**Blindness Detection using Deep Learning Techniques**

Nicholas Gustafson, Tejaswi Gadde,

Lakshmi Chandana Shaik, Hari Chandana Datla, and Shiva Shettar

University of Maryland, Baltimore County

DATA 602: Introduction to Data Analysis and Machine Learning

Dr. Tony Diana

December 10, 2021

**Abstract**

We explore the application of state of the art deep learning techniques for ordinal image classification on images of diabetic retinopathy. In 2019, there was a large Kaggle competition featuring thousands of training images and large prizes for the competitors that produced the best results on a hidden test set. Even though the competition is closed today, Kaggle still accepts submissions and runs them against their secret hidden test set. We apply state of the art CNNs, densenet, and even new visual transformer techniques to classify the images. We were able to get competitive results with densenet despite having a severe disadvantage in computational resources compared to the winning solutions. Visual Transformers (VT) are evolving as an alternative to the architectural paradigm Convolutional Neural network (CNN).However**,** they proved to be inadequate for the task in this competition. The visual transformer approach leverages the benefits of the self-attention mechanism, obviating any convolution operations involved in commonly used deep learning models utilized for blindness detection. Transformers have recently demonstrated very good performance in a wide range of time-dependent applications, but without large and robust datasets with millions of images to pretrain them on, they are not able to perform well, often failing to do much better than a naive strategy. We leverage the free 36 hours of GPU time that Kaggle offers to it’s accounts; in the analysis of this problem we used over 100 hours of GPU time. It would have been impossible to train even a single model without the GPU acceleration, but still more resources are needed to fully explore this domain.

**Introduction**

Diabetic retinopathy is an eye condition that can cause vision loss and blindness in people who have diabetes. It affects blood vessels in the retina (the light-sensitive layer of tissue in the back of your eye). Millions of people suffer from diabetic retinopathy, the leading cause of blindness among working aged adults. The early stage of diabetic retinopathy is considered crucial and can be treated if diabetic retinopathy is diagnosed on time. The treatment of diabetic retinopathy is often made with the help of fundus images, acquired by fundus photographs(Poostachi M, Silamut K, Maude RJ, Jaegar S, 2018).With advances in computing models and technologies such as artificial intelligence (AI) and deep learning (DL), opportunities to detect diabetic retinopathy in early stages have increased significantly. This paper shows vision transformers as the alternative method that can be applied for blindness detection. The evaluates the use case of various vision transformers like CVT, Linformer ViT, LeViT, CaiT ViT when compared to Convolutional Neural Networks.

**Dataset**

The training data consists of 38,788 images of eyes from two separate Kaggle competitions. All images were transformed to 244x244 .jpg files and cropped in such a way so that the image of the eye takes up as much of the image as possible. The preprocessing was done and the results were published as a public Kaggle dataset for anyone to use. A gaussian blur was added as a filter, since there was some suggestion amongst the winning solutions from 2019 that it was helpful, but there was no consensus. Our initial modeling indicates that the gaussian blur provided minimal advantage over the raw images, but transforming 38,788 images is a computationally intensive process, so rather than regenerating the data, we kept the 38,788 images with blur.

