RGBMoME: Multi-Modal Learning of Images by Separating RGB Characteristics in Mixture of Modality Experts Architecture

MinSeo Cho
School of Electrical Engineering
Korea Univ
chominseo65@gmail.com

Abstract

This paper proposes modified structure of ViT by considering each RGB channels from image as different modalities. Encoder of proposed model consists of two parts, channel-by layer, which follows mixture of modality expert (MoME) architecture used in VLMO, and standard ViT layer. In channel-by layer, substituted feed-forward network (FFN) to channel-by FFN to learn intra-feature of each channel. In standard ViT layer, it learns inter-feature between channels. Abbreviate this structure as RGBMoME through this paper. Performances are measured on image classification task with cifar100 dataset, and showed that, as model gets deeper, RGBMoME achieved higher performance considering the number of parameters. In ablation studies, demonstrated that channel-by patch embedding layer and channel-by multi-head self-attention (MSA) composed in similar method as channel-by FFN are not useful.

1 Introduction

After foundation of transformer structure, not only in text-only tasks, achieved state-of art on many vision-only tasks with development of ViT[1] structure. Although ViT shows great performance, it requires heavier resources compared to CNN structures.

With great success of BEiT-3[2] at vision question answering (VQA) task and semantic segmentation task, its backbone architecture, MoME from VLMO[3], has proved its usefulness in vision-language tasks and vision-only tasks. In BEiT-3, only the vision encoder was used for vision-only tasks, however in this paper, as shown in figure 1, suggests applying MoME architecture inside the vision encoder, which is channel-by FFN. With this strategy, RGBMoME can learn channel-specific information in channel-by layers and learn relationship between each channel in standard ViT layers. In addition, channel-by FFN requires less parameters than original FFN.

Experiments are conducted on image classification task using cifar100[4] dataset and demonstrated that adopting channel-by layer is an effective approach not only in perspective of performance but also in the number of parameters. This channel-by block idea can be easily applied to patch embedding layer and MSA. However, as ablation studies, demonstrated that although the number of parameters drops with channel-by blocks, performance also drops significantly.

2 Related Work

ViT variation Vision Transformer has achieved state of art performance in many vision tasks. Therefore, there are many attempts to improve its performance and expand its usage by modifying structure from ViT. ChannelViT[5] modified patch embedding for better understanding of multi-channel images such as satellite images and medical images. SepViT[6] suggested depthwise self-attention and pointwise self-attention instead of standard MSA. And decreased patch resolution as layer gets deeper. VLMO applied on vision-language task by introducing MoME architecture to share MSA between vision input and text inputs.

Mixture of Modality Expert MoME architecture was first developed in VLMO. As image input and text input both can be converted into same form of patch embedding, MoME architecture treats vision language input in similar manner. Each input first goes through shared MSA, and output of MSA is separated into the ratio of the number of vision patches and text patches. Then, each goes through different FFN, which are modality experts, each called V-FFN and L-FFN. MoME has proved its performance in BeIT-3, which adopted MoME architecture with masked data modeling as pretraining method and achieved state-of-art performance in many vision-language tasks.

This paper suggests adopting MoME architecture only for image input by viewing each RGB characteristics as different modalities. Split image by channel and first propagates through shared MSA then channel-by modality experts, which are R-FFN, G-FFN, and B-FFN.

