Image Classification with the Transformer Architecture: A Survey

Michael Wheeler*

November 2, 2020

Abstract: This paper seeks to survey the state of the art in image classification using the Transformer neural network architecture. I motivate the task of image classification and discuss its potential applications, while also considering ethical pitfalls created by the existence of image classification solutions and their derivatives. I explore open-source labeled image datasets commonly used as benchmarks, as well as common metrics used to evaluate a model's predictive and computational performance. Finally, I summarize three state-of-the-art approaches for performing image classification with Transformers and offer questions that could guide future work.

Contents

1 Background **2**
1.1 Problem Description 2
1.2 Previous Approaches: CNNs 2
1.3 The Transformer 3
1.4 Ethical Issues 4

2 Datasets **5**
2.1 MNIST and SVHN 5
2.2 CIFAR-10 and CIFAR-100 5
2.3 STL-10 5
2.4 ImageNet 5
2.5 OpenImages 6

3 Metrics **6**
3.1 Accuracy 6
3.2 Cross-Entropy (Logarithmic) Loss 6
3.3 Visual Task Adaptation Benchmark 7
3.4 FLOPs and MACs 7

*wheeler.m@northeastern.edu. Midterm for *DS4440: Practical Neural Networks*, Fall 2020

4 Techniques **7**
4.1 Chen et al. (OpenAI) 7
4.2 Wu et al. (Facebook, UC Berkeley) 8
4.3 Dosovitskiy et al. (Google) 8
4.4 Potential Improvements 9

1 Background

1.1 Problem Description

Image classification is the task of predicting one or more classes to assign to an image given its data representation as a series of pixel values. [1] Classification in general falls under the domain of supervised machine learning, meaning that solutions involve splitting a labeled dataset into training and test sets, and evaluating the test performance of a model fit to the training data — in the case of this paper, a neural network. Image classification is a critical component of virtually all computer vision systems, and classification is preliminary to more complicated image recognition tasks; including object detection, automatic captioning, spam filtering, robotic decision-making, and disease diagnosis from medical imagery. [2, 3]

More formally, let $X = \left[\begin{array}{ccc} x_1 & \dots & x_n \end{array} \right]$ be the list of images where each image is a tensor $x \in \mathbb{R}^{W \times H \times C}$, where W and H are the image's width and height and C is the number of channels (i.e. black-and-white is $C = 1$, while red-green-blue is $C = 3$). Let $Y = \left[\begin{array}{ccc} y_1 & \dots & y_n \end{array} \right]$ be the classifications of those images, where each classification is a vector $y \in \{0, 1\}^q$, where q is the number of classes. A general classification function $f : \mathbb{R}^{W \times H \times C} \to \{0, 1\}^q$ accepts an image and outputs a binary vector predicting whether that image belongs to each of the q classes of interest. In practice, the image tensor is often transformed into a vector or matrix for ease of use with existing methods.

1.2 Previous Approaches: CNNs

There exists a rich literature surrounding the application of convolutional neural networks (CNNs) to solve image classification problems. CNNs are highly structured feed-forward neural networks consisting of layers where each hidden unit corresponds to a receptive field over the input: convolutions to extract feature representations from images based on the proximity of pixels in two dimensions, and pooling layers to reduce the size of the feature mapping with the aim of training the model to be robust to translations ("spatial invariance"). [2]

One of the earliest successes of CNNs in image classification is [4] performing optical character recognition on handwritten digits in 1989. About a decade afterwards the MNIST database was introduced, [5] encouraging further research. Improvements that followed included the introduction of max-pooling in place of average pooling, the use of the ReLU activation function, and the introduction of GPUs for training. [2] Then the SuperVision model [6] broke ground in

2012 by winning the ImageNet Large Scale Visual Recognition Challenge with a ten percentage-point increase in top-5 classification accuracy: eight layers trained on two GPUs, with dropout for model regularization and max-pooling with overlapping receptive fields. Inception [7] won in 2014: 22 layers, aiming for sparse representations across the model with 1×1 convolutions, and twelve times fewer parameters than [6]. ResNet [8] won in 2015: 152 layers, with shortcut connections between layers for better error propagation through deep networks. Variants of [8] are largely still state-of-the-art in CNNs today and are frequently used as baseline models in the literature.

