An Image is Worth 16 by 16 Words: Transformers for Image Recognition at Scale (This presentation is all you need)

By Curtis Fox

Outline

- Introduction
- Related Work and Motivation
- Architecture
- Results
- Conclusion

Introduction

Introduction

- As we've discussed in previous MLRG's, the main use of transformers has been for NLP tasks
  - This involves training on some large corpus, then doing fine-tuning on some smaller dataset
- Do not need CNN's to effectively perform image tasks, and their dominance in vision tasks is being challenged by the Vision Transformer (ViT)
- This talk will instead focus on the discussion of transformers for computer vision tasks
  - Will discuss more details about this in a bit

Related Work and Motivation

Related Work and Motivation

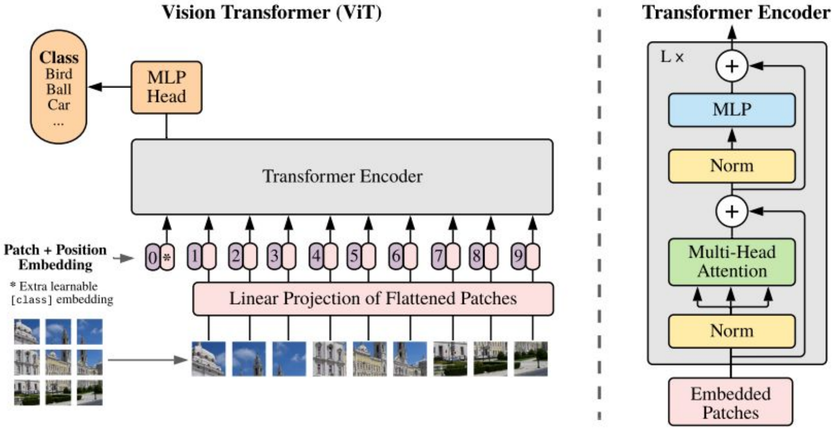
- The simplest use of self-attention for images would be to have every pixel in the imagine attend to every other pixel.
    - This has quadratic cost in the number of pixels, will not scale for larger inputs
- Another approach is to apply self-attention only in local neighbourhood for each query pixel
- Finally another approach is to do a type of approximation to global self-attention, referred to as Sparse Transformers
- The paper I will discuss takes a simple approach to scale transformers to images, which allows an almost direct application of transformers

Related Work and Motivation

- When pre-training on smaller datasets, transformers do not outperform CNN's for image tasks
  - This is because transformers lack many of the inductive biases that CNN's have for images. As in, CNN's are designed in such a way to be used for image data, which is not the case for transformers.
- Require large pre-training datasets to actually see the usefulness of the ViT models
  - This motivates the work discussed in this paper

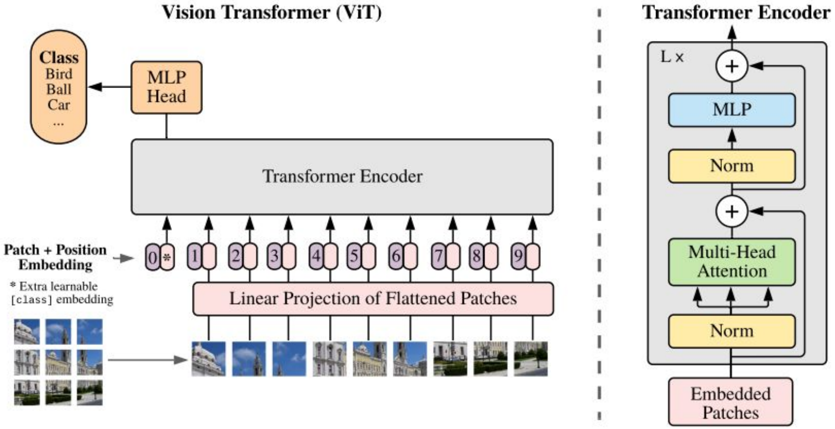
Architecture

Architecture



- Break image into patches of a chosen size (say 16 by 16 pixels)
    - Can think of these patches as corresponding to words in the NLP setting
- Flatten these patches into vectors
- Multiply these flattened patches by a matrix E
    - This is the linear projection step that transforms the input into a smaller D-dimensional space