**The Quadratic Weighted Kappa**

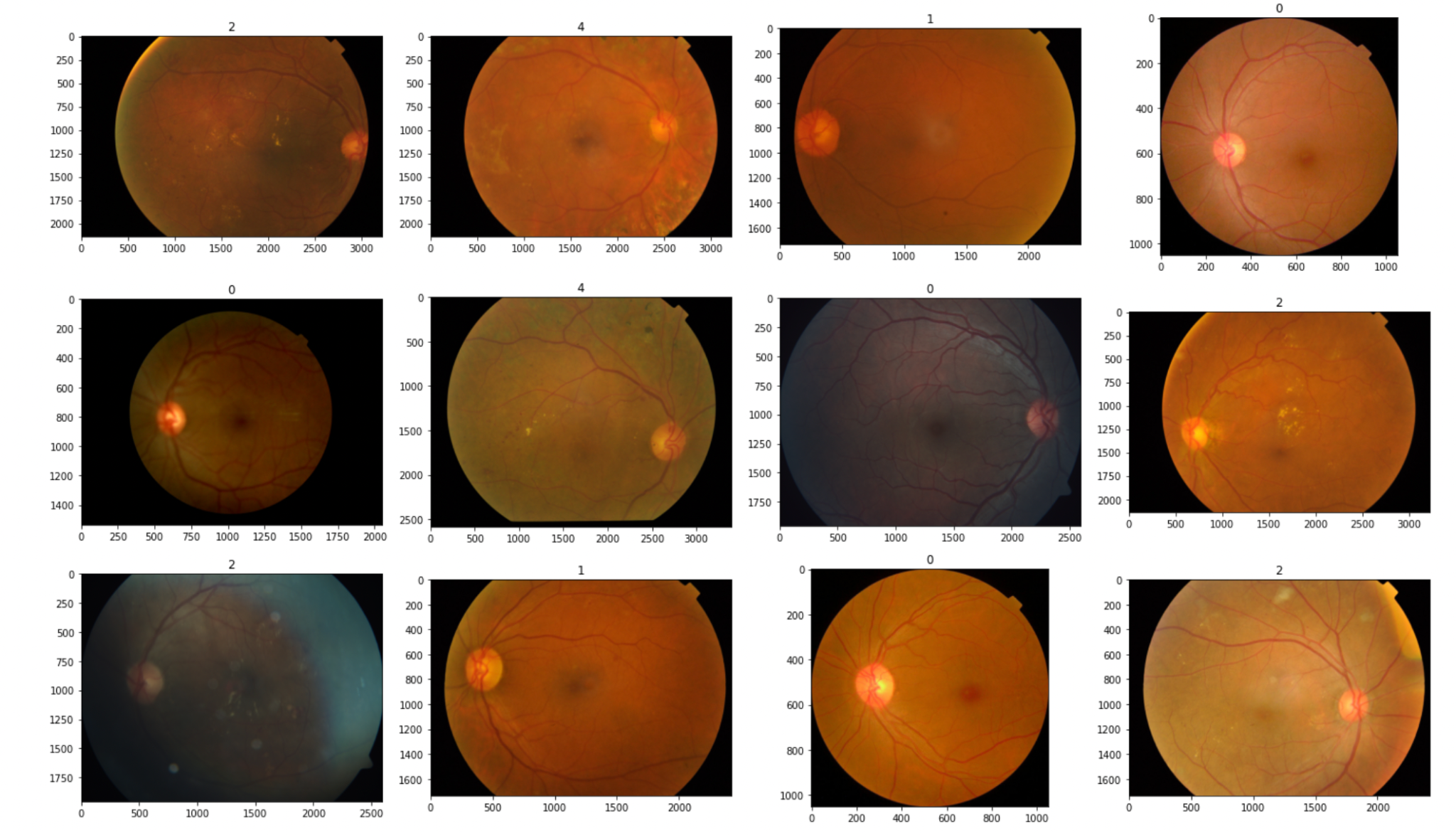
A weighted Kappa is a metric which is used to calculate the amount of similarity between predictions and actuals. A perfect score of 1.0 is granted when both the predictions and actuals are the same. Whereas, the least possible score is -1 which is given when the predictions are furthest away from actuals. In our case, consider all actuals were 0's and all predictions were 4's. This would lead to a QWKP score of -1. The aim is to get as close to 1 as possible. Generally a score of 0.6+ is considered to be a really good score(Mitra,2020).

The scores of our models in comparison to the winning solutions score:

| Model | Quadratic Weighted Kappa Score |
| --- | --- |
| Winning Solution (Large Ensemble | **0.936129** |
| Our Best Model (Densenet) | **0.87790** |
| Our Small Simple CNN | **0.629998** |
| Our Best ViT | **0.010156** |

**The Images**

The images are all of high quality, but there is the issue of zooming and cropping. As you can see from some of the images in the figure below, there is a significant amount of margin in some of the images, and this margin is inconsistent across the images. We were able to programmatically identify the border of the image by analysing the transition from light to dark. If there was a stark change from light to dark, we knew that was the start of the eye.



**Literature Review**

In the work of Wejdan L. Alyoubi, Wafaa M. Shalash, Maysoon F. Abulkhair in their paper Diabetic retinopathy detection through deep learning techniques: A review mentioned that Microaneurysms (MA) is the earliest sign of diabetic retinopathy that appears as small red round dots on the retina due to the weakness of the vessels' walls. The type of MA were seen with AOSLO reflectance and conventional imaging (Wejdan L. Alyoubi, 2020).