1.3 The Transformer

The Transformer architecture was first demonstrated in [9] and applied to the problem of machine translation. Since then it has found broad applicability in the domain of natural language processing. The Transformer moves away from recurrent models with hidden states at each position in the input sequence. Instead the architecture relies heavily on attention mechanisms to perform encoding and decoding to the output sequence. Attention is a mathematical objective for rating the importance or relevance of each of the candidate members of a sequence (expressed as *keys* and *values*) with respect to to some other quantity (expressed as a *query*). When attention is evaluated for some query, the key representations are used to compute the importance scores of the sequence members, and the value representations are weighted by those scores against other sequence members to produce the final attention values. [10] In its original application in the decoders of recurrent neural networks, the query comes from the previous hidden decoder state, and the keys and values come from each of the encoder's hidden states (each associated most closely with one member of the input sequence). [11]

The Transformer leverages attention in groundbreaking ways to do away with the recurrent hidden state. In particular, it introduces self-attention as a special case of an attention mechanism: where the query, key, and value representations are all calculated from the same sequence. It enables the model to learn which members of the input sequence are relevant to each other. In addition attention is not calculated directly on the the input, but on linear embeddings calculated with weight matrices for the queries, keys, and values. The Transformer's multi-head attention mechanism calculates attention over multiple sets of those weight matrices — this increases likelihood of converging on a high-information feature representation, without increasing the computational cost significantly beyond that of single-head attention. [9, 12]

The original Transformer as specified in [9] accepts a sequence of input embeddings and returns a sequence of output embeddings after passing through an encoder and a decoder. The encoder consists of six identical sequential steps, with residual connection and normalization after each layer:

- first a multi-head self-attention layer,
- then a feed-forward neural network with two layers.

The decoder also consists of six steps identical to each other, with the same residual connection and normalization after each layer:

- first a multi-head self-attention layer, with masking of future positions to ensure that a prediction can only depend on previous entries in the sequence;
- then multi-head attention between the result and the encoder, where the queries are the output from the previous step, and the keys and values are the final output of the whole encoder;
- finally a feed-forward neural network with two layers.

The encoder-decoder connection is critical because it allows the decoder to attend to each position in the input sequence at every step while generating the output sequence, similar to attention applied in recurrent neural networks. [9]

1.4 Ethical Issues

The tremendous predictive power provided by the latest neural networks is not without its downsides. State-of-the-art image classification technology and its derivative applications strengthen the oppressive capabilities of authoritarian states, empowering them to build systems of mass surveillance to target individual civilians with impunity. *The New York Times* reports that the Chinese government aims to utilize large-scale computer vision systems connected to millions of surveillance cameras in order to identify individuals and constantly monitor their behavior in public spaces. [13] At its most oppressive, these software systems — built with state-of-the-art predictive models — are being used as instruments of modern genocide. At the same time that hundreds of thousands of Uighur Muslims are imprisoned in re-education camps in the province of Xinjiang, the government of China is deploying facial recognition technology over a wider geographical regions for automated racial profiling: classifying individuals as Uighurs in real-time, and tracking the activity and movements of Uighurs across large swaths of China. [14] While historically unprecedented and deeply troubling, the use of image classification technology for racist and totalitarian oppression are only expected to rise in coming years: China is now the world's largest market for surveillance technology, with several billion dollars of anticipated growth and investment in the near future. [13]

Image classification also has the potential to reinforce societal prejudice by learning the tendencies of a society whose systems of power and marginalization give rise to data reflecting its biases. [15] makes the case that the construction of machine learning models from datasets tainted by privacy and ethics problems is analogous to the sale of conflict diamonds. The authors argue that sourcing images of people from the Internet without an institutional review process has led to the inclusion of images of clearly identifiable individuals who did not consent to inclusion in the dataset, raising privacy concerns. They search through the *ImageNet ILSVRC-2012* dataset and report finding images of non-consensual nudity traceable back to the individuals via reverse image search, as well as finding non-consensual images of children in the *Open Images* dataset. In addition, the authors argue that ImageNet and other image datasets with taxonomy designed around WordNet are problematic when not scrutinized for inclusivity and impact, leading to classifications and decisions that reinforce oppressive hierarchies. A detailed examination of the *TinyImages* dataset by those authors reveals that inappropriate categories were included in the image classification taxonomy, each with thousands of labeled examples. These categories ranged from the purely "non-imageable" like "orphan", to ethnic and misogynistic slurs censored in the paper. The publication of [15] led to the maintainers of the *TinyImages* dataset to take down the dataset entirely and call for the suspension of its existing use.

2 Datasets

2.1 MNIST and SVHN

MNIST (short for *Modified National Institute of Standards and Technology*) is a dataset of 70,000 images of handwritten digits, each 28 × 28 pixels with one channel. [5] It is one of the oldest benchmark datasets in image classification, in common use since 1998. [16] A similar dataset is SVHN (short for *Street View House Numbers*), a dataset of almost 600,000 32 × 32 color images of digits displaying address numbers on the front of buildings. [17]

2.2 CIFAR-10 and CIFAR-100

CIFAR-10 is a commonly-used natural color image dataset consisting of 50,000 training images and 10,000 test images, each belonging to one of ten classes and each 32 × 32 pixels in size. A variant also exists called CIFAR-100, consisting of 500 training images and 100 test images, each belonging to one of 20 "superclasses" and one of 100 specific classes. [18]

2.3 STL-10

STL-10 is a natural color image dataset consisting of 500 training images and 100,000 unlabeled images, each belonging to one of ten classes and each 96×96 pixels in size. It was derived from ImageNet data with the goal of discovering unsupervised models that perform well on image classification tasks for which only a small number of labeled training examples exist. [19]

2.4 ImageNet

ImageNet is a large-scale image dataset generated by gathering image search engine results for a subset of nouns taken from the WordNet taxonomy. [20, 21] The dataset has grown from 1.4 million images in 2009 to over 14 million in 2014. It is the dataset associated with and used during the Large-Scale

Visual Recognition Challenge: a competition for image classification, single-object localization, and object detection held annually between 2010 and 2016. [22] The Challenge produced many breakthroughs in state-of-the-art CNNs, [6, 7, 8]making ImageNet a ubiquitous benchmark across image classification literature.

2.5 OpenImages

OpenImages V6 is one of the largest internet-based image datasets, consisting of 9 million images with 19,000 different image-level classes. It is important to note, however, that the a substantial amount of the dataset's class labels are machine-generated. [23]

3 Metrics

3.1 Accuracy

Accuracy is the ratio of correct classifications to total classifications made by a model, while Error is the ratio of incorrect to total classifications. Accuracy is by far the most common metric reported for image classification, and is generally how the computer vision community decides on "state-of-the-art" against a particular benchmark. There are however a few different ways that papers report accuracy:

- Top-n accuracy refers to the frequency with which the correct classification appears in the top n predictions made by the model. [24] Generally only top-1 accuracy is reported in the literature for single-class classification.

- Mean-per-class accuracy is the mean of the accuracy per each class c, where per-class accuracy is defined as the ratio of correct classifications of c to the total count of class c in the test set. [25]

- Linear probe accuracy: measures top-1 accuracy as a function of the number of layers. For each layer in the network, all "downstream" layers are discarded and the hidden states at that point are fed into a softmax or other transformations to produce a classification using only the features learned at that point in the model. [26]

3.2 Cross-Entropy (Logarithmic) Loss

Here let $\hat{y}_i^{(q)}$refer to the model's predicted probability of image number $i = [1, n]$ belonging to class $q = [1, C]$, and let $y_i^{(q)}$refer to the observed probability of belonging to that class (in the case of a test image whose classifications are known, a binary value). The cross-entropy loss on the set of images is defined as: [27, 28]

$$L(X, Y) - \frac{1}{n} \sum_{i=1}^{n} \sum_{q=1}^{c} y_i^{(q)} \cdot log(\hat{y}_i^{(q)})$$

Cross-entropy loss is also commonly reported, and often used as the objective for minimization in image classification.

3.3 Visual Task Adaptation Benchmark

The Visual Task Adaptation Benchmark (abbr. VTAB) [29] is a new framework for measuring the performance of computer vision classifiers against a variety of unseen tasks with as few labeled examples as possible (i.e. using pre-training and transfer learning). It consists of three sets of nineteen classification tasks whose performances are all measured in top-1 accuracy:

- *Natural tasks* are those corresponding to traditional classification tasks, and make use of some of the datasets mentioned above.
- *Specialized tasks* are those from domains that process specialized images, like medical or satellite imagery.
- *Structured tasks* are those focused on computer vision for robotic control and reinforcement learning, whose datasets consist of images of simulated graphics environments.

3.4 FLOPs and MACs

Floating-point operations (abbr. FLOPs) refers to the total number of low-level hardware computations required to perform a calculation with a model, like classification or training on a single image. This is not to be confused with floating-point operations per second (FLOPS), which is commonly used to measure the performance of a computer's processor. [30, 31] A similar measure is "multiply-and-accumulate" operations performed by the ALU, or MACs. [32] These two metrics are used to assess the computational tractability of a particular neural network architecture. Sometimes more hardware-specific metrics are given, like compute-days on a particular kind of machine, which can be difficult to compare.

4 Techniques

4.1 Chen et al. (OpenAI)

This approach by Chen et al. [33] downsizes an image and compresses colors via k-means clustering of RGB values before flattening and embedding the images. The embeddings are then passed into a Transformer closely resembling that of [9] but without the use of positional embeddings. The Transformer is pre-trained on the task of pixel auto-regression — that is, given a sequence of pixels in raster order predicting the value of the next pixel — on ImageNet ILSVRC-2012 data, CIFAR-10, CIFAR-100, and STL-10. The model is then adapted to the task of image classification, using either linear probes at the best layer or fine-tuning, and evaluated against CIFAR-10, CIFAR-100, and ImageNet. Configuration ranges between 24 and 60 layers in the Transformer's encoder and decoder and between 512 and 1536 dimensions in the embedding, resulting in between 76 million and 6.8 billion total parameters. The authors claim that their iGPT-L model attains top-1 accuracy of 96.3% on CIFAR-10, 82.8% on CIFAR-100, and 95.5% on STL-10 when classification is preformed using a linear probe of the feature representations. The authors also claim that iGPT-L attains top-1 accuracy of 99.0% on CIFAR-10 and 88.5% on CIFAR-100 after fine-tuning to image classification.

4.2 Wu et al. (Facebook, UC Berkeley)

This approach by Wu et al. [34] uses a state-of-the-art ResNet to extract convolutional feature mappings from an image. Those feature maps are passed through several three-part layers: first a tokenizer to extract semantics from the pixels, then a Transformer to perform multi-head self attention on the different tokens, and finally a projector that augments the original feature mapping with the results of the Transformer. Tokenization is the conversion of feature mappings into a set of resulting visual tokens: first by mapping from the feature map channel size to the token channel size with a learned weight matrix, and then by averaging the contribution of each pixel to each token with a second learned weight matrix. Once the visual tokens are generated they are concatenated with positional embeddings to allow the Transformer to attend to the tokens based on their location in the image. Each layer of the Transformer uses a single weight matrix to generate its output instead of a full feed-forward neural network as in [9]. Projection occurs as an element-wise addition of the existing feature mapping and another attention calculation where the query is the existing feature mapping, and the keys and values are the output tokens.

The VT-Resnet models are trained and validated on the ImageNet database. Configuration ranges are between 18 and 101 layers in the upstream ResNet, channel size of either 512 or 1024 for feature mappings, and exactly 8 tokenizer-Transformer steps with token channel size of 1024. The authors claim that the largest model that they trained (VT-R-101) attains top-1 accuracy of 81.09±3.70% on ImageNet, with 7.1 billion MACs during training — fewer than some of the baseline ResNets of equal size.

4.3 Dosovitskiy et al. (Google)

This approach by Dosovitskiy et al. [35] splits an image up into "patches" (analogous to tokens or words in natural language) before flattening all the patches to one dimension in order and embedding them. A class token and one-dimensional positional embeddings for the patches are concatenated onto the embedding before it is passed into the Transformer. The Transformer closely resembles the encoder half of [9]; only in this architecture layer normalization occurs before each layer while residual connections still occur after each layer. The same approach of patching and flattening can also be used on the feature maps of convolutional neural networks for a "hybrid" approach. After pre-training the position embeddings lose their meaning on images of different sizes, so [35] recommends scaling the pre-trained position embeddings in two dimensions to match the position of patches in new images.

The ViT models are pre-trained on ImageNet ILSVRC-2012 data, ImageNet-21k, and JFT-300M; and fine-tuned and evaluated against ImageNet, CIFAR-10, and VTAB, among others. Configuration ranges are between 12 and 36 layers in the encoder, 12 and 16 heads for attention, and hundreds to thousands of embedding dimensions to produce final models with between tens and hundreds of millions of parameters. Models were evaluated against two other state-of-the-art convolutional network architectures — Big Transfer (a ResNet) and Noisy Student (an EfficientNet) — suitable for transfer learning. The authors claim that the largest model they trained (ViT-H/14 on JFT-300) attains top-1 accuracies of 88.55±0.04% on ImageNet, 99.50±0.06% on CIFAR-10, and 77.63±0.23% macro-averaged across the VTAB suite, and took 2,500 "TPUv3-core-days" to pre-train. On a fixed computational budget for fine-tuning (measured in exaFLOPs), ViT models excel compared to Big Transfer on accuracy, performing better with limited compute for the fine-tuning task. The authors also discovered a tradeoff between ViT trained on raw pixel data versus "hybrid" feature maps: models will perform better when given the hybrid features on a small compute budget, yet better on the raw pixels on a larger compute budget. (see [35] Figure 5)

4.4 Potential Improvements

Across all three approaches overall accuracy is emphasized at the expense of detailed metrics on model performance and balance per-class. Future work could evaluate the architectures on benchmarks to generate precision and recall values for each class of images, and examine which classes these models each perform poorly on. In addition, none of the three methods share a common metric for measuring computational performance. The three architectures could be profiled side-by-side on the same hardware in order to report truly comparable performance metrics. Much work could then be done adjusting the hyperparameters — learning rates, optimizers, embedding sizes, Transformer layers — to maximize computational efficiency at smaller scales. Finally adjusting the models' steps themselves could affect overall accuracy in unknown ways: for instance a novel way of performing "patching" across an image as in [35], or using tokenization and projection on raw pixel data instead of convolutional feature maps as in [34].

References

[1] Papers with Code - Image Classification.

[2] Waseem Rawat and Zenghui Wang. Deep Convolutional Neural Networks for Image Classification: A Comprehensive Review. *Neural computation*, 29(9):2352–2449, September 2017. Place: United States.

[3] Wikipedia: Computer vision, November 2020. Page Version ID: 986546741.

[4] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1(4):541–551, December 1989.

[5] MNIST handwritten digit database, Yann LeCun, Corinna Cortes and Chris Burges.

[6] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. ImageNet classification with deep convolutional neural networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'12, pages 1097–1105, Red Hook, NY, USA, December 2012. Curran Associates Inc.

[7] C. Szegedy, Wei Liu, Yangqing Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9, June 2015. ISSN: 1063-6919.

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. *arXiv:1512.03385 [cs]*, December 2015. arXiv: 1512.03385.

[9] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pages 6000–6010, Red Hook, NY, USA, December 2017. Curran Associates Inc.

[10] Stack Exchange: What exactly are keys, queries, and values in attention mechanisms?

[11] Jay Alammar. Visualizing A Neural Machine Translation Model (Mechanics of Seq2seq Models With Attention).

[12] Jay Alammar. The Illustrated Transformer – Jay Alammar – Visualizing machine learning one concept at a time.

[13] Paul Mozur. Inside China's Dystopian Dreams: A.I., Shame and Lots of Cameras. *The New York Times*, July 2018.

[14] Paul Mozur. One Month, 500,000 Face Scans: How China Is Using A.I. to Profile a Minority. *The New York Times*, April 2019.

[15] Vinay Uday Prabhu and Abeba Birhane. Large image datasets: A pyrrhic win for computer vision? *arXiv:2006.16923 [cs, stat]*, July 2020. arXiv: 2006.16923.

[16] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, November 1998. Conference Name: Proceedings of the IEEE.

[17] The Street View House Numbers (SVHN) Dataset.

[18] CIFAR-10 and CIFAR-100 datasets.

[19] STL-10 dataset.

[20] ImageNet.

[21] J. Deng, W. Dong, R. Socher, L. Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, June 2009. ISSN: 1063-6919.

[22] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *arXiv:1409.0575 [cs]*, January 2015. arXiv: 1409.0575.

[23] Open Images V6 - Description.

[24] Stack Exchange: What is the definition of Top-n accuracy?

[25] Stack Exchange: Calculating mean per class accuracy in Multi class classification Problem?

[26] Guillaume Alain and Yoshua Bengio. Understanding intermediate layers using linear classifier probes. *arXiv:1610.01644 [cs, stat]*, November 2018. arXiv: 1610.01644.

[27] CS231n: Linear Classification.

[28] Stack Exchange: The cross-entropy error function in neural networks.

[29] Xiaohua Zhai, Joan Puigcerver, Alexander Kolesnikov, Pierre Ruyssen, Carlos Riquelme, Mario Lucic, Josip Djolonga, Andre Susano Pinto, Maxim Neumann, Alexey Dosovitskiy, Lucas Beyer, Olivier Bachem, Michael Tschannen, Marcin Michalski, Olivier Bousquet, Sylvain Gelly, and Neil Houlsby. A Large-scale Study of Representation Learning with the Visual Task Adaptation Benchmark. *arXiv:1910.04867 [cs, stat]*, February 2020. arXiv: 1910.04867.

[30] Wikipedia: FLOPS, October 2020. Page Version ID: 986377832.

[31] Stack Exchange: What is FLOPS in field of deep learning?

[32] Wikipedia: Multiply–accumulate operation, November 2020. Page Version ID: 986513284.

[33] Mark Chen, Alec Radford, Rewon Child, Jeff Wu, Heewoo Jun, David Luan, and Ilya Sutskever. Generative Pretraining from Pixels. In *Proceedings of the 37th International Conference on Machine Learning*, Vienna, Austria, July 2020. Proceedings of Machine Learning Research.

[34] Bichen Wu, Chenfeng Xu, Xiaoliang Dai, Alvin Wan, Peizhao Zhang, Masayoshi Tomizuka, Kurt Keutzer, and Peter Vajda. Visual Transformers: Token-based Image Representation and Processing for Computer Vision. *arXiv:2006.03677 [cs, eess]*, July 2020. arXiv: 2006.03677.

[35] 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. *arXiv:2010.11929 [cs]*, October 2020. arXiv: 2010.11929.