Skin40 Classification Using Few-Shot Learning and Data Augmentation strategy

Zhicheng Wang
School of Computer Science and Engineering
Sun Yat-sen University
Guangzhou, China
wangzhch23@mail2.sysu.edu.cn

***Abstract*—This report is the final assignment for the Artificial Neural Networks course at Sun Yat-sen University. We built a neural network framework to solve the classification problem of Skin40 dataset. We try various neural network architectures, including CNN-based and Transformer-based networks, including AlexNet, VGG, ResNet, DenseNet, and RepLKNet, Transformer-based neural networks include Swin transformer, ViT, and for Swin transformer we test four different sizes of network data. In addition, because of the small amount of data in this dataset, we try various data augmentation methods and improve the accuracy of the experiments by adjusting the parameters. We try the transfer learning technique, first using the Dermnet and HAM10000 datasets on Kaggle for training and then fine-tuning it with the Skin40 dataset. We try data distillation to train the transformer model, and improve the accuracy of the vision-transformer model. Our model finally achieves 83.625% accuracy in the Skin40 data.**

I. Introduction

Convolutional neural networks have traditionally dominated computer vision modeling (CNNs). Starting with AlexNet and its revolutionary performance on the ImageNet image classification challenge, CNN architectures have become increasingly potent through larger scale, more extensive connections, and more complex forms of convolution. The performance of CNNs, which serve as the backbone networks for a variety of vision tasks, has vastly improved as a result of these architectural advancements, which have benefited the entire field. On the other hand, the evolution of network architectures in natural language processing (NLP) has taken a different path, with the Transformer being the prevalent architecture today. Transformer is notable for its use of attention to model long-range data dependencies; it was designed for sequence modeling and transduction tasks. Its enormous success in the language domain has prompted researchers to investigate its adaptation to computer vision, where it has recently demonstrated promising results on specific tasks, including image classification and joint vision-language modeling.

In Task 1, we were provided with the Skin40 dataset, a dermatological classification problem consisting of 40 categories corresponding to 40 dermatological conditions. There are sixty images for each skin disease, ranging from 1080p to 240p in resolution. Due to the limited amount of data, the total number of images, 2400, is categorized as a few-shot problem. As required, we attempt to construct the model using Pytorch and determine its precision using five-fold cross-validation.

We initially attempted to experiment with CNN architecture models including AlexNet, VGG, ResNet, and DenseNet. DenseNet achieved the highest accuracy in our tests, but additional experiments, including not using the pretrained model and only training the classifier layer, revealed that these two approaches did not produce superior results. We also reproduced a more recent convolutional model, RepLKNet, from CVPR 2022, which employs a large convolutional kernel and achieves the same accuracy as Swin transformer on ImageNet, but the experimental outcomes were unsatisfactory in our tests.

We also attempted to use a new transformer-based model, transformer as a widely used model in NLP tasks, also recently gained good results in CV field, many models based on the original ViT were proposed, gradually increased the accuracy, and became a popular topic of research in recent years. We also attempted to build experiments using Swin transformers of various sizes, including Swin-Tiny, Swin-Based, Swin-Small, and Swin-Large models for testing, but the initial results were unsatisfactory due to the large size of the model and the lack of data.

We tried a variety of data augmentation techniques, including horizontal and vertical inversion, image rotation, color transformation, random cropping, and CutMix to increase the volume of data. These methods were partially validated, and we combined valid data augmentation methods to train our final model.

We tried the transfer learning technique, where we pre-trained the model on some skin-related datasets and fine-tuned it on the Skin40 dataset. However, due to the size and content of the dataset, we did not get good training results in the end.

In addition, we employ knowledge distillation, utilize ResNet50 as a teacher network, and extract knowledge to ViT to improve the ViT-Tiny model's accuracy on Skin40.

Our main contributions are as follows:

- We test the performance of various models and compare the difference in accuracy with limited data augmentation.
- We use different means of data augmentation, and compare the impact of each method on accuracy.
- We use transfer learning, which introduces similar datasets, although the results of this experiment are not ideal.

- We use knowledge distillation to improve the ViT-Tiny's accuracy.