**CNN**

A Convolutional Neural Network (ConvNet/CNN) is a deep learning algorithm which can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image and be able to differentiate one from the other. One of the good things about CNN algorithm is the preprocessing required in this algorithm is much lower when compared to other classification algorithms.

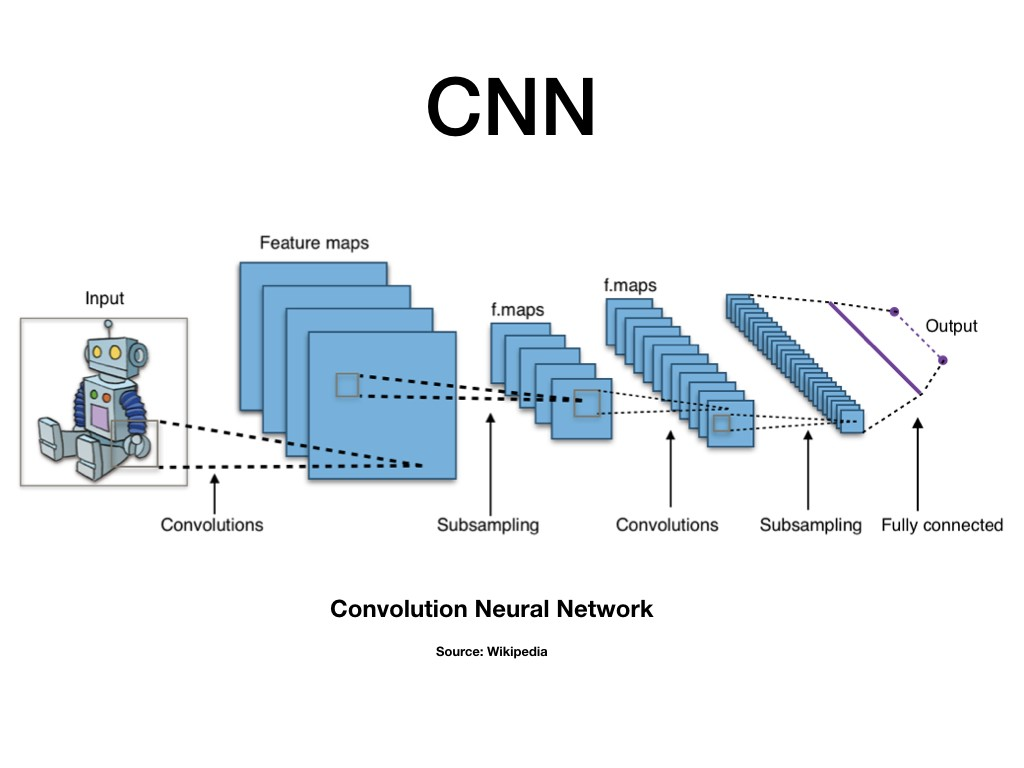


Fig-1: Image Classification using CNN (Pal, 2019)

As stated in Hanadi Achi’s article: Artificial Intelligence and Digital Microscopy Applications in Diagnostic Hematopathology, the architecture of a CNN is analogous to that of the connectivity pattern of Neurons in the human brain and was inspired by the organization of the visual cortex (Achi, 2020).