Architecture



- Add learnable class embeddings
  - Can think of this as being used to learn labels for image
- Add learnable positional embeddings to each of the patch embeddings
  - These embeddings only hold 1D information, as the authors tried using 2D information and found this did not help much
- This is then passed to the transformer encoder that we've previously seen (same as in BERT paper)

Architecture

- The paper also discusses a hybrid model which combines transformers and CNN's
- Instead of directly multiplying the image patches by the matrix E (which performs the projection), we replace the image patches with patches extracted from a CNN feature map
    - The feature map is the result of applying a filter to the image of interest

Results

Results

Model	Layers	Hidden size D	MLP size	Heads	Params
ViT-Base	12	768	3072	12	86M
ViT-Large	24	1024	4096	16	307M
ViT-Huge	32	1280	5120	16	632M

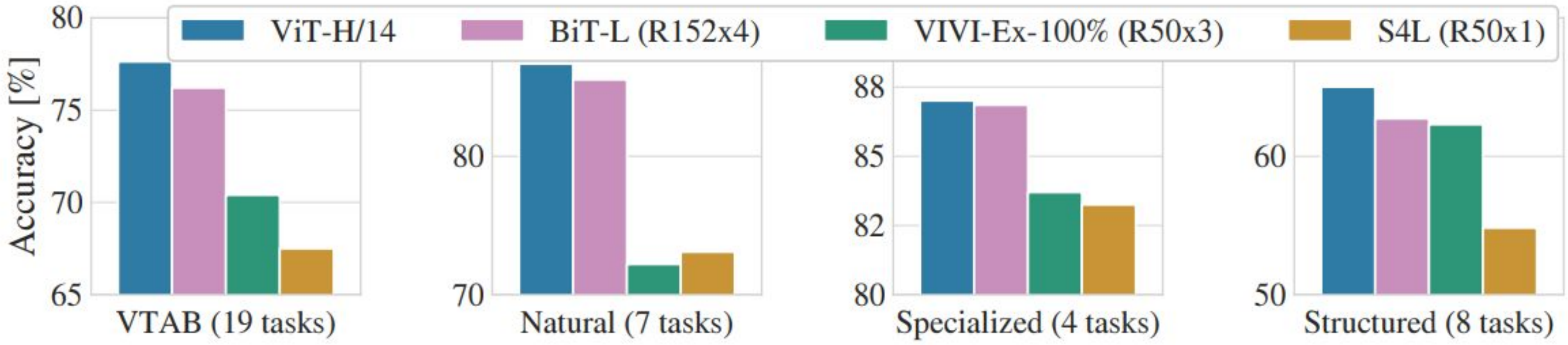
- Train 3 versions of their model, of varying sizes
- For context, the Base and Large models are taken from BERT

Results

	Ours-JFT (ViT-H/14)	Ours-JFT (ViT-L/16)	Ours-I21k (ViT-L/16)	BiT-L (ResNet152x4)	Noisy Student (EfficientNet-L2)
ImageNet	**88.55** $\pm$ 0.04	87.76 $\pm$ 0.03	85.30 $\pm$ 0.02	87.54 $\pm$ 0.02	88.4/88.5*
ImageNet ReaL	**90.72** $\pm$ 0.05	90.54 $\pm$ 0.03	88.62 $\pm$ 0.05	90.54	90.55
CIFAR-10	**99.50** $\pm$ 0.06	99.42 $\pm$ 0.03	99.15 $\pm$ 0.03	99.37 $\pm$ 0.06	–
CIFAR-100	**94.55** $\pm$ 0.04	93.90 $\pm$ 0.05	93.25 $\pm$ 0.05	93.51 $\pm$ 0.08	–
Oxford-IIIT Pets	**97.56** $\pm$ 0.03	97.32 $\pm$ 0.11	94.67 $\pm$ 0.15	96.62 $\pm$ 0.23	–
Oxford Flowers-102	99.68 $\pm$ 0.02	**99.74** $\pm$ 0.00	99.61 $\pm$ 0.02	99.63 $\pm$ 0.03	–
VTAB (19 tasks)	**77.63** $\pm$ 0.23	76.28 $\pm$ 0.46	72.72 $\pm$ 0.21	76.29 $\pm$ 1.70	–
TPUv3-core-days	2.5k	0.68k	0.23k	9.9k	12.3k