II. RELATED WORK

A. Alex And VGG

LeCun developed LetNet5 [1], a model for handwritten number identification that shows the CNN model. This paradigm was not widely adopted, however, due to the limitations imposed by the computers of the period. Later, Alex improved LeNet5 in three ways: 1) adopting ReLU as the CNN activation function; 2) boosting dropout model parameters randomly to improve the model's generalizability; and 3) overlapping with pooling maximum. On imageNet2012, AlexNet [2] ranked first with a 15.3 percent error rate, outperforming the runner-up by 10 percent. Karen Simonyan and Andrew Ziserman replaced the VGG model with a series of 3x3 convolutional kernels to replace AlexNet's large convolutional kernel, therefore increasing the network's depth and, to a lesser degree, its performance while maintaining the same perceptual field. VGG [3] placed second in the 2014 ILSVRC tournament.

B. ResNet

The CNN model has changed through time, beginning with AlexNet. The fitting ability of the model should improve as the depth of the network model grows, and the recognition ability of the picture should improve as well. Although batch normalization effectively eliminates gradient disappearance and gradient explosion, the model's capabilities have not greatly enhanced. Kaiming He argues that the model's optimization is the reason for its poor performance, and he offers a residual structure that permits the gradient to be handed forward through certain convolutional blocks. Kaiming He residual technique adds depth to the network model and makes it easier to train. ResNet [4], ResNeXt [5], SENet [6], and ResNet have all appeared on base of ResNet, and they all have achieved good performance.

C. RepLKNet

The convolutional kernel size of a CNN has traditionally been intended to be 3x3 or 5x5, and improved performance is predicted by modifying the model's parameters such as depth, width, groups and so on. This recent study on CVPR 2022 takes a different approach make out the RepLKNet [7], demonstrating that just increasing the size of the convolutional kernel of a CNN may provide performance comparable to Transformer while being efficient. On bigger bulk tasks, it's comparable to Transformer. The model has a maximum accuracy of 87.8% for the super-largest model with additional data training, which is equivalent to Swin-Base on ImageNet. It outperforms the classic ResNetXt-101 model in COCO object detection by a factor of ten. comparable to Swin, reaching 55.5% mAP. In the semantic segmentation of Cityscapes, only use the RepLKNet-Base of ImageNet-1K pretrain. Even more than the Swin-Large of ImageNet-22K pretrain. This is a transcendence across model magnitudes and data magnitudes.

D. Vision Transformer

The Transformer architecture's applications to computer vision are still limited, despite the fact that it has emerged as the de facto standard for natural language processing tasks. Attention is either used in conjunction with convolutional networks in vision or is used to replace some of the convolutional network's constituent parts while maintaining the overall structure of the network. We demonstrate that there is no need for this dependency on CNNs and that pure transformers used directly on sequences of picture patches can get excellent results on image classification tasks. Vision Transformer (ViT) [8] achieves excellent results compared to state-of-the-art convolutional networks while using significantly less computational resources to train when pre-trained on large amounts of data and applied to multiple mid-sized or small image recognition benchmarks (ImageNet, CIFAR-100, VTAB, etc.).

E. Swin Transformer

Transformer was proposed by Vaswani et al [9] and was originally a model for machine translation. The NLP field has been considered the most advanced method for a long time, and the Transformer model has been applied to the field of computer vision in recent years. MingHang Zheng and Peng Gao designed the DERT [10] model for object detection tasks. Vision Transformer is the first model that applies the model of the NLP domain directly to computer vision. Its achieves impressive speed and accuracy trade-off image classification compares to convolutional networks. However, Vision Transformer [8] have corresponding drawbacks, such as it requires a lot of data to train, the memory delay and cost are very expensive, and the model's number of parameters is large and the computational complexity is very complicate. For the first question, Hugo Touvron and Mattieu Cord [11] proposed some training strategies that allow the model to perform better without an additional dataset. Ze Liu and Yutong Lin proposed a problem where moving windows and hierarchical designs overcome the complexity of model computation and the high memory overhead.

F. Transfer Learning

Due to the high price of human manual labeling and environmental restrictions, sufficient training data belonging to the same feature space or the same distribution as the testing data may not always be available. Thus, it is believed that the existing knowledge gained from previous known objects assists the new learning process through their connections with the new object categories. In transfer learning, both the training data and the testing data can contribute to two types of domains: 1) the target domain and 2) the source domain. The target domain contains the testing instances, which are the task of the categorization system, and the source domain contains training instances, which are under a different distribution with the target domain data. In most cases, there is only one target domain for a transfer learning task, while either single or multiple source domains can exist. In [12], action recognition

is conducted across data sets from different domains, where the KTH data set [13], which has a clean background and limited viewpoint and scale changes, is set as the source data set, and the Microsoft research action data set 1 and the TRECVID surveillance data [14], which are captured from realistic scenarios, are used as the target data set. In [15], the source and target data sets are chosen from different TV program channels for the task of video concept detection.