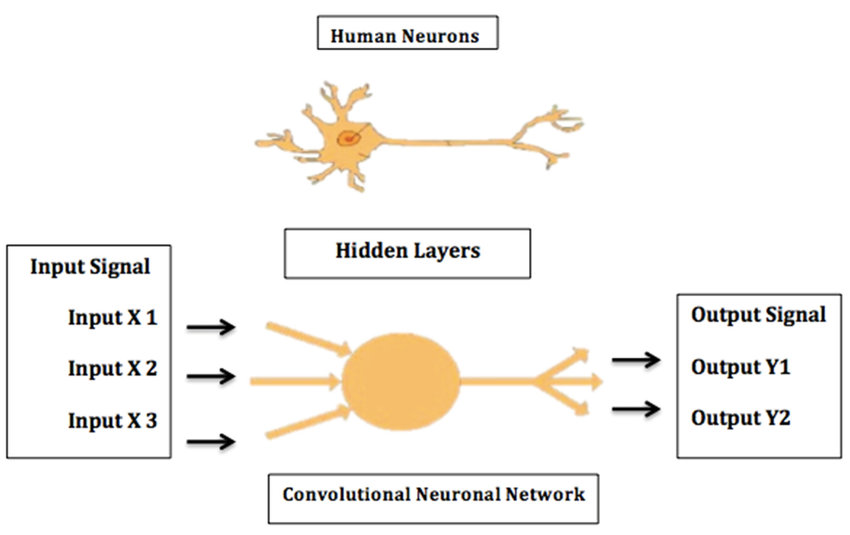


Fig - 2: A neural network is a computer system modeled on the human brain.

Let us see the flattening concept explained by Sumit Saha in 2018. For example, an image is nothing but a matrix of pixel values. So just flatten the image 3\*3 image matrix into a 9\*1 vector and feed it to a multi-level perceptron for classification purposes (Saha, 2018).

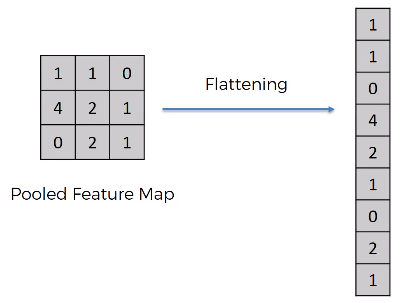


Fig 3: Flattening of a 3x3 image matrix into a 9x1 vector

The role of CNN is to reduce the images into a form which is easier to process without losing features which are critical for getting a good prediction.

**DenseNet**

A DenseNet is a type of convolutional neural network that utilizes dense connections between layers through dense blocks where we connect all layers directly with each other. As stated in Gao Huang’s paper, to preserve the feed-forward nature, each layer obtains additional inputs from all preceding layers and passes on its own feature-maps to all subsequent layers (Huang’s, 2016).

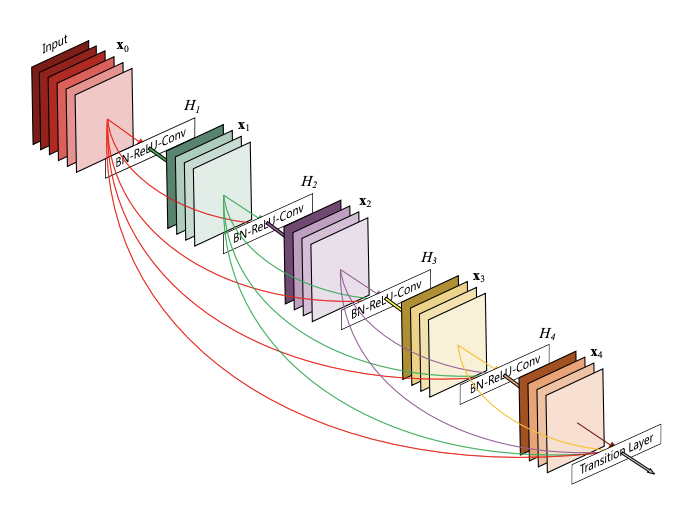


Fig:4 DenseNet (Agrahari, 2018)

To look into a small comparison, for ResNet, the identity shortcut that stabilizes training also limits its representation capacity, while DenseNet has a higher capacity with multi-layer feature concatenation as shown in Fig:4. However, the dense concatenation causes a new problem of requiring high GPU memory and more training time.

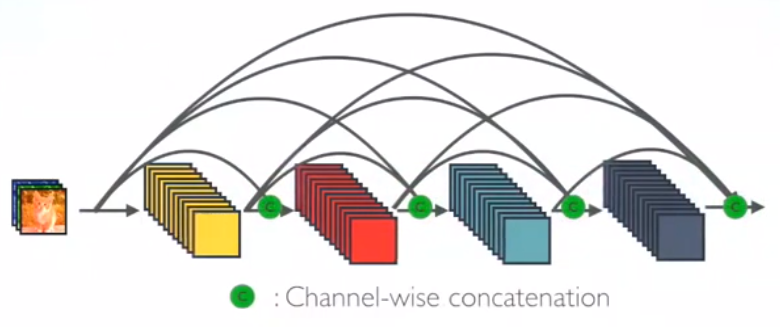


Fig:5 One Dense Block in DenseNet (Tsang, 2018)

In DenseNet, each layer obtains additional inputs from all preceding layers and passes on its own feature-maps to all subsequent layers. Here the concept of concatenation is used. Sik-Ho Tsang explained in his paper that each layer is receiving a “collective knowledge” from all preceding layers (Tsang, 2017) .