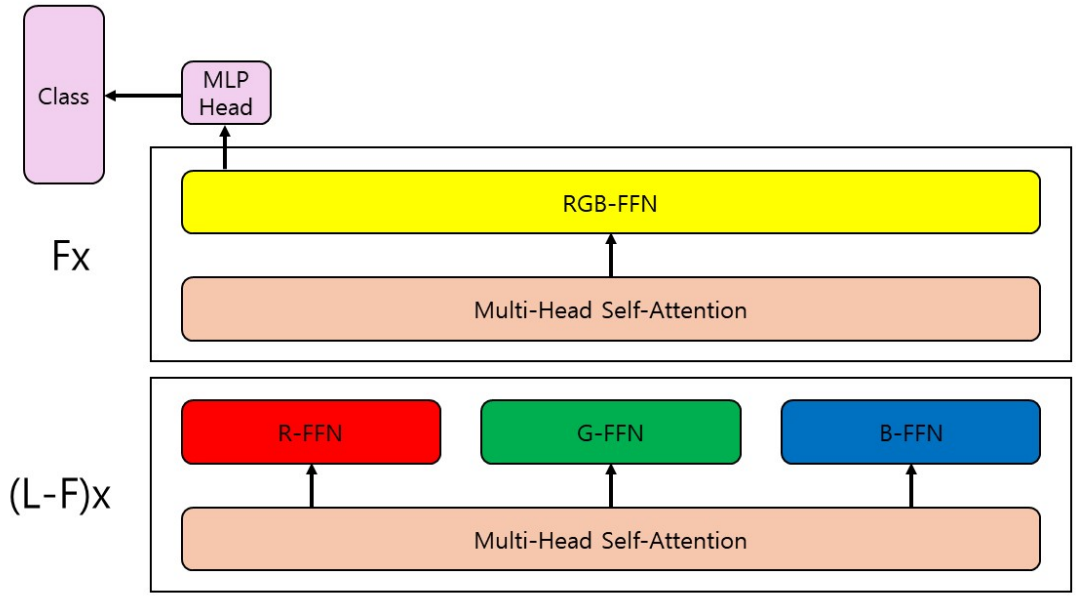

Figure 1: Illustration of encoder architecture. Similar to MoME architecture in multiway transformer in VLMO, channel-by FFN in first several layers followed by standard ViT encoder layers.

3 Methods

3.1 Overview

RGBMoME follows most of the architectural details in standard ViT. As shown in figure 1, the main method is to stack two different types of layers. Out of total L layers, firstly, stack (L-F) channel-by layers which adopted MoME architecture to view each RGB characteristics of the image as different modality, secondly, stack F standard ViT layers to view it as one image. Channel-by layer consists of two blocks. MSA and channel-by FFN, which is MoME architecture. MSA is same as standard ViT layer, but channel-by FFN splits output from MSA and calculates through 3 respective linear layers. Details will be explained in section 3.2. Plus, in this paper, not only modifying FFN to MoME architecture, tried channel-by patch embedding layer and channel-by MSA. Details will be explained in section 3.3 and 3.4 respectively. [1]

3.2 Channel-by Feed-forward Network

Output X of each MSA has shape of $X \in R^{(N+1)\times D}$. Here, N is the number of patches ($N = HW/P^2$), D is embedding size of the model. D should be divisible by the number of input channel, which is 3 for typical image input. X is splitted by the number of channels, which are $X_R \in R^{(N+1)\times \frac{D}{3}}, X_G, X_B$. Each X_C goes through different FFN. For example, with red channel, output of FFN can be written as $Y_R = W_{R2}(\sigma(W_{R1} \bullet X_R))$. Here, W_{R1} is the first

[1] model codes are available in `https://github.com/ChoChoMinSeo/RGBMoME.git`

linear layer, $W_{R1} \in R^{\acute{F} \times \frac{D}{3}}$ $\acute{F}$ is size for feed-forward layer in channel-by layer. $\sigma(X)$ is activation function, which is GELU. W_{R2} is the second linear layer, $W_{R2} \in R^{\frac{D}{3} \times \acute{F}}$. Then, $Y_R \in R^{(N+1) \times \frac{D}{3}}$ has same shape as input X_R. Lastly, by concatenating every Y_Cs, written as $Y = \{Y_R; Y_G; Y_B\}$, $Y \in R^{(N+1) \times D}$.

The number of parameters required for standard FFN is $2 \times D \times F$. However, with channel-by FFN, it requires $3 \times 2 \times \frac{D}{3} \times \acute{F}$, simplified to $2 \times D \times \acute{F}$. If $\acute{F} = F/3$, which is selected in this paper, the number of parameters is 33% of standard FFN. For better understanding, if channel-by FFN is applied at ViT-B/16 in 6 layers out of total 12 layers, the number of parameter drops from 86M to 67M, which is 22% decrease.

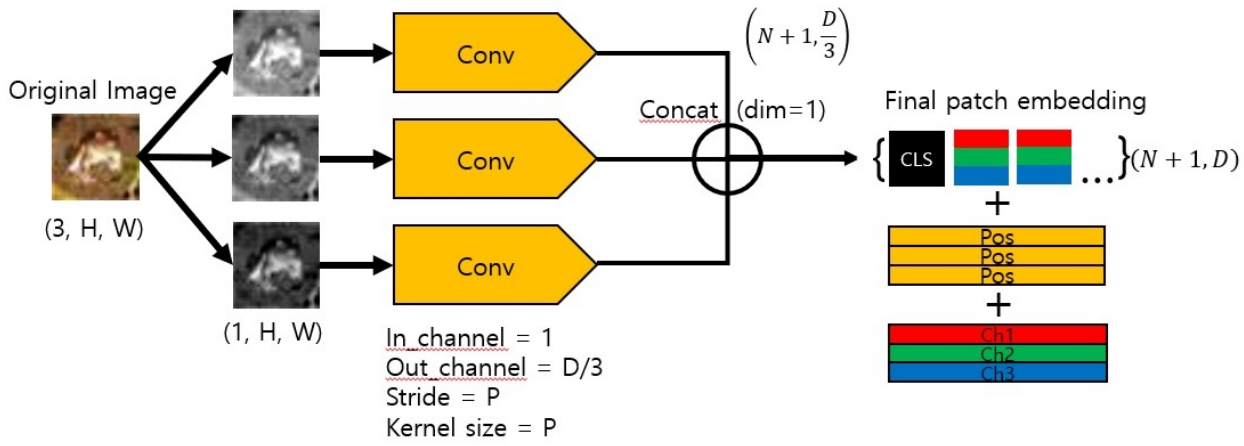

Figure 2: Illustration of channel-by patch embedding. Apply different convolution block to each channel of input to get patch embedding. Output of each convolution block is 1/3 of embedding size. After concatenating each outputs from convolution blocks, add 3 concatenation of 1/3 size of embedding size of positional embedding. And add identical channel embedding, from channelViT, to patches that are from same convolution block.

3.3 Channel-by Patch-embedding

Since inputs in VLMO are vision and language, each modality goes through independent patch embedding layer. To view RGB characteristics as different modality, implemented it as channel-by patch embedding in RGBMoME.

As shown in figure 2, final patch embedding is gathered in following steps. First, input $X \in R^{C \times H \times W}$ is splitted into $X_C \in R^{1 \times H \times W}$ for each channel. Secondly, each X_c goes through different convolution block to get channel-by patch embedding $X_{p,C} \in R^{N \times \frac{D}{C}}$ and concatenated to $X_p \in R^{N \times D}$. Since patches from different convolution blocks are from same position, partial positional embedding $[pos_1, \ldots, pos_N]$, where $pos_k \in R^{\frac{D}{C}}$, is duplicated and concatenated C times. Therefore, actual positional embedding added to X is $[\{pos_1, \ldots, pos_N\}; \{pos_1, \ldots, pos_N\}; \{pos_1, \ldots, pos_N\}] \in R^{N \times D}$. Plus, idea from ChannelViT, learnable channel embeddings $[chn_R, chn_G, chn_B]$, where $chn_C \in R^{\frac{D}{C}}$ is duplicated for each patch that actual channel embedding added to X is $[\{chn_R; chn_G; chn_B\}^1, \ldots, \{chn_R; chn_G; chn_B\}^N] \in R^{N \times D}$. Specific experiment results will be shown in section 4.2.

3.4 Channel by Multi-head Self-Attention

Instead of using shared MSA, MSA can be also modified into channel-by MSA. This approach is conducted in experimental manner to see calculating attention score within a channel is useful or not. In section 4.2, showed that it reduced the number of parameters noticeably but effected performance more negatively.

Specific method is shown in figure 3. Output of patch embedding $X \in R^{(N+1) \times D}$ is separated into each channel, $X_R \in R^{(N+1) \times \frac{D}{3}}, X_G, X_B$ and goes through different layer normalization block. Each X_C conducts linear layer to gather channel-by key (K), query (Q), and value (V) vector. For example, key vector in red channel can be expressed as $K_R = W_{RK} X_R$, where $W_{RK} \in R^{\frac{D}{3} \times \frac{D}{3}}, K_R \in R^{\frac{D}{3} \times (N+1)}$. Final key vector will be $K = \{K_R, K_G, K_B\}$, final query and value vector are gathered in same manner. Lastly, conduct self-attention calculation with final K, Q, V vectors, $softmax(\frac{QK^T}{\sqrt{d_k}})V$.