III. METHOD

A. Image Processing

Fig. 2. Image Cropping

$$acc = \sum_{i} acc_i \tag{1}$$

where acc is the final accuracy of the whole task and acc_i is the accuracy for fold i.

In addition, we normalize the image and remove the watermark from the bottom of the image, which is shown in the figure 2.

For training and validation set, we resize the image to 256 or 296 size image and then centercrop it to get a 224*224 or 384*384 size image. The choice of image size depends on the pretrained models.

B. Data Augmentation

As a base data augmentation, we rotate the image randomly shown in fig. 8, transform it randomly horizontally shown in fig. 4 or vertically shown in fig. 5, transform the color shown in fig. 7, and randomly crop it shown in fig. 8.

Fig. 3. Origin Image

Fig. 4. Horizontal transform

Fig. 5. Vertical transform

Fig. 6. Rotation transform

Fig. 7. Color transform

Fig. 8. Randomly crop

What is more, we try some advanced data augmentation method like CutMix [16] [17]. CutMix is a data augmentation technique that addresses the issue of information loss and inefficiency present in regional dropout strategies. Instead of removing pixels and filling them with black or grey pixels or Gaussian noise, it replaces the removed regions with a patch from another image, while the ground truth labels are mixed proportionally to the number of pixels of combined images. For example x_A and x_B are two different training examples, y_A and y_B are the label of them, CutMix is needed to generate the new training samples $\widetilde{x}$ and the related tag $\widetilde{y}$.

The equation is:

$$\tilde{x} = \mathbf{M} \bigodot x_A + (\mathbf{1} - \mathbf{M}) \bigodot x_B \tag{2}$$

$$\tilde{y} = \lambda y_A + (1 - \lambda) y_B \tag{3}$$

where $\mathbf{M} \in \{0,1\}^{W \times H}$ is a binary mask to drop parts of the area and to fill it. $\bigodot$ is the pixel-by-pixel multiplication, λ is the Beta distribution. In order to sample the binary mask M, the bounding box B of the clipping region is first sampled and used to do the indicated calibration of the clipping region for samples x_A and x_B. The rectangular mask M is sampled in the paper (the length and width are proportional to the sample size).

The sampling equation for the bounding box of the clipping region is as follows.

$$r_x \sim \mathrm{Unif}(0, W), r_w = W\sqrt{1-\lambda} \tag{4}$$

$$r_y \sim \mathrm{Unif}(0, H), r_h = H\sqrt{1-\lambda} \tag{5}$$

The ratio of the clipping area is guaranteed to be:

$$\frac{r_w r_h}{WH} = 1 - \lambda \tag{6}$$

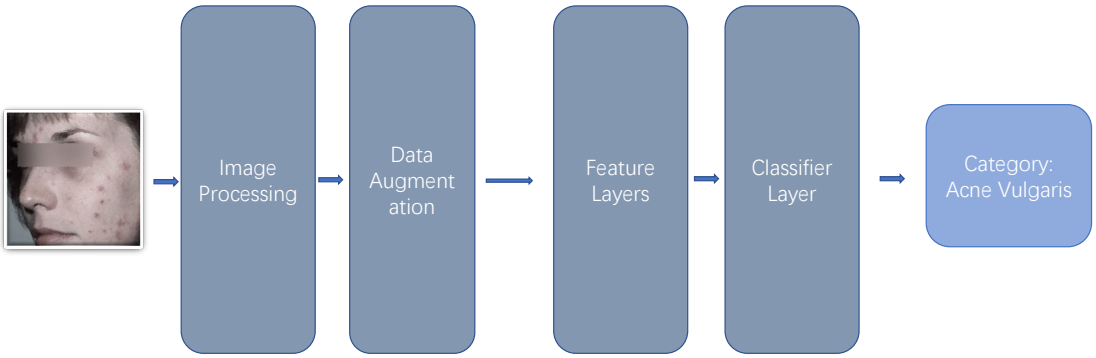

Fig. 1. Basic Model

After determining the cropping region B, set the cropping region B in the mask M to 0 and the other regions to 1. This completes the sampling of the mask, then removes the cropping region B from sample A, crops the cropping region in sample B and fills it with sample A.

C. Classification Models

We use several models which are CNN-based or transformer-based. And we use the pretrained models and adjust the classification layer of these models to 40 categories for the Skin40 dataset.

