RobustVision: Making CNN and ViT Good Friends with Pre-Trained Vision Model

Haichao Wei
Stanford University
450 Jane Stanford Way. Stanford, CA
haichaow@stanford.edu

Ethan Cheng
ethanlc2@stanford.edu

Chunming Peng
cmpeng@stanford.edu

Abstract

There has been a competition between Convolutional Neural Networks (CNN) and Vision Transformers (ViT) in computer vision field. While CNNs are good at extracting local features due to its inductive bias leading to better generalization, ViT-like models can capture long distance feature interactions and are good at learning global representations. There are existing works to explore ways to fuse them together with hybrid network architectures. However, this approach can make the model too complex, and therefore harder to optimize. In this paper, a simplified hybrid network architecture is proposed, making it easy to leverage existing self-supervised pre-trained CNN and ViT models, and therefore easier to optimize. Our results show that this proposed architecture can reach state-of-the-art results in Image Classification on Image-Net with just 20 epochs, outperforming the original Conformer by 2.4% on ImageNet, and achieving similar results on MSCOCO for object detection while being under- trained. We also show that this approach works across different tasks such as Object Detection and Style Transformation via fine tuning,

1. Introduction

In the past 2 years, inspired by transformer-based pre-trained models in NLP [27], the transformer architecture has been introduced to visual tasks, and self-supervised pre-training for vision representation [2,9] has proved to outperform supervised pre-training [13] in computer vision tasks such as image classification. Transformer-based models, which were used in NLP, have now shown promising results in computer vision benchmarks in fields such as Object Detection [5], Video Classification [37], Image Classification [25] and Image Generation [12]. Thanks to the self-attention mechanism and Multilayer Perceptron (MLP) structure, transformer-based methods show the power of capturing global representation of the images. However, CNN models still have the advantage of extracting local feature representation which decreases the discriminability between background and foreground, and is important for instance segmentation tasks.

There are some recent works combining CNN and Transformer (a.k.a., Conformer) which could potentially lead to a robust vision model leveraging the best of both worlds [26]. This methodology is similar to the Two-Streams Hypothesis in Neuroscience, which claims that the brain has a dorsal and ventral pathway [15] [7] which separately process spatial, location-based information and recognition / identification respectively. The corollaries to current deep learning approaches would be transformer-based models for the recognition / classification tasks, and CNN-based models for spatial locality. It is a good idea to combine them into two branches, while conventional CNNs tend to retain discriminative local regions, the CNN branch of Conformer should activate the full object extent. In contrast to using only the visual transformers, when it is difficult to distinguish the object from the background, esp. for the weak local features, the coupling of local features and global representations should enhance the discriminability of transformer-based features. Indeed, Conformer shows that it can generalize well across different tasks. However, combining both makes the model harder to train: On ImageNet, Conformer needs over 300 epochs to reach state of art results, and due to its design choice of network architecture, it is not easy to initialize it from other pretrained models. Following this path, we propose a variation of conformer that can easily leveraging self-supervised / supervised pre-trained models of both CNN and ViT, and fuse them together in different layers so that it generalize well to Image Classification, Object Detection, Instance Segmentation and Style Transformation.

To evaluate our model, we will test our model against Image classification [10], Object Detection [23], and Style Transformation [22].

The contributions of this paper include:

[0]Github repo: https://github.com/cs231n-finalproject/robustvision

- We introduce a new dual network structure, which retains local features and global representations to the maximum extent from pre-trained model, while still allows interaction between them to learn to fit multiple computer vision tasks.

- We evaluate the the new dual network structure by initializing it with state-of-the-art pre-trained models, show that it is easy to optimize and can reach the state-of-the-art results from CNN and ViT in different tasks, and demonstrate the potential of it being a general backbone network.

2. Related Work

2.1. CNNs with Global Cues

Being regarded as a hierarchical ensemble of local features with different reception fields, most CNNs are good at extracting local features. However, they [24, 30] more or less experience difficulty with capturing global cues. The Conformer approach attempts to alleviate such a limitation, by defining larger receptive fields via introducing deeper architectures and/or more pooling operations [18, 19]. Other techniques include using dilated convolution [40, 41] with increased sampling step size, deformable convolution [8] with learned sampling positions, SENet [19] and GENet [18] using global Avgpooling to aggregate global context and re-weight feature channels, and CBAM [38] respectively applying global Maxpooling and global Avgpooling to refine features independently in the spatial and channel dimensions.

Another way to alleviate the difficulty in capturing global cues is to use the global attention mechanism [36], with which research in natural language processing [35] has demonstrated great advantages in capturing long-distance dependencies. The non-local means method [4] and the non-local operation [36] were introduced to CNNs in a self-attentive manner so that the response at each position is a weighted sum of the features at all (global) positions. Together, attention augmented convolutional networks [3] used convolutional feature maps along with self-attentive feature maps to augment convolution operations for capturing long-range interactions. An object attention module is proposed within Relation Networks [20], which processes a set of objects simultaneously through interaction between their appearance feature and geometry.

Though exciting progress has been made to introduce global cues to CNNs, the existing approaches to solve this problem have apparent disadvantages as well. For example, in the first solution, the larger receptive fields, more intensive pooling operations are needed, hence implying lower spatial resolution. In situations when convolutional operations are not properly fused with attention mechanism, such as in the second solution, then local feature would deteriorate.