DenseNet is used for visual object recognition. ResNet uses an additive method (+) that merges the previous layer (identity) with the feature layer, whereas DenseNet concatenates (.) the output of the previous layer with the future layer as shown in the above figure-5.

DenseNet was developed specifically to improve the declined accuracy caused by the vanishing gradient in high-level neural networks. In simple words due to the longer path between the input and output layer, the information vanishes before reaching its destination.

Advantages:

* Alleviate the vanishing-gradient problem
* Strengthen feature propagation
* Encourage feature reuse

Vision Transformers

Transformers were proposed by Vaswani et al. (2017) for machine translation and have since become the state-of-the-art method in many NLP tasks.Transformers make use of attention schema which is a kind of correlation of vectorized words with each other to get the final prediction.The images are divided into patches and then they are converted to embeddings which are then fed as sequenced equivalents to the embeddings in NLP to find the attentions between one another. For computer vision tasks there are a number of different types of vision transformers to choose from. We implemented state of the art versions of the following ViTs.

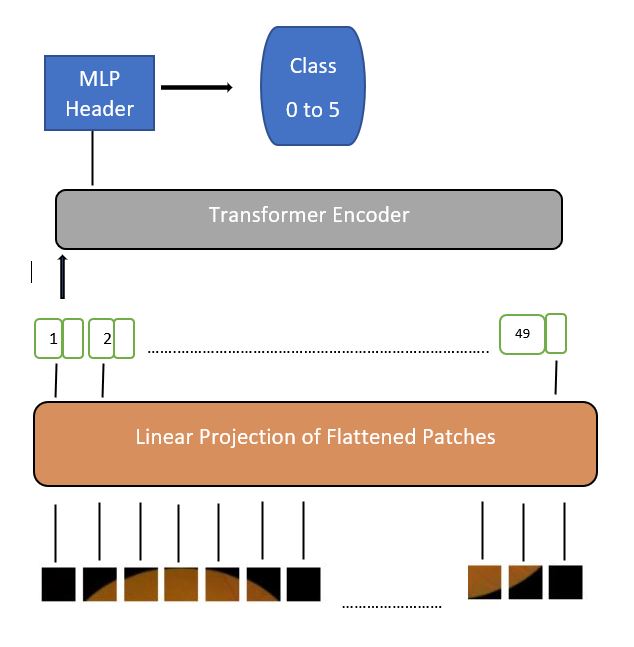


Fig:6

**Linformer ViT**

Linformer is a linear Transformer that utilizes a linear self-attention mechanism to tackle the self-attention bottleneck with Transformer models. The original scaled dot-product attention is decomposed into multiple smaller attentions through linear projections as shown in the below fig:7, such that the combination of these operations forms a low-rank factorization of the original attention.