- This table summarizes the number of CPU days taken to pre-train each of the various models, as well as the achieved test accuracies on various datasets
- Overall the huge VIT model seems to perform the best
- The ViT models also took a lot less time to pre-train

Results



- This table analyzes the dataset VTAB by task group. In particular:
    - Natural: Pets, CIFAR, etc,
    - Specialized: Medical and Satellite Imagery
    - Structured: Tasks that require geometric understanding like localization
- ViT performs well across different types of task groups

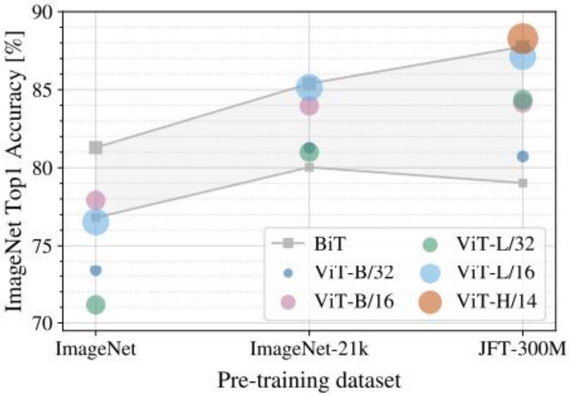
Results

Table 9: Breakdown of VTAB-1k performance across tasks.

	Caltech101	CIFAR-100	DTD	Flowers102	Pets	Sun397	SVHN	Camelyon	EuroSAT	Resisc45	Retinopathy	Clevr-Count	Clevr-Dist	DMLab	dSpr-Loc	dSpr-Ori	KITTI-Dist	sNORB-Azim	sNORB-Elev	Mean
ViT-H/14 (JFT)	95.3	85.5	75.2	99.7	97.2	65.0	88.9	83.3	96.7	91.4	76.6	91.7	63.8	53.1	79.4	63.3	84.5	33.2	51.2	77.6
ViT-L/16 (JFT)	95.4	81.9	74.3	99.7	96.7	63.5	87.4	83.6	96.5	89.7	77.1	86.4	63.1	49.7	74.5	60.5	82.2	36.2	51.1	76.3
ViT-L/16 (I21k)	90.8	84.1	74.1	99.3	92.7	61.0	80.9	82.5	95.6	85.2	75.3	70.3	56.1	41.9	74.7	64.9	79.9	30.5	41.7	72.7

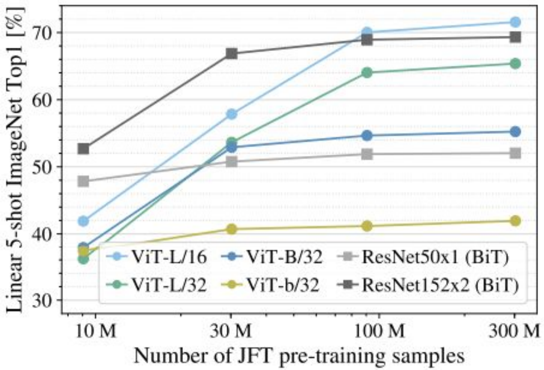
- Breaks down the task groups into their separate tasks
- Further experiments to support/justify the use of their larger model
- In some places the dataset pre-trained on makes a big difference in task accuracies