2.2. Visual Transformers

As an alternative to recurrent and convolutional neural networks, transformers [35] were first proposed for machine translation tasks and has been widely used in various NLP tasks [27]. Inspired by the breakthrough of transformers in NLP, many researchers have developed vision transformers for various image/video related tasks, and proved Transformer to be a better model to handle really long sequences, while the RNN- and CNN-based models could still work well or even better than Transformer in the short-sequences tasks.

As a pioneer among them, Dosovitskiy et al. [12] proposed ViT to be capable of validating the feasibility of pure transformer architectures for computer vision tasks. In the studies, transformer blocks either act as independent architectures, or get married with CNNs, to leverage the long-distance dependencies in tasks such as image classification [39], object detection [5], image generation [6], semantic segmentation, and image enhancement etc. Despite the progress, the self-attention mechanism in visual transformers often ignores local feature details.

In order to make up for this shortcoming, a distillation token is proposed in DeiT [32] to transfer CNN-based features to visual transformer. On the other hand, T2TViT [42] proposed using a tokenization module to recursively reorganize the image to tokens considering neighboring pixels. Additionally, the DETR method [5] proposed feeding local features extracted by CNN to the transformer encoder-decoder to model the global relationships between features in a serial fashion.

2.3. Hybrid

ViT [12] proposes using several Serial hybrid ViT (CNN $\rightarrow$ Transformer) architectures as an alternative to raw image patches, where the input sequence can be formed from feature maps of a CNN. In this hybrid model, the patch embedding projection is applied to patches extracted from a CNN feature map. Hybrid networks improve upon pure Transformers for smaller model sizes, but the gap vanishes for larger models.

Conformer [26] proposed the first dual structure which has initial CNN based stem modules followed by dual structure of stacked CNN blocks and stacked Transformer blocks. The interaction between the dual structure is via FCU applied on each pair of CNN and Transformer block.

2.4. Pre-Trained Model

Masked image encoding methods learn representations from images corrupted by masking, motivated by similar work in NLP (Generative pretraining from pixels [6] as a

pioneer among them). The ViT paper [12] studies masked patch prediction for self-supervised learning. SimCLR [26] introduced A Simple Framework for Contrastive Learning of Visual Representations in self-supervised way. BEiT [2] proposes to predict discrete tokens [34]. Zero-shot text-to-image generation [28] are following a similar approach. Most recently, MAE [16] proposes to apply Autoencoders to computer vision by using ViT [12] as an encoder on visible (unmasked) patches and a decoder to predict the missing (masked) patches. Our work will leverage the encoder of a pre-trained model of MAE [16] and use it as a starting point for transformer branch. Conformer [26] also introduced a supervised pre-trained model on ImageNet, using it as a backbone for other tasks Object Detection, Instance Segmentation. Our work will leverage the pre-trained model of Conformer [26] and use it as a starting point for convolutional branch.

3. Data

Image Classification We used a combination of ImageNet 2017 and MSCOCO 2017 for classification and detection tasks. Like the original Conformer paper, we train the backbone on ImageNet and then finetune on MSCOCO for detection tasks.

Object Detection The COCO 2017 dataset has 80 classes with 118K images in the training dataset. Each training image is annotated with both bounding boxes and instance segmentation masks for use with their respective tasks.

Style Transformation The dataset includes 300 Monet-style images and 7028 photos. The task is to build a GAN based upon the output of the Conformer block to generate 7,000 to 10,000 Monet-style images.

3.1. Evaluation

For model evaluation, we will use the evaluation criteria defined in the competitions.

Object Detection AP (Average Precision) score is evaluated for each of the classes

$$AP = \sum_n (R_n - R_{n-1}) P_n \tag{1}$$

Instance Segmentation mAP is evaluated with the mean taken over the segmentable classes.

Style Transformation MiFID (Memorization-informed Fréchet Inception Distance) is evaluated on test set, which is a modification from Fréchet Inception Distance (FID).

$$FID = ||\mu_r - \mu_g||^2 + Tr(\Sigma_r + \Sigma_g - 2(\Sigma_r \Sigma_g)^{1/2}) \tag{2}$$

where Tr$(\cdot)$ is the sum of the diagonal elements of the input. FID is calculated by computing the Fréchet distance between two Gaussians fitted to feature representations of the Inception network.

Memorization distance is defined as the minimum cosine distance of all training samples to the feature space, averaged across all user generated image samples:

$$d_{ij} = 1 - \cos(f_{gi}, f_{rj}) = 1 - \frac{f_{gi} \cdot f_{rj}}{|f_{gi}||f_{rj}|} \tag{3}$$

where f_g and f_r represent the "generated" and "real" images in feature space, and f_{gi} and f_{rj} represent the i-th and j-th vectors of f_g and f_r respectively.

$$d = \frac{1}{N} \sum_i \min_j d_{ij} \tag{4}$$

defines the minimum distance of a certain generated image i across all real images j, and then averaged across all generated images.

$$d_{thr} = \begin{cases} d & \text{If } d < \epsilon \\ 1 & \text{Otherwise} \end{cases} \tag{5}$$

then applies an empirically determined threshold ϵ and restricts the weight d to only apply when it is below this threshold. Finally, the FID is scaled by this weight:

$$MiFID = FID \cdot \frac{1}{d_{thr}} \tag{6}$$

4. Methods

CNN collects local features in a hierarchical manner via convolutional operations and retains the local cues as feature maps. The inductive biases inherent to it are believed to be beneficial for different tasks, such as translational equivariance and locality. Visual transformers are believed to aggregate global representations among the compressed patch embeddings in a soft fashion by the cascaded self-attention modules, it has recently demonstrated promising results on certain tasks, specifically image classification [12] and joint vision-language modeling [31]. However, Transformers lack some of the inductive biases and therefore do not generalize well when trained on insufficient amounts of data, and more importantly fail to expand the applicability of Transformer such that it can serve as a general-purpose backbone for computer vision, as it does for NLP and as CNNs do in vision. One way to fix this is by introducing a hierarchical Transformer whose representation is computed with Shifted windows as shown in recent work of Swin Transformer [14], which shows promising results in object detection and instance segmentation.