D. Transfer Learning

We also try to use other datasets for training first, and then use Skin40 for fine-tune. We choose the Dermnet dataset from Kaggle. The data consists of images of 23 types of skin diseases. The total number of images are around 19,500, out of which approximately 15,500 have been split in the training set and the remaining in the test set. And we also test the dataset HAM10000 which is a large collection of multi-source dermatoscopic images of pigmented lesions. We try to trained the pretrained model in these datasets and get fine-tuned on the Skin40 dataset.

E. Knowledge Distillation

In some latest work, some Transformer-based classification model like DeiT are distilled from the traditional CNN-based models like ResNet. The result is that the distilled transformer-based model can have better performance on the test set. We accomplished the experiment that training the ViT model from the ResNet model. Generally speaking, there are two types of distillation, one is soft distillation and the other is hard distillation. The soft distillation is shown in the following equation.

In the right half, Z_s and Z_t are the outputs of student model and teacher model respectively, KL denotes the KL scatter, ψ denotes the softmax function, and λ and τ are the hyperparameters.

$$\mathcal{L}_{\text{global}} = (1-\lambda)\mathcal{L}_{\text{CE}}\left(\psi\left(Z_{\text{s}}\right), y\right)+\lambda\tau^{2}\text{KL}\left(\psi\left(Z_{\text{s}}/\tau\right), \psi\left(Z_{\text{t}}/\tau\right)\right) \tag{7}$$

The hard distillation is shown in the following equation, still in the right half, with CE denoting cross entropy.

$$\mathcal{L}_{\text{hard}} = \frac{1}{2}\mathcal{L}_{\text{CE}}\left(\psi\left(Z_s\right), y\right) + \frac{1}{2}\mathcal{L}_{\text{CE}}\left(\psi\left(Z_s\right), y_{\text{t}}\right) \tag{8}$$

$$y_{\text{t}} = \text{argmax}_c\, Z_{\text{t}}(c) \tag{9}$$

Because of the long training time shown in the original paper, the hard distillation effect is better. So we trained only the DeiT-tiny model and chose the hard distillation approach. The final result was not fully trained due to hardware and time constraints, and our model is trained with four GPUs for one day.

IV. EXPERIMENT

On the Skin40 dataset, we assessed our method using a 5 fold cross-validation procedure. Skin40 is a collection including 60 photos for each of 40 skin disease classifications. Using the 5 fold cross-validation technique, there will be 1920 training photos and 480 assessment images. The resolution of the image in the Skin40 dataset is 1440*1080 and 260*480, and there is a watermark beneath. These photos must undergo cautious preprocessing.

CNN is employed as the model framework in the tests, and the transformer-based models are also evaluated. Top-1 accuracy is used as a performance statistic to assess model performance.

A. Classification Model Test

1) CNN Model: To examine the effectiveness of these state-of-art CNN models on the Skin-40 dataset and utilize this as a baseline for the Transform model later, the following models were tested: AlexNet, VGG-11, ResNet34, ResNet50, ResNet101, and DenseNet-121.

TABLE I
PERFORMANCE FOR CNN MODEL WITH OR WITHOUT DATA AUGMENTATION

Model	Accuracy(Top-1)
AlexNet	55.0%
VGG-11	63.9%
ResNet-34	53.9%
ResNet-50	69.9%
ResNet-101	68.8%
DenseNet-121	72.1%

Simultaneously, it is discovered that the ResNet-34 model's fitting capacity is insufficient for this situation. ResNet-50, which is somewhat more complicated, can raise accuracy by 15% with limited data augmentation. Additionally, the more complicated ResNet-101 lacks Comparatively to ResNet-50, the improvement is the same or there is no evident improvement, showing that the model's complexity is sufficient and other elements require development.

2) Transformer model: ViT-T, ViT-S, ViT-B, ViT-L, DeiT-T, Swin-T, Swin-S, Swin-B, and Swin-L are tested with limited data augmentation methods include Horizontal, Vertical transform and random crop. As demonstrated in the table II below, where DeiT-Tiny is the ViT-Tiny model trained using the knowledge distillation technique.

TABLE II
PERFORMANCE FOR SWIN MODEL WITH OR WITHOUT DATA AUGMENTATION

Model	Accuracy(Top-1)
ViT-Tiny	77.2%
ViT-Small	78.2%
ViT-Base	80.9%
ViT-Large	80.2%
DeiT-Tiny	77.5%
Swin-Tiny	78.6%
Swin-Small	79.0%
Swin-Base	81.1%
Swin-Large	81.2%

It can be seen that the CNN model in this task does not perform as well as the newer transformer-based. Even the earliest ViT model outperforms the DenseNet model, which has the highest accuracy among CNNs. However, the DeiT-Tiny model, which is pre-trained with knowledge distillation using resnet, has a higher accuracy than the ViT-Tiny model, which shows that knowledge distillation allows the model to learn some hidden information to improve accuracy to some extent. Because of the higher accuracy of the transformer model, the subsequent experiments of data augmentation were tested using the Swin models. Also because the Swin-Large training is too time-consuming, we use the Swin-Tiny model for testing in the following section.

B. Data Augmentation

1) Rotation transform: We use the RandomRotation function in Pytorch and set its maximum rotation angle to 15, 30 and 45 to test its enhancement on the Swin-Tiny model.

TABLE III
PERFORMANCE FOR SWIN MODEL WITH RANDOM ROTATION OF DIFFERENT DEGREES

Model	Accuracy(Top-1)
Baseline	78.6%
15 degree	78.5%
30 degree	78.0%
45 degree	77.8%

The results showed that randomRotation did not work well and we did not use it in the final model.

2) Affine transform: We use the RandomAffine function in Pytorch and set its maximum angle to 30 and 60 to test its enhancement on the Swin-Tiny model.

TABLE IV
PERFORMANCE FOR SWIN MODEL WITH RANDOM AFFINE TRANSFORMATION

Model	Accuracy(Top-1)
Baseline	78.6%
30 degree	79.0%
60 degree	78.9%

The results show that RandomAffine is valid, and we finally choose a degree of 30.

3) Color transform: We use the ColorJitter function in Pytorch to test the impact of color transform.

TABLE V
PERFORMANCE FOR SWIN MODEL WITH RANDOM COLOR TRANSFORMATION

Model	Accuracy(Top-1)
Baseline	78.6%
ColorJitter	78.6%

4) CutMix: We use CutMix data augmentation with the same parameter settings as the original paper.

TABLE VI
PERFORMANCE FOR SWIN MODEL WITH CUTMIX

Model	Accuracy(Top-1)
Baseline	78.6%
CutMix	79.9%

The results show that this type of data augmentation is very effective.

C. Transfer Learning

Before training on the Skin40 dataset, other datasets are used to pre-train the model. The feature extraction layer is then fixed, and the classification layer is trained on the Skin40 dataset. The MNIST-CANCER-HUMAN-10000(HAM10000) and Dermnet dataset on Kaggle are introduced here. Each category in these datasets comprises more than one thousand photos, which may be used to train a more accurate feature extraction layer.

TABLE VII
PERFORMANCE FOR SWIN MODEL WITH KNOWLEDGE TRANSFER FROM OTHER DATASET

Model	Accuracy(Top-1)
Baseline	78.6%
Dermnet	70.5%
HAM10000	67.1%

The final results show that the pre-trained models perform poorly on these datasets, and we conjecture that the datasets themselves are not generalized well enough.

D. Final Model Setup

Finally, we utilize the Swin-Large model to obtain an average accuracy of 83.625 % , and the training hyperparameters used to accomplish this average accuracy is given here.

TABLE VIII
THE PARAMETERS USED BY CUTMIX DATA AUGUMENTATION MODULE

Parameters	Set
Cutmix beta	1.0
Cutmix prob	0.5
Cutmix num_mix	2

TABLE IX
OTHER PARAMETERS

Parameters	set
Batch_Size	32
optimizer	AdamW
optimizer lr	1e-4
optimizer weight decay	1e-3
scheduler	StepLR
scheduler step size	1
scheduler gamma	0.88
loss function	CrossEntropyLoss
torch seed	3407

In this model, we use RandomAffine, RandomResizeCrop, RandomHorizontalFlip and CutMix for data augmentation. The final result has an accuracy improvement of about 2.4% compared to baseline.

TABLE X
PERFORMANCE FOR OUR FINAL SWIN-LARGE MODEL

Model	Accuracy(Top-1)
Baseline	81.2%
Final	83.6%

E. Error Analysis

1) Features are blurred: For instance, in the following two images from the sample Blue nevus, the center of one 9 image must be focused on the image's right edge, and the other 10 image's details are too tiny to be recorded adequately by CNN.

2) Classification boundaries are Blurred: The visual characteristics of the Blue nevus sample and the compound nevus sample are extremely comparable. It is difficult to discern these two images based on their visual characteristics, and other classes in Skin40 dataset have similar difficulties.

Fig. 9. features on image's right edge [1]

Fig. 10. features too tiny to be catch [1]

Fig. 11. Blue-nevus sample [1]

Fig. 12. Compound-nevus sample [1]

In the illustration in the fig. 11 is the Blue nevus sample, while the illustration in the fig. 12 is the Compound nevus sample. In the visual representation, their physical characteristics are remarkably similar, making it difficult to distinguish between them.

3) Interference in Feature: Some images do not effectively depict the visual features of disease, and there is an excess of irrelevant interference information.

Fig. 13. Interference left [1]

Fig. 14. Interference Right [1]

For instance, the model may capture image aspects of hands fig. 13 and feet fig. 14 rather than pathogenic abnormalities.

V. CONCLUSION

We test the most advanced CNN and Transformer models and try a number of different data augmentation methods. Finally, using the Swin-Transformer model, a relatively good test set accuracy rate was successfully achieved, and the highest test result in the five-cross-validation was 87% correct and the lowest accuracy rate was 81%. We utilize transfer learning, but the actual effect was not good, and the final

[1]Considering the image viewing experience, we convert it to grayscale.

effect was not ideal, did not improve the performance of the model, and even reduce it. Moreover the method of knowledge distillation is used to improve the correctness of the ViT-Tiny (DeiT-Tiny) model. Our model is open source on the Github https://github.com/predatorq/ann.

REFERENCES

[1] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 1998.

[2] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. *Communications of The ACM*, 2012.

[3] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *computer vision and pattern recognition*, 2014.

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *arXiv: Computer Vision and Pattern Recognition*, 2015.

[5] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. *computer vision and pattern recognition*, 2016.

[6] Jie Hu, Li Shen, Samuel Albanie, Gang Sun, and Enhua Wu. Squeeze-and-excitation networks. *computer vision and pattern recognition*, 2018.

[7] Xiaohan Ding, Xiangyu Zhang, Yizhuang Zhou, Jungong Han, Guiguang Ding, and Jian Sun. Scaling up your kernels to 31x31: Revisiting large kernel design in cnns. 2022.

[8] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *Learning*, 2020.

[9] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *neural information processing systems*, 2017.

[10] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. *european conference on computer vision*, 2020.

[11] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. *arXiv: Computer Vision and Pattern Recognition*, 2020.

[12] Liangliang Cao, Zicheng Liu, and Thomas S. Huang. Cross-dataset action detection. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1998–2005, 2010.

[13] C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions: a local svm approach. In *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, volume 3, pages 32–36 Vol.3, 2004.

[14] Alan F. Smeaton, Paul Over, and Wessel Kraaij. Evaluation campaigns and trecvid. In *Proceedings of the 8th ACM International Workshop on Multimedia Information Retrieval*, MIR '06, page 321–330, New York, NY, USA, 2006. Association for Computing Machinery.

[15] Jun Yang, Rong Yan, and Alexander G. Hauptmann. Cross-domain video concept detection using adaptive svms. In *Proceedings of the 15th ACM International Conference on Multimedia*, MM '07, page 188–197, New York, NY, USA, 2007. Association for Computing Machinery.

[16] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. *CoRR*, abs/1905.04899, 2019.

[17] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *International Conference on Computer Vision (ICCV)*, 2019.

APPENDIX

Training Result Image:

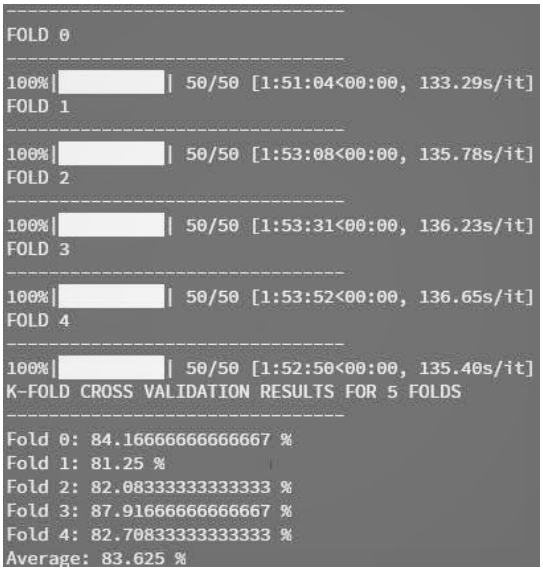

Fig. 15. Training Results