Results



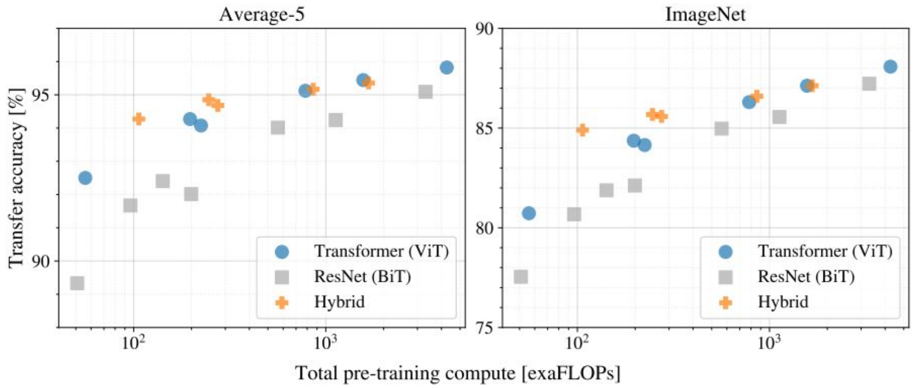
- When pre-training on smaller datasets, the ResNets outperform the ViT’s
- As you increase the pre-training dataset size, the ViT begins to perform better
  - It seems like more data really helps with ViT’s

Results



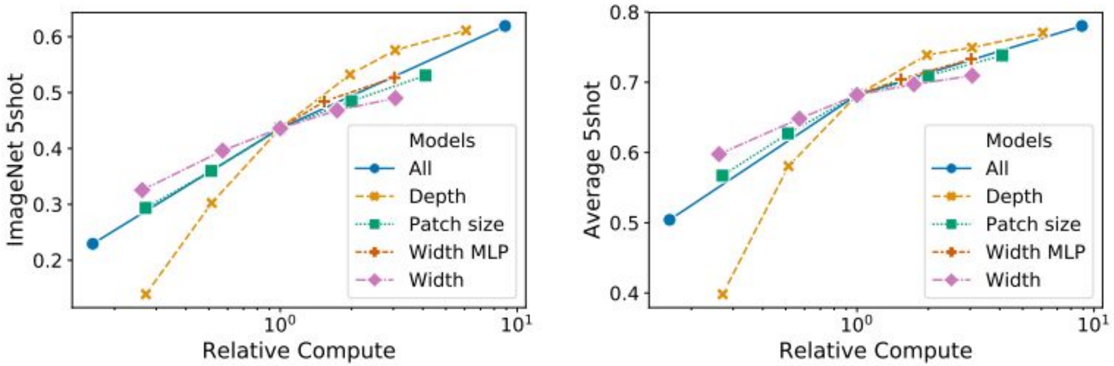
- Here a subset of the JFT is used for pre-training
  - As the subset used is increased again we eventually see the ViT models overtake ResNets

Results



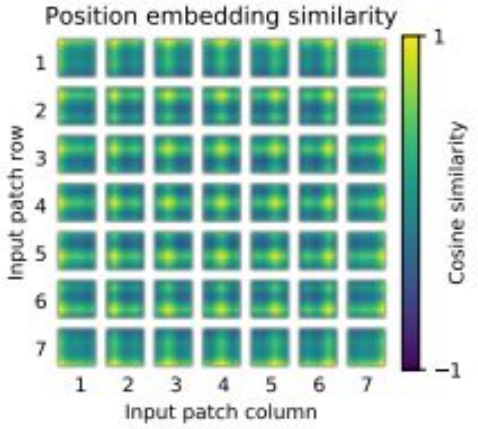
- For the same amount of computing power, the ViT models outperform ResNet
- The Hybrid models outperform the ViT models for smaller amounts of compute, but as we use more computational resources, the ViT model also outperforms the hybrid methods

Results



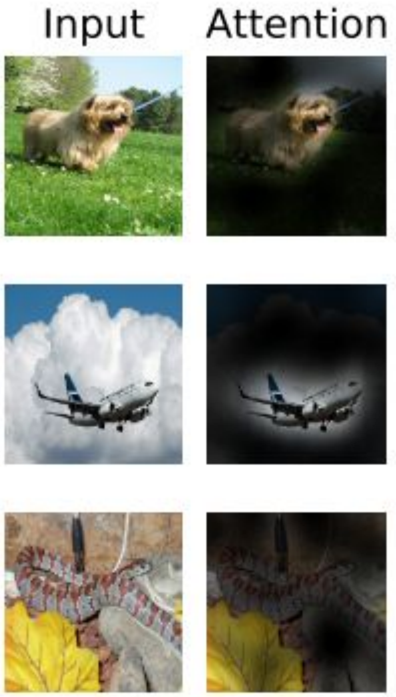
- This data shows how the error changes as we vary different parameters
- Varying all parameters proportionally seems to work well
- Varying depth seems better than width, which seems to level off
- Scaling the patch size seems to also help as well

Results



- This image shows the learned position embeddings of the model
- The ViT model actually learns to encode distance within the images provided based on the similarity of position embeddings
- Patches closer together have similar position embeddings

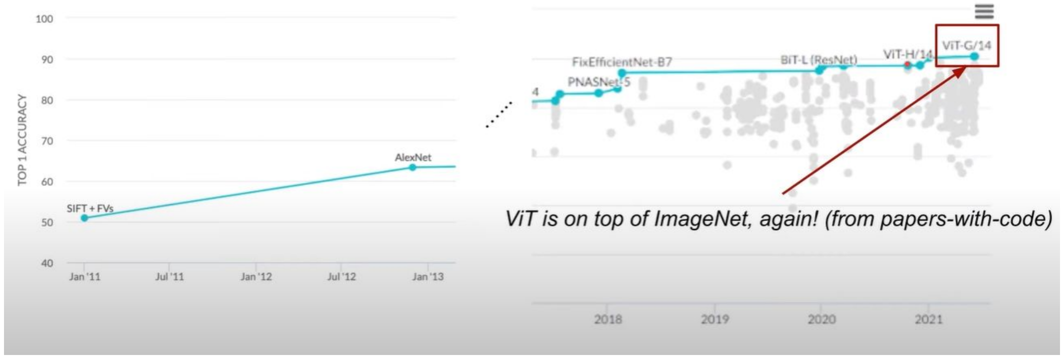
Results



- Shows that the transformer is attending to the important regions of the provided images

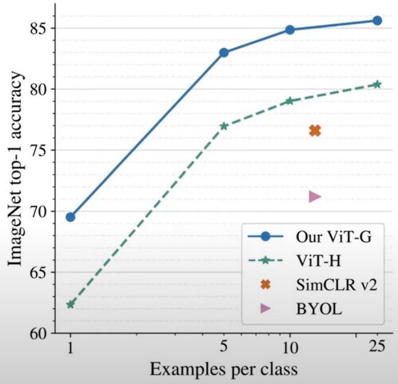
Additional Results (beyond the original paper)

Additional Results



- An even larger ViT model was trained, and beat the previous "huge" ViT model
- Also beats the ResNet results (grey dots)

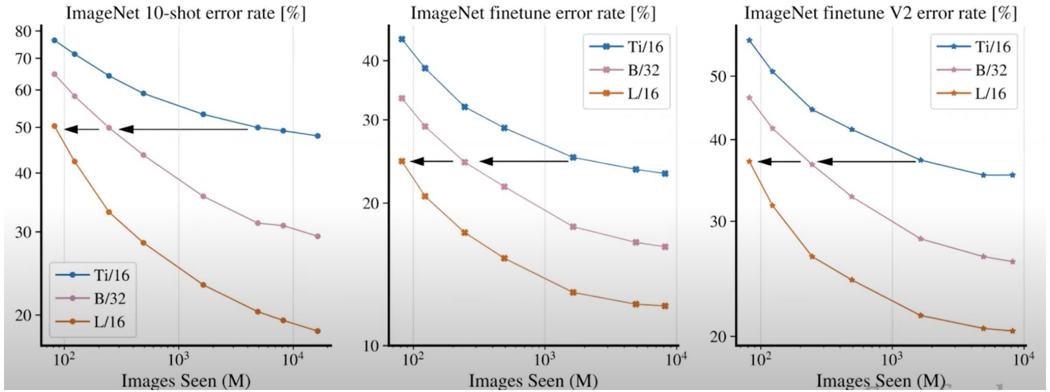
Additional Results



Benchmark	ImageNet	INet V2	INet ReaL	ObjectNet	VTAB (light)
NS (Eff.-L2) [39]	88.3	80.2	-	68.5	-
MPL (Eff.-L2) [24]	90.2	-	**91.02**	-	-
CLIP (ViT-L/14) [26]	85.4	75.9	-	**72.3**	-
ALIGN (Eff.-L2) [16]	88.6	70.1	-	-	-
BiT-L (ResNet) [18]	87.54	-	90.54	58.7	76.29
ViT-H/14 [11]	88.55	-	90.72	-	77.63
Our ViT-G/14	**90.45**±**0.03**	**83.33**±**0.03**	90.81±0.01	70.53±0.52	**78.29**±**0.53**

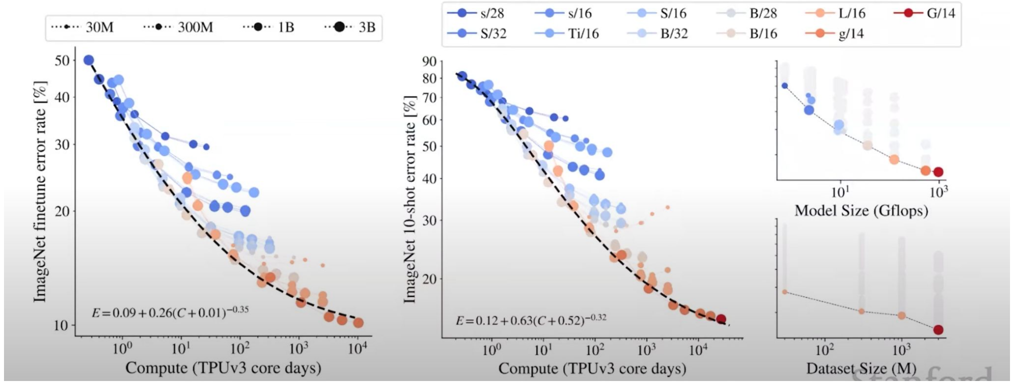
- Some more results with the larger model, showing that scaling up even further can make a difference

Additional Results



- Scaling down the ViT seems to really hurt performance
  - For the same sampling rate, the larger models perform significantly better
- As well, keeping the amount of data used for training constant and scaling down the model hurts performance significantly

Additional Results



- We see here that as we continually scale up the models and provide larger pre-training datasets we can get better results
- Also see a leveling off for each of the models, and providing larger datasets provides marginal gains in error rate improvements
- Can read more about this in the “Scaling Laws for Neural Language Models” paper

Conclusion

Conclusion

- Require significantly fewer resources to pre-train than previous methods, but perform even better
  - Increasing pre-training dataset sizes and the transformer model sizes help with performance
- Even though transformers were originally designed for NLP tasks, they have extended well to image classification tasks
  - What other tasks could transformers be used for?
- Some challenges that still remain are how transformers can be extended to other image tasks such as image segmentation

Conclusion

- Let us conclude with some issues with this work (at least in my opinion)
- Those of us with fewer computational resources cannot replicate these results, limiting this line of work to very large companies
- All of the usual bias and prejudice issues that come from labeling image data
- Main takeaway from the paper as far as I can tell is that bigger is better, which I don’t consider overly insightful

Thank you for listening!

Questions?

References

- https://arxiv.org/abs/2010.11929
- https://www.youtube.com/watch?v=BP5CM0YxbP8