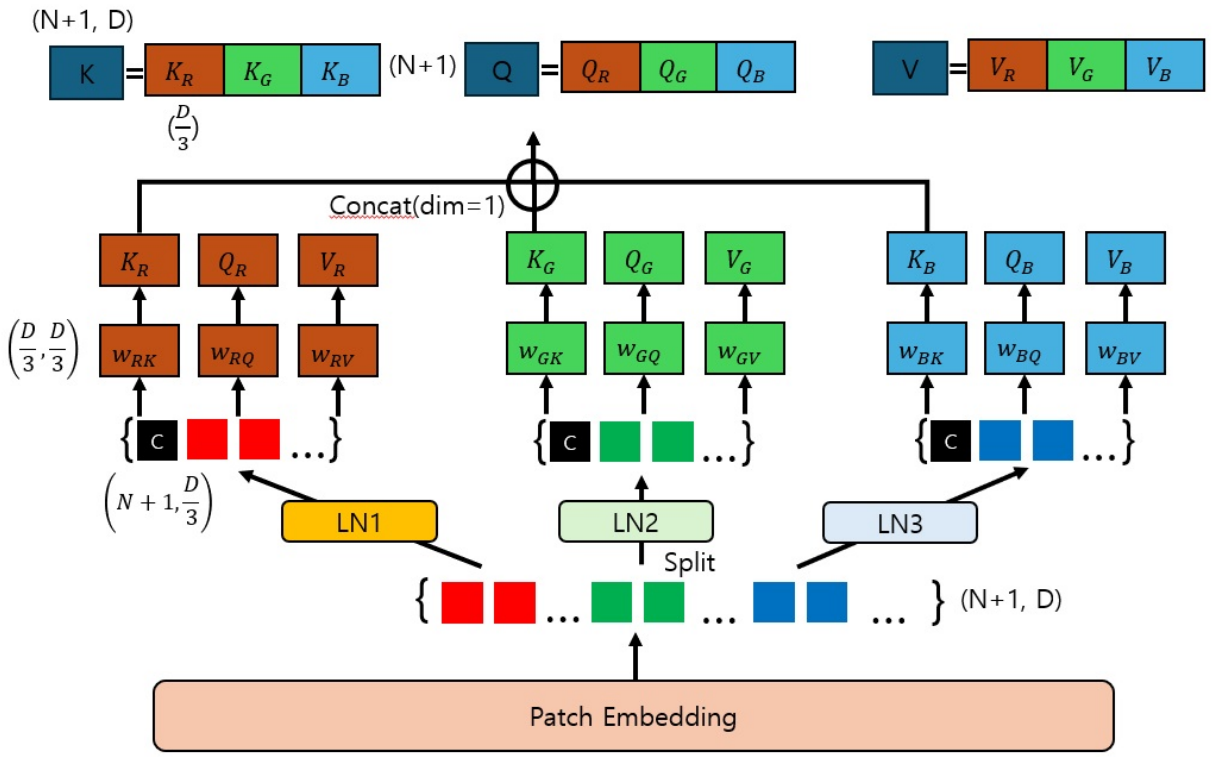

Figure 3: Illustration of channel-by multi-head embedding. Output of patch embedding layer is separated into each channel and executes independent layer normalization and linear layer to gather channel-by key (K), query (Q), and value (V) vector. K, Q, Vs from each channel get concatenated and conduct self-attention calculation, $softmax(\frac{QK^T}{\sqrt{d_k}})V$.

4 Experiments

4.1 Setup

Dataset Cifar100 dataset which has 100 classes and 50k train images and 10k test images resolution of 32x32. Considering computing resources, used used image size of 32x32 and patch size of 4x4.

Experiments Trained for 3000 epochs using AdamW with betas=(0.9,0.999), eps=1e-8, weight decay=0.05 and batch size of 256. Used cosine annealing warm restart with period, T=20, through entire epochs for learning rate scheduler.

Augmentations To give strong inductive bias, strong augmentations were applied. Sequentially, random crop with padding size 2, random horizontal flip, auto augment[7] with Cifar10 policy, normalization, lastly one of cutmix[8] and mixup[9] method were applied on train set. Only normalization was applied on test set.

Model	Layers	Hidden Size D	FFN Size	Heads	# Params
ViT-4	4	384	768	16	8.72M
ViT-10	10	384	768	16	21.71M
RGBMoME-4	2/2	384	256/768	16	4.01M
RGBMoME-10	5/5	384	256/768	16	9.94M

Table 1: Details of hyperparameters of model backbones. First number in 'layers' and 'FFN size' column of RGBMoME indicates hyperparameters in channel-by layers, second number indicates hyperparameters in standard layers.

