling for fine-grained image recognition., in: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), pp. 6599–6608.

[17] Yu, W., Luo, M., Zhou, P., Si, C., Zhou, Y., Wang, X., Feng, J., Yan, S., 2021. Metaformer is actually what you need for vision.

[18] Zhang, N., Donahue, J., Girshick, R., Darrell, T., 2014. Part-based r-cnns for fine-grained category detection, in: European conference on computer vision, Springer. pp. 834–849.

[19] Zheng, H., Fu, J., Mei, T., Luo, J., 2017. Learning multi-attention convolutional neural network for fine-grained image recognition, in: Proceedings of the IEEE international conference on computer vision, pp. 5209–5217.