Another way is to design a concurrent network structure putting CNN and ViT together. Conformer [26] introduced a dual branch structure (CNN branch and Transformer branch), and showed that the CNN and Transformer branch

Initialization	Iterations	Acc@1	Acc@5
Pre-Trained Model	10K	12.3	21.2
Random	10K	11.5	20.5

Table 1. Experiment Result of Initializing Conformer with Pre-trained Model with Batch Size 180

can respectively preserve the local features and global representations to the maximum extent. It has an initial stem stage to extract initial local features (e.g., edge and texture information) which are then fed to the dual branches. This initial stage makes it difficult to leverage pre-trained CNN and ViT models, by initializing the two branches with CNN kernel and ViT transformer weights from pre-trained model of Conformer and MAE respectively. Table-1 shows that its performance is close to random initialization. We suspect this is due to the input of ViT module is not the raw image pixel as it expects, and also the interaction between the two branches are not gated, so fusing different weight distribution from pre-trained model together will be a challenge.

To tackle these two challenges, based on the Conformer architecture, we propose a simpler dual structure and slightly different feature interaction units between each pair of blocks of the two towers(conv tower and transformer tower) as shown in Fig.1. Compared to the original Conformer architecture Fig.2, the initial convolutional layers are removed, as ViT [12] shows pure Transformer architecture can work with raw image patches instead of feature maps of a CNN. This makes it easy to load the pre-trained ViT model as initialization of the transformer branch, giving the model a good starting point to fine tune on. Later, we will show that this idea actually works. Similarly, the convolutional branch also starts with raw pixels, the network design of convolutional branch follows ResNet [17] so that it is easy to load pre-trained ResNet-type models as initialization of the convolutional branch. This dual structure is organized into N repeated TransConv Blocks (seen in Fig.3).

To make the optimization and fine tuning easier, we designed the architecture such that the heads of each of the two branches has a separate loss term.

For each TransConv Block, a FIU (Feature Interaction Unit as shown in Fig.3) is designed to act as a gate between the transformer unit and convolutional unit.

4.1. Network Architecture

4.1.1 Dual Structure

CNN Branch. As shown in Fig.1, the CNN branch adopts a feature pyramid structure, where the resolution of feature maps decreases with network depth while the channel number increases. Following ResNet [17], a bottleneck contains a 1×1 down-projection convolution, a 3×3 spatial convolution, a 1×1 up-projection convolution, and a residual connection between the input and output of the bottleneck.

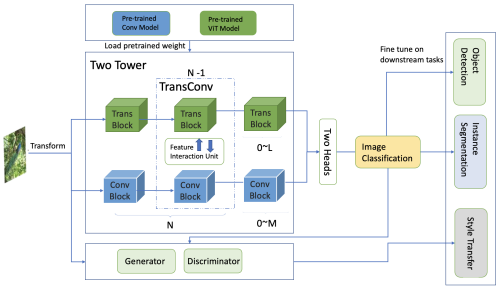

Figure 1. Network architecture of proposed RobustNet

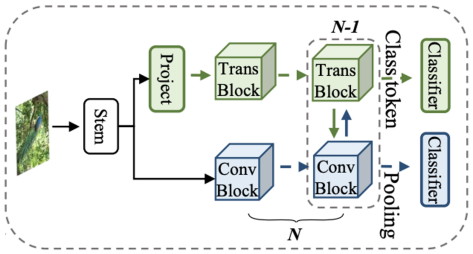

Figure 2. Network architecture of Conformer

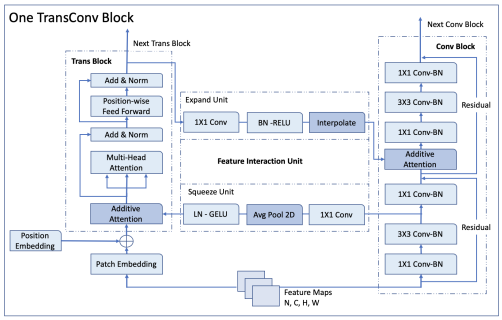

Figure 3. Each TransConv Block with Feature Interaction Unit

Transformer Branch. Following MAE [16]'s encoder ViT architecture, this branch contains N repeated transformer blocks. As shown in Fig.3, there is a special Patch Embedding Conv operation to split the raw pixel feature map into 16×16 or 14×14 patch embeddings according to small or large MAE model.

Note that the pre-trained models of ResNet and ViT may not have exactly the same blocks. So besides N-1 repeated TransConv Blocks, the proposed model can have up to L additional Trans Blocks and up to M additional Conv Blocks. Our experiments shows that it is the best to keep it the same structure as original pre-trained model in both branches, and weave them together with Feature Interaction Units in the first K blocks, where K = min(N+L, N+M).