Backbones Specific hyperparameters are shown in table 1. Compared standard ViT and RGBMoME by equally setting the number of layers. Evaluated effects of substituting width of model to depth by setting the number of layers with similar number of parameters between ViT-4 and channel-by architecture, 10.

Model	# Params	Accuracy
ViT-4	8.72M	0.7851
RGBMoME-4	4.01M	0.7839
ViT-10	21.71M	0.7860
RGBMoME-10	9.94M	0.8076

Table 2: Details of hyperparameters of model backbones. First number in 'layers' and 'FFN size' column of RGBMoME indicates hyperparameters in channel-by layers, second number indicates hyperparameters in standard layers.

# Layers	Patch embed.	# Params	Accuracy
4	standard	4.01M	0.7839
4	channel-by	4.00M	0.7704
10	standard	9.94M	0.8076
10	channel-by	9.93M	0.7944

Table 3: Comparison of with or without channel-by patch embedding in RGBMoME.

# Layers	MSA	# Params	Accuracy
4	standard	4.01M	0.7839
4	channel-by	3.21M	0.7323

Table 4: Comparison of with or without separating multi-head self-attention in RGBMoME.

# Channel-by	# Standard	# Params	Accuracy
2	2	4.00M	0.7704
1	3	4.39M	0.7781
3	1	3.61M	0.7696

Table 5: Comparison of different ratio of channel-by layer and standard ViT layer in RGBMoME.

4.2 Preliminary Result

As shown in table 2, by comparing standard ViT and RGBMoME with same depth of 4, there was a slight performance drop. Then, with increase depth of to 10, which has similar number of parameters with ViT-4, there was bigger performance gain in RGBMoME than standard ViT.

Applying channel-by patch embedding As described in section 3.3, if channel-by patch embedding is applied, as shown in table 3, there is small performance drop.

Applying channel-by Multi-head Self-attention Similarly, if channel-by multi-head self-attention is applied, as shown in table 4, there is about 20% of parameter decrease, but significant performance drop.

Changing ratio of channel-by layer and standard ViT layer As shown in table 5, with higher ratio of standard ViT layer, there was performance increase with the number of parameters.

5 Conclusion

Contributions This paper proposes to implement MoME architecture inside the vision-only encoder to view RGB characteristics as different modality. Through experiments of substituting standard blocks in ViT layer to channel-by blocks, showed that architecture with channel-by FFN outperformed standard ViT. However, substituting other channel-by blocks, channel-by patch embedding and channel-by MSA, is not an optimal choice to increase performance.

Limitations Many prior experiments indicate that even though ViT architecture has larger number of parameters than CNN architecture, ViT architecture on small datasets performs poorly than CNN architecture. However, due to limitation of time and computation resources, it was not able to perform pretraining on larger datasets, such as Imagenet 21k.

Hypothesis 'substituting standard ViT block to channel-by block will increase performance' has no mathematical reasoning. The only reason is that this architecture worked well on multi-modal task.

References

[1] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[2] Wenhui Wang, Hangbo Bao, Li Dong, Johan Bjorck, Zhiliang Peng, Qiang Liu, Kriti Aggarwal, Owais Khan Mohammed, Saksham Singhal, Subhojit Som, et al. Image as a foreign language: Beit pretraining for all vision and vision-language tasks. *arXiv preprint arXiv:2208.10442*, 2022.

[3] Hangbo Bao, Wenhui Wang, Li Dong, Qiang Liu, Owais Khan Mohammed, Kriti Aggarwal, Subhojit Som, Songhao Piao, and Furu Wei. Vlmo: Unified vision-language pre-training with mixture-of-modality-experts. *Advances in Neural Information Processing Systems*, 35:32897–32912, 2022.

[4] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[5] Yujia Bao, Srinivasan Sivanandan, and Theofanis Karaletsos. Channel vision transformers: An image is worth c x 16 x 16 words. 2023.

[6] Wei Li, Xing Wang, Xin Xia, Jie Wu, Xuefeng Xiao, Min Zheng, and Shiping Wen. Sepvit: Separable vision transformer. *arXiv preprint arXiv:2203.15380*, 2022.

[7] Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation policies from data. *arXiv preprint arXiv:1805.09501*, 2018.

[8] 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 *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6023–6032, 2019.

[9] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.