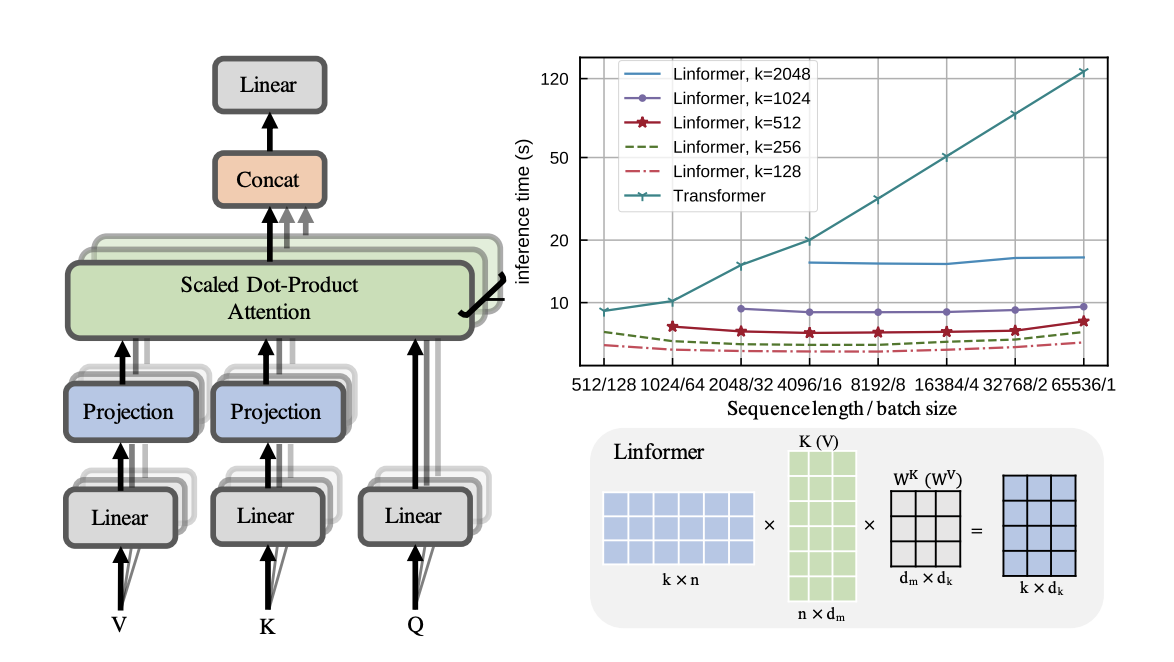


Fig:7 Left and bottom right show Multihead linear self-attention, top right shows inference time vs sequence length for various Linformer models (Wang, 2020).