4.1.2 TransConv Block

Trans Block. The purpose of the proposed robustnet is to combat the problem of learning both local and global representations in order to perform well in different computer vision tasks by fine tuning. The TransConv block is designed to allow information to flow from the feature map of Trans block to Conv Block via the Expand Unit, and vice versa via the Squeeze Unit as shown in Fig.1. The output of the last layer of each Trans Block is connected to Conv Block's first Conv Unit by additive attention, followed by another Conv Unit to capture spatial features from the fused feature map.

Conv Block. A Conv Block comprises of two Conv Units: Each Unit has a 1×1 down-projection convolution, a 3×3 spatial convolution, and a 1×1 up-projection convolution. The output of the 3×3 convolution is connected to Trans Block's input by additive attention. Note that it is connected to the output of the 3x3 Conv layer of the Conv Unit, instead of the output of the 1×1 up-projection convolution. This reduced the number of parameters, and also achieved better results.

Feature Interaction Unit. As shown in Fig.1, it has one Expand Unit and one Squeeze Unit. 1x1 Conv layer is used to match the Channel dimension of Conv Block and Trans Block. The Squeeze Unit is a down-sampling layer with Avg Pool. The kernel size is a configurable setting to match the dimensions of any two pre-trained models. The Expand Unit is an up-sampling layer with interpolate operations, and is used to match the spatial dimension of the Conv Block.

4.1.3 Two Heads

It is observed that different pre-trained models of CNN and ViT have different heads, ViT [12] uses cls tokens as the final output while MAE [16] uses the average pool of the tokens other than cls token. Most recent pre-trained CNN models use global average pool, while some are MLPs. It turns out to be very important to keep the head of the pre-trained model, as it allows the model to have a good start point for training. This design also allows the two separate heads to be copied exactly from pre-trained models, and each head has a separate loss function using the same one hot encoded or label smoothed labels. At testing time, the result of the two heads is simply added together.

5. Experiments

5.1. Model Variants

The experiment is done with pre-trained model released by Conformer [26] to initialize the Conv Branch and pre-trained model released by MAE [16] to initialize the Trans Branch. Conformer releases 3 models Conformer-Ti, Conformer-S and Conformer-B. MAE released 3 models ViT-Base, ViT-Large, and ViT-Huge.

Based on this, three variants are trained and evaluated with the proposed RobustNet: RN-small-patch16(ViT-Base + Conformer-S), RN-large-patch16(ViT-Large + Conformer-B), RN-base-patch14(ViT-Huge + Conformer-B).

Model	Epochs	Top-1(%)
ResNet-50 [17]	-	76.2
ResNet-101 [17]	-	77.4
ViT-B [12]	-	77.9
ViT-L [12]	-	76.5
Conformer-S [26]	300	83.4
Conformer-B [26]	300	84.1
MAE-ViT-Base [16]	-	83.6
MAE-ViT-Large [16]	-	85.9
MAE-ViT-Huge [16]	-	86.9
RN-small-patch16	20	83.6
RN-large-patch16	20	85.4
RN-base-patch14	20	**86.5**

Table 2. Top-1 accuracy for image classification on the ImageNet validation set.

5.2. Image Classification

Experimental Setting. RobustNet is trained on ImageNet-1K [10] training set with 1.3M images and tested upon the validation set. The Top-1 accuracy is reported in Table-2. Most of experimental settings follows Conformer [26], with data augmentation and regularization techniques following DeiT [33]. The model is trained for only 20 epochs with the AdamW optimizer, batchsize 180 and weight decay 0.05. The initial learning rate is set to 5e-4 and decay in a cosine schedule with a warm-up of 3 epochs. Note that this is different from Conformer: a smaller learning rate and warm-up turns out to be important for the model to converge in 20 epochs.

Performance. As shown in table-2, under similar parameters and number of layers, with just 20 epochs, RN-large-patch16 reaches 85.4, outperforming Conformer-B with 300 epochs by 1.3% (85.4 vs 84.1). RN-base-patch14 outperforms Conformer-B by 2.3% (85.4 vs 84.1). Besides superior performance, RN converges much faster than Conformer.

However, the performance is still slightly worse than MAE-ViT. We will analyze the performance of two heads separately in analysis and discussion to understand this better.

5.2.1 Analysis and Discussion

Model Convergence. To understand why the model converges so fast, we did analysis on the model training of RN-small-patch16. As shown in Fig.-8, the model starts with Test Accuracy@1 $\sim$ 79% even in the first epoch, attributed to the pre-trained models and carefully designed structure. The same training was repeated 3+ times and this result is consistent. First observation is both CNN (head 1) and Conv (head 2) are learning in the same pace. Second observation is that the model performance actually drops to 76% in epoch 6 and then steadily improves afterwards. It could be because the FIU (feature interaction unit) starts to learn and stabilizes at epoch 6, when the rest of the network starts to learn.

Understanding FIU. To understand why the model performance drops until epoch 6 and then steadily improves afterwards, we did analysis on what has been learned by looking at the weight histogram for each layer of the two branches, as shown in Fig.-9. Before epoch 6, the weight distribution of the expand block of FIU fluctuated a lot, but after epoch 6, it is following the same distribution but steadily decays. This aligned perfectly with the test accuracy@1 curve. This indicates that the dual structure needs to learn a meaningful FIU and then steadily improves the performance.

Understanding by Saliency Maps. We found that dual structure as well as ViT only model does not show meaningful saliency map compared to CNN only architecture. Fig.-6 shows the saliency map of CNN-only which maps the class objects. However Fig.-8 shows the saliency map of dual structure, and it does not show perfect match. Notice that the saliency map is with small squared boxes. This is likely due to the ViT uses patch embedding.

Understanding by CAM and Attention Map. To understand each branch's contribution, we did analysis based on class activation maps for Conv-only branch Fig.-5 on the left as well as both Conv and Transformer branch Fig.-5 on the right . It shows that they are complementary to each other.

Understand Local Vs Global feature understanding. We want to understand how RobustNet's dual structure learns both local and global features and pays attention to both. Surprisingly, Fig.-4 shows that Attention Map of the transformer branch pays attention to very local regions. The Class Activation Map of the Conv branch pays attention to larger region.

5.3. Object Detection

Experimental Setting. MMDetection lib from open-mmlab is used to do object detection. It allows pre-trained model as a backbone and fine tune on Faster R-CNN [29] model. We just add RN-small-patch16 model that we pretrained on ImageNet as a backbone in 'mmdet/models/backbones/Conformer.py'. Due to its high memory requirement, we haven't done experiments against better backbone RN-large-patch16 or RN-base-patch14. The AP on bbox is reported in Table-3.

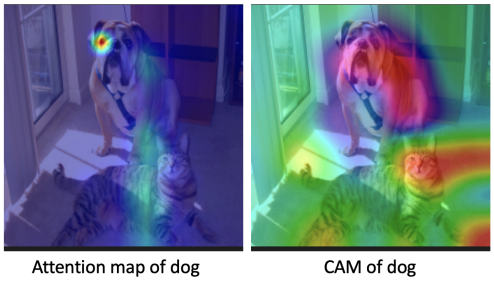

Figure 4. Local Features VS Global Features

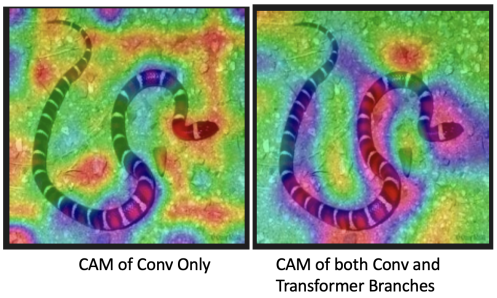

Figure 5. CAM of RobustNet

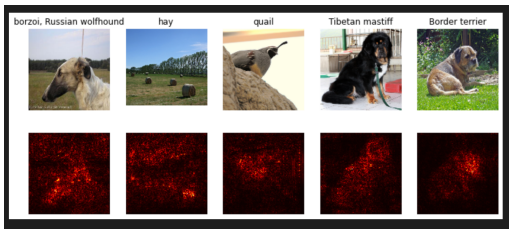

Figure 6. Salicency Map of CNN only

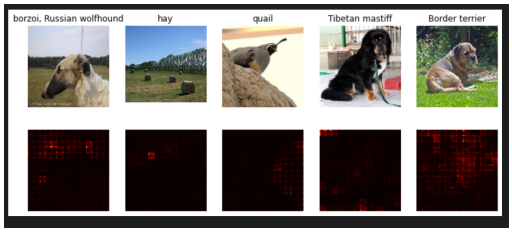

Figure 7. Salicency Map of RobustNet

Performance. As reported in Table-3, with comparable number of parameters and similar network structure, RN-small-patch16 reaches the same performance of Conformer-B. Note that we only trained it for 5 epochs and it is already reached comparable performance with Conformer-B, which

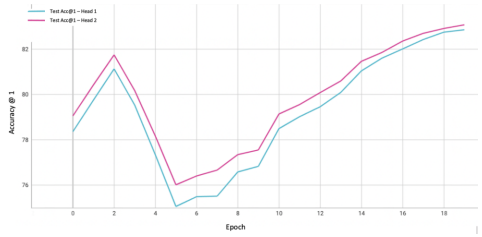

Figure 8. Why it converges so fast - Test Accuracy@1 for both branches during training

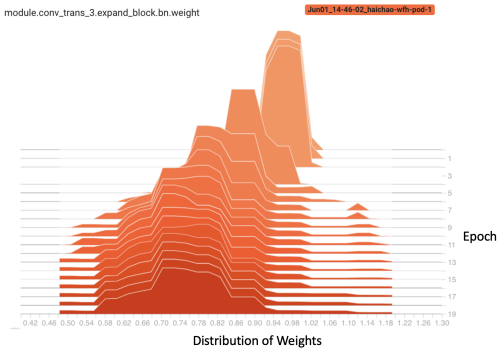

Figure 9. Understanding FIU via Weight Histogram

Model	Backbone	AP(%)
ResNet-50 [17]	-	38.2
ResNet-101 [17]	-	40.0
Conformer-S [26]	300	43.6
Conformer-B [26]	300	44.9
RN-small-patch16	20	**44.6**

Table 3. Performance for object detection on the MSCOCO minival set. Other Results are reported by the mmdetection library or Conformer Paper

was reported to be trained for 12 epochs, so it is likely that RobustNet as the new backbone is under-trained and its performance can be better.

We also show qualitative results after performing inference on the MMDetector model from the trained RN-small-patch16 model as backbone, while in figures 11 and 12 (in the Supplementary Material section), we are showing the input before and output after the MMDetector.

5.4. Style Transformation

Implementation Details For the content dataset, we are using MS-COCO, and for style dataset, WikiArt is used. During training, all images are randomly cropped into 256 by 256, while in the testing phase, any sized images are supported. As for optimizer, we adopt the Adam method, while the learning rate is set to 0.00001. Note that, the implementation of GAN is based on the existing TransGAN [21] framework, while the image generation process is inspired by the $StyTr^2$ project [11].

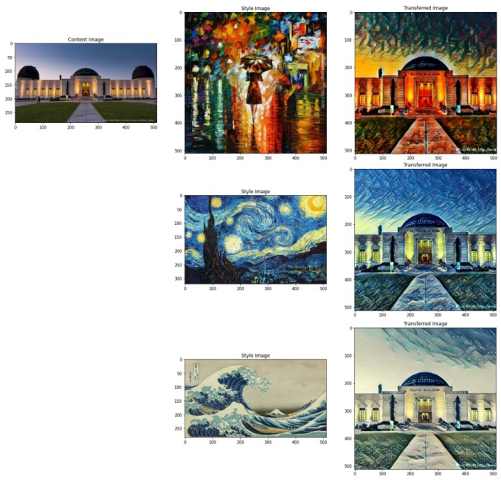

Figure 10. Visualization of a sample transferred result with three style inputs.

Comparisons with Other Methods To get a better view of how the Conformer+TransGAN (C+T) approach is doing, the model is to be compared with prevailing CNN models AdaIN, AAMS, DFP, MCC and ArtFlow, and also transformer models MSG, StyleGAN, and PGGAN.

Visualizations of Intermediate Results Before evaluation, we can first visualize the intermediate results created with matplotlib and tensorflow libraries. The feature maps (Figures 13, 14), gradients (Figures 15, 16), and sobel edges (Fig.17) are shown in the Supplementary Material section.

Qualitative Evaluation The evaluation is based on these metrics: if (1) there are sufficient style patterns, (2) the color distribution is looking reasonable, (3) the main and non-main structures in the content style both get handled, (4) there are obvious patch stitching traces, (5) any object edge overflow effect is spotted, and (6) it has strong feature representation ability. The comparison results are shown in Table 6 (in the Supplementary Material section).

By contrast, the C+T model leverages the advantages of both the CNN and transformer networks, which has better feature representation to capture long-range dependencies of input image features and to avoid missing of content and style details. Therefore, the results show that this model can achieve well-preserved content structures and desirable style patterns. A sample transferred result with different style inputs (shown in three columns: the original content image, the style inputs, and the transferred results) is visualized in Fig 10.

Quantitative Evaluation 21 style images and 10 content images were used to generate 210 stylized images. For each method to be compared with, the content differences

Losses	C+T	ArtFlow	MCC	AAMS	AdaIN
$L_{content}$	2.17	2.13	2.38	2.44	2.34
L_{style}	2.42	3.08	1.56	3.18	1.91

Table 4. QUANTITATIVE COMPARISONS. WE COMPUTE THE AVERAGE CONTENT AND STYLE LOSS VALUES OF RESULTS BY DIFFERENT METHODS TO MEASURE HOW WELL THE INPUT CONTENT AND STYLE ARE PRESERVED.

FiD per embedding	C+T	MSG	StyleGAN	PGGAN
CELEBAHQ	0.0136	0.008	0.009	0.012
FFHQ	0.01475	0.009	0.010	-
LSUN-BEDROOM	0.0533	-	0.012	0.037
LSUN-CHURCH	0.04117	0.030	0.067	0.030

Table 5. QUANTITATIVE COMPARISONS: FID VALUES COMPUTED WITH DIFFERENT EMBEDDINGS.

between stylized images and content images is calculated, and the style differences between stylized images and style images is also computed. Our goal is to see which method can achieve both low content and style loss, and also make a good balance between content and style. The quantitative comparison results for content and style losses are shown in Table-4.

FiD is yet another way to evaluate the models quantitatively (as stated in [22]). The FID values for the MS-COCO dataset computed with different embeddings are shown in Table 5, based on Equations 2-6. Note that, we have revised the Generative Memorization benchmark [1] for the computation of FiD.

Analysis Results Overall, considering the qualitative comparison results, the style/content losses, and FID values, the C+T model presents reasonable rankings across existing MS-COCO benchmarks (under self-supervised embeddings), which meets our expectation due to its balanced nature of capturing both global cues and local feature representation.

6. Conclusion

We introduced a new dual structure that fuses CNN and ViT together, we name it RobustNet, and showed that it can perform well in different kinds of Computer Vision tasks. The key contribution is that this new architecture can leverage state-of-the-art pre-trained models, and show superior performance on a wide range of tasks with comparably less training time.

We have done analysis to understand the model training, what does each layer and especially the FIU(feature interaction unit) learn along the training process via relating the model performance per epoch to key components' weight histogram.

We have used methods of network visualization to better understand what the network is doing via Saliency Maps, Class Activation Map, Attention Map etc.

Also, we have conducted Ablation Studies to understand which part of the network is essential and tentatively conclude that both Conv Branch and Transformer Branch complement to each other.

It is worth noting that while this architecture can work as a general backbone for different tasks, it comes with the cost that hybrid model being more complex and requires more memory similar to the pros and cons of ensemble method.

7. Future Works

Instance Segmentation We originally planned on applying our architecture on the Instance Segmentation task. However, due to time constraints, we were unable to finish training the network by the deadline, as it is currently on epoch 3 out of 12 at the time of submission.

Larger datasets Originally, we planned on using the OpenImages dataset for fine-tuning and evaluation for both Object Detection and Image Classification tasks. However, this dataset was very large, and fine-tuning the models on this dataset became very difficult given the time constraint for this project, so instead we used ImageNet 1000 and COCO 2017, which is a significantly smaller dataset.

In the future, we wish to apply our approaches on OpenImages and other datasets to show robustness.

One important future work is to compare this architecture with simple ensemble method to fully understand the benefit of introduced FIU(feature interaction unit) between Conv branch and Transformer branch.

8. Contributions

Haichao Wei proposed and developed the RobustNet, did the experiment and analysis on ImageNet classification, MSCOCO object detection plus the model understanding. Ethan Cheng performed data analysis and instance segmentation experiment. Chunming Peng performed the style transfer experiments and analysis.

References

[1] Ching-Yuan Bai, Hsuan-Tien Lin, Colin Raffel, and Wendy Chih wen Kan. On training sample memorization: Lessons from benchmarking generative modeling with a large-scale competition. In *Proceedings of the 27th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, Aug. 2021. 8

[2] Hangbo Bao, Dong Li, and Wei Furu. Beit: Bert pre-trained of image transformers. 2021. ArXiv abs/2106.08254 (2021): n. pag. https://arxiv.org/pdf/2106.08254.pdf. 1, 3

[3] Irwan Bello, Barret Zoph, Ashish Vaswani, Jonathon Shlens, and Quoc V. Le. Attention augmented convolutional networks. *IEEE ICCV*, page 3286–3295, 2019. 2

[4] Antoni Buades, Bartomeu Coll, and Jean-Michel Morel. A non-local algorithm for image denoising. *IEEE CVPR*, page 60–65, 2005. 2

[5] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko. End-to-end object detection with transformers. 2002. ArXiv, abs/2005.12872. https://arxiv.org/pdf/2005.12872.pdf. 1, 2

[6] Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, and Ilya Sutskever. Generative pretraining from pixels. *ICML*, pages 1691–1703, 2020. PMLR. 2

[7] Wikipedia Contributors. Two-streams hypothesis. Wikipedia, Wikimedia Foundation, 17 Dec. 2019, en.wikipedia.org/wiki/Two-streams_hypothesis. 1

[8] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. *IEEE ICCV*, page 764–773, 2017. 2

[9] Z. Dai, B. Cai, Y. Lin, and J. Chen. Up-detr: Unsupervised pre-trained for object detection with transformers. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1601–1610, 2021. https://openaccess.thecvf.com/content/CVPR2021/papers/Dai_UP-DETR_Unsupervised_pre-trained_for_Object_Detection_With_Transformers_CVPR_2021_paper.pdf. 1

[10] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and Fei-Fei Li. Imagenet: A large-scale hierarchical image database. *2009 IEEE conference on computer vision and pattern recognition*, 248–255, 2009. 1, 5

[11] Yingying Deng, Fan Tang, Weiming Dong, Chongyang Ma, Xingjia Pan, Lei Wang, and Changsheng Xu. Stytr2: Image style transfer with transformers. 2022. 7

[12] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. 2021. ArXiv, abs/2010.11929. https://arxiv.org/pdf/2010.11929.pdf. 1, 2, 3, 4, 5

[13] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby. Transformers for image recognition at scale. *ICLR*, 2021. https://openreview.net/pdf?id=YicbFdNTTy&amp=1. 1

[14] Liu et al. Swin transformer: Hierarchical vision transformer using shifted windows. *CVPR*, 2021. 3

[15] Melvyn A. Goodale and A.David Milner. Separate visual pathways for perception and action" trends in neurosciences. 1992. https://www.sciencedirect.com/science/article/pii/0166223692903448. 1

[16] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. *arXiv preprint arXiv:2111.06377*, 2021. 3, 4, 5

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015. 4, 5, 7

[18] Jie Hu, Li Shen, Samuel Albanie, Gang Sun, and Andrea Vedaldi. Gather-excite: Exploiting feature context in convolutional neural networks. 2018. arXiv preprint arXiv:1810.12348. 2

[19] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. *IEEE CVPR*, page 7132–7141, 2018. 2

[20] Han Hua, Jiayuan Gu, Zheng Zhang, Jifeng Dai, and Yichen Wei. Relation networks for object detection. *IEEE CVPR*, page 3588–3597, 2018. 2

[21] Yifan Jiang, S. Chang, and Z. Wang. Transgan: Two pure transformers can make one strong gan, and that can scale up. *Advances in Neural Information Processing Systems*, 34, 2021. 7

[22] Kaggle. Generates monet-style images. https://www.kaggle.com/competitions/gan-getting-started/overview. 1, 8

[23] Kaggle. Open image 2020 object detection. https://www.kaggle.com/competitions/open-images-object-detection-rvc-2020/overview. 1

[24] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *NeuralIPS*, page 1097–1105, 2012. 2

[25] N. Parmar, A. Vaswani, J. Uszkoreit, L. Kaiser, N. Shazeer, A. Ku, and D. Tran. Image transformer. *International Conference on Machine Learning*, pages 4055–4064, 2018. PMLR. https://arxiv.org/pdf/1802.05751.pdf. 1

[26] Z. Peng, W. Huang, S. Gu, L. Xie, Y. Wang, J. Jiao, and Q. Ye. Conformer: Local features coupling global representations for visual recognition. 2021. 2021 IEEE/CVF International Conference on Computer Vision (ICCV), 357-366. https://arxiv.org/pdf/2105.03889.pdf. 1, 2, 3, 5, 7

[27] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever. Language models are unsupervised multitask learners. 2019. OpenAI blog, vol. 1, no. 8, p. 9. 1, 2

[28] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. *ICML*, 2021. 3

[29] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. 2016. arXiv:1506.01497 [cs.CV]. 6

[30] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. 2014. arXiv preprint arXiv:1409.1556. 2

[31] Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. Vl-bert: Pre-training of generic visual-linguistic representations. *International Conference on Learning Representations*, 2020. 3

[32] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Herve J'egou. Training data-efficient image transformers and distillation through attention. 2020. arXiv preprint arXiv:2012.12877. 2

Figure 11. Inferenced input of MMDetector (Street objects)

Figure 12. Inferenced output of MMDetector (Street objects)

[33] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers and distillation through attention. 2020. 5

[34] Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. *NeurIPS*, 2017. 3

[35] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *31st Conference on Neural Information Processing Systems (NIPS 2017)*, 2017. arXiv preprint arXiv:1706.03762. 2

[36] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. *IEEE CVPR*, page 7794–7803, 2018. 2

[37] X. Wang, R. B. Girshick, A. K. Gupta, and K. He. Non-local neural networks. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7794–7803, 2018. https://arxiv.org/pdf/1711.07971.pdf. 1

[38] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. Cbam: Convolutional block attention module. *ECCV*, page 3–19, 2018. 2

[39] Bichen Wu, Chenfeng Xu, Xiaoliang Dai, Peizhao Zhang Alvin Wan, Masayoshi Tomizuka, Kurt Keutzer, and Peter Vajda. Visual transformers: Token-based image representation and processing for computer vision. 2020. arXiv preprint arXiv:2006.03677. 2

[40] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. 2015. arXiv preprint arXiv:1511.07122. 2

[41] Fisher Yu, Vladlen Koltun, and Thomas Funkhouser. Dilated residual networks. *IEEE CVPR*, page 472–480, 2017. 2

[42] Li Yuan, Yunpeng Chen, Tao Wang, Weihao Yu, Yujun Shi, Francis EH Tay, Jiashi Feng, and Shuicheng Yan. Tokens-to-token vit: Training vision transformers from scratch on imagenet. 2021. arXiv preprint arXiv:2101.11986. 2

9. Supplementary Material

Section 3. Data (Additional Figures)
Figures 11 12
Section 5. Experiments (Additional Figures)
Figures 13 14 15 16 17
Section 5. Experiments (Additional Tables)
Table 6

Figure 13. The feature maps at the first convolutional layer inside the $trans\_conv$ block.

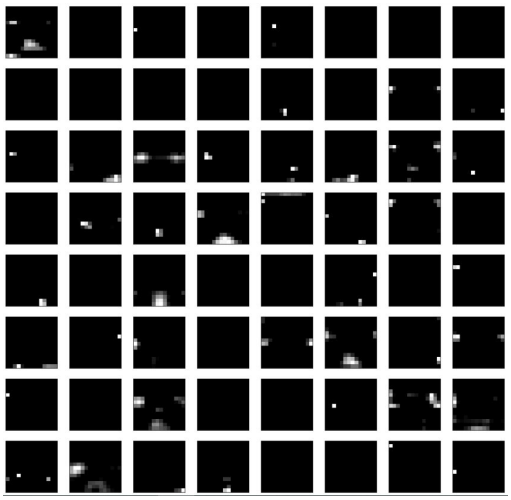

Figure 14. The feature maps at the last convolutional layer inside the $trans\_conv$ block.

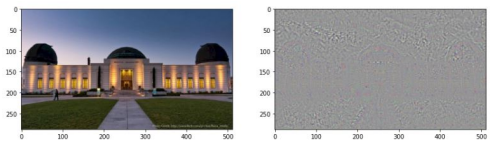

Figure 15. Visualization of the gradients at the first convolutional layer inside the $trans\_conv$ block.

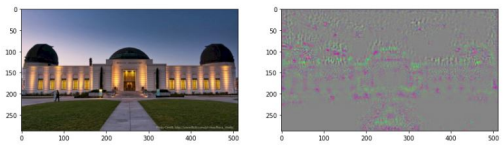

Figure 16. Visualization of the gradients at the last convolutional layer inside the $trans\_conv$ block.

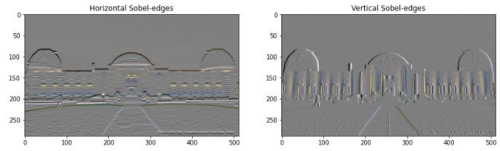

Figure 17. Visualization of the horizontal and vertical sobel edges.

Metrics	C+T	ArtFlow	MCC	AAMS	AdaIN
(1)	✓	✗			✗
(2)		✗	✗	✗	✗
(3)				✗	
(4)			✗		
(5)					✗
(6)	✓			✓	✗

Table 6. QUALITATIVE COMPARISONS. THIS TABLE SHOWS THE RESULTS BY DIFFERENT METHODS EVALUATED WITH QUALITATIVE METRICS 1-6 AT HOW WELL THE TRANSFERRED OUTPUT PERFORMS PER CATEGORY.