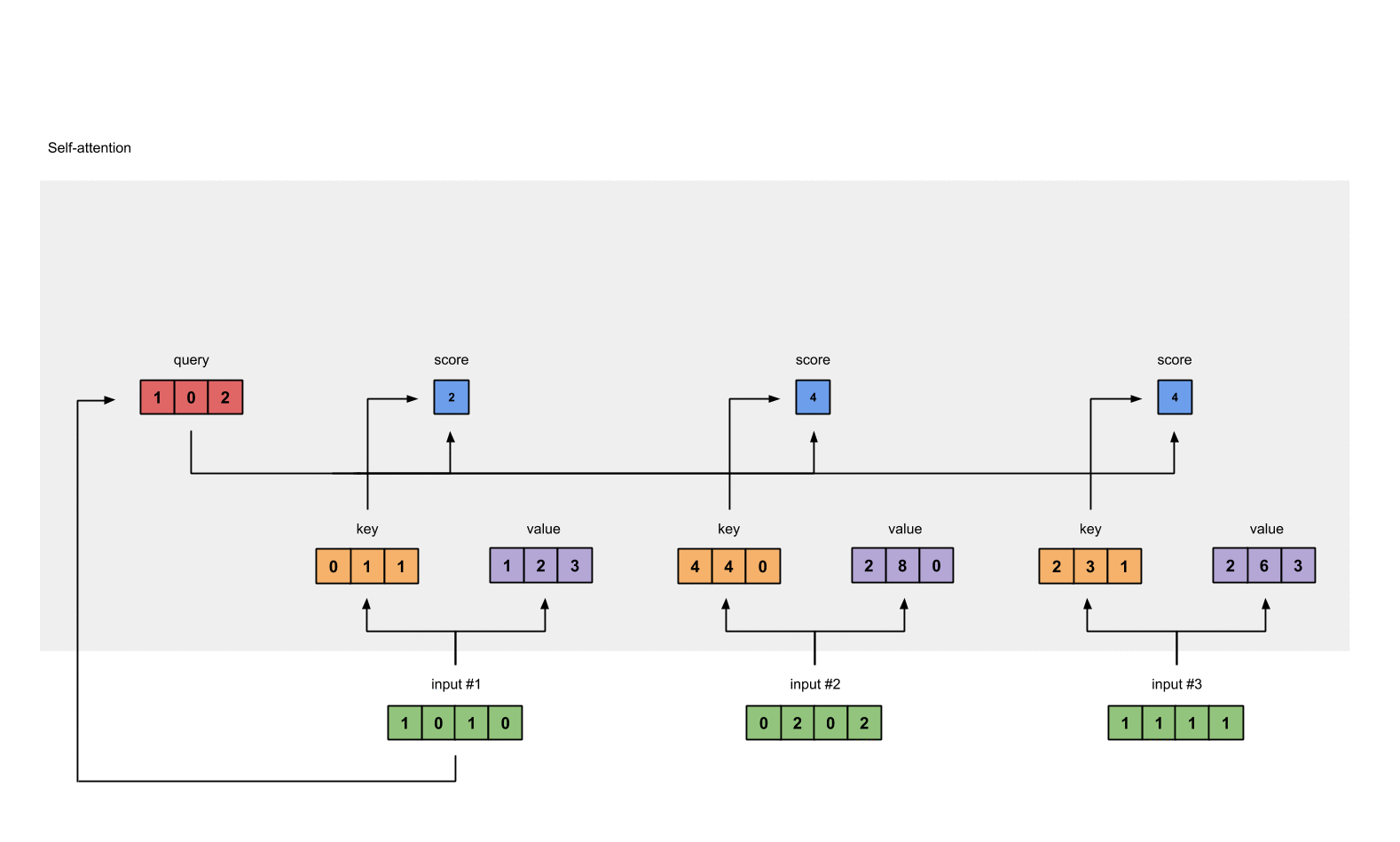


Fig:8 Deriving Key, query and value (Karim, 2019).

Raimi in his article: Illustrated: Self-Attention mentioned that, Self-attention in layman's terms is a mechanism that allows the inputs to interact with each other (“self”) and find out who they should pay more attention to (“attention”). The outputs are aggregates of these interactions and attention scores. So this module takes in n inputs and gives out n outputs (Karim, 2019).

The illustrations for this module can be divided into the below steps.

* Prepare inputs
* Initialize weights
* Derive key, query and value
* Calculate attention scores for Input 1
* Calculate softmax
* Multiply scores with values
* Sum weighted values values to get output 1
* Repeat the steps from calculating attention scores to summing the weighted values for the next inputs 2 & 3 (Karim, 2019)

**LeViT**

According to the paper by Ben Graham and team, LeViT is a hybrid neural network for fast inference image classification which significantly outperforms existing convnets and vision transformers with respect to the speed/accuracy tradeoff. From this perspective, it is said that at 80% ImageNet top-1 accuracy, LeViT is 5 times faster than EfficiencyNet on CPU (Graham & team).

LeViT builds upon the ViT (Visual Transformers) architecture and DeiT (Data efficient image Transformer) training method. Its speed comes from a series of carefully controlled design choices. When compared to other efficients neural nets used for the feature extraction in data centers or on mobile phones, LeViT is 1.5 to 5 times faster at comparable precision. Therefore it sets a new state of the art in the trade-off between accuracy and precision in the high-speed domain.

Below is the image of the LeViT architecture. The two bars on the right indicate the relative resource consumption of each layer, measured in FLOPs, and the number of parameters.

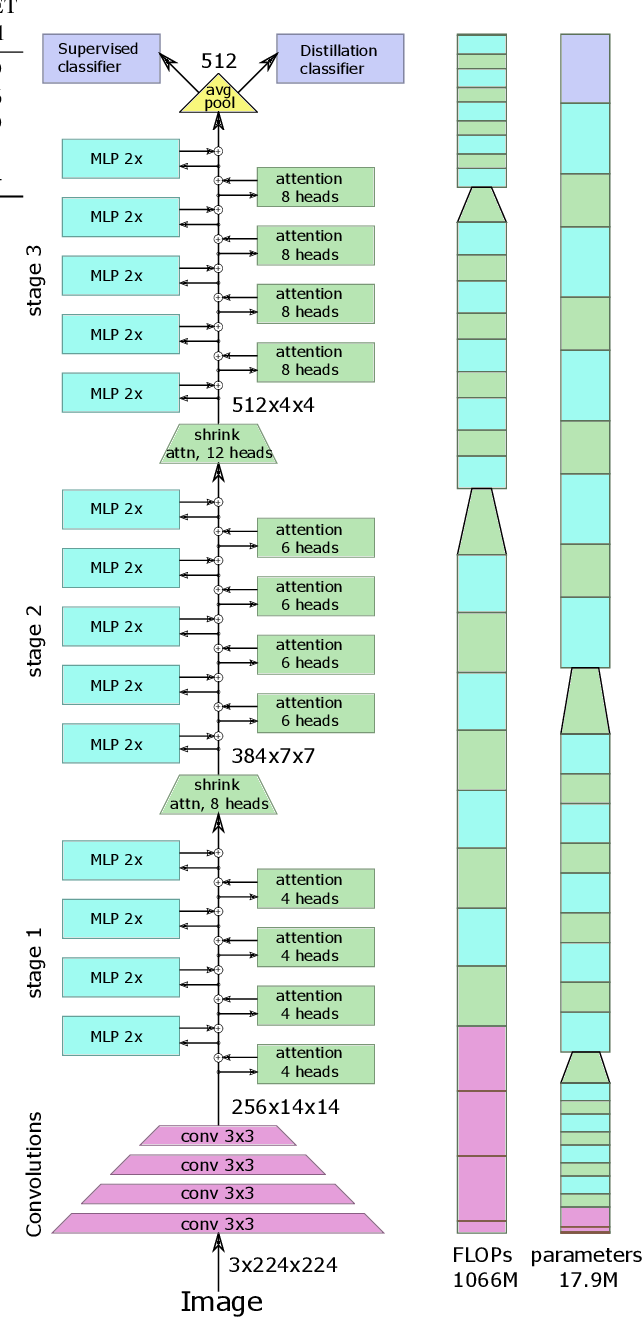


Fig:9 Block diagram of the LeViT-256 architecture (Graham & team)

**CaiT ViT**

As described in a paper by Hugo Touvron, CaiT or Class-Attention in Image Transformers is a type of vision transformer with several design alterations upon the original ViT. First a new layer scaling approach called LayerScale (a step applied to the output of each feed forward or self-attention block that allows learned parameters to control how much weight is given to the residuals vs. the output of the block) is used, adding a learnable diagonal matrix on output of each residual block, initializing close to 0, but not at 0, which improves the training dynamics (Touvron, 2021).

Secondly, class-attention layers are introduced to the architecture. This created an architecture where the transformer layers involving self-attention between patches are explicitly separated from class-attention layers that are devoted to extract the content of the processed patches into a single vector so that it can be fed to a linear classifier.

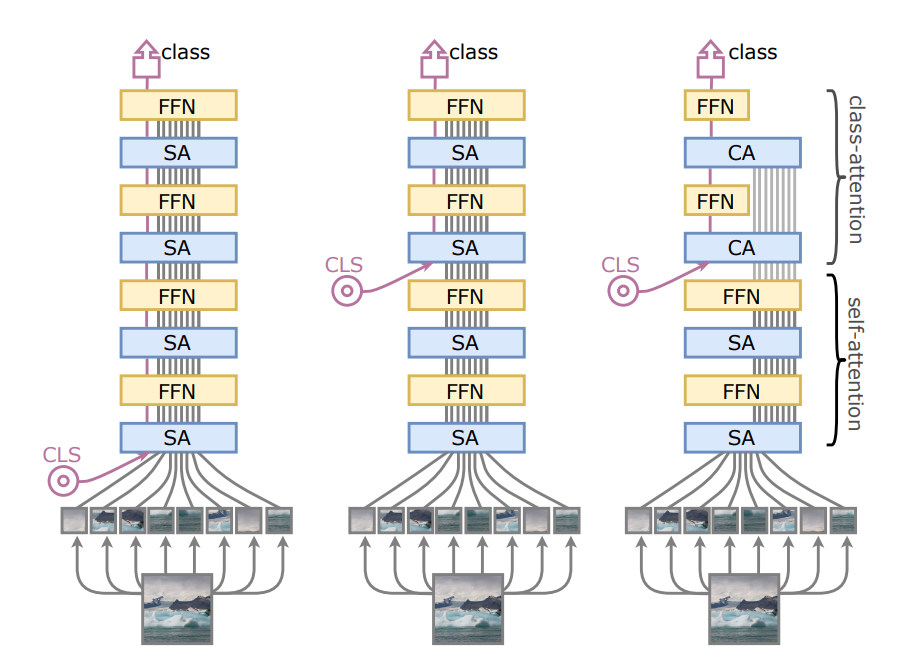


Fig:10 Class embedding in different models (Touvron, 2021)

In the above image, in the ViT transformer (on the left), the class embedding (CLS) is inserted along with the patch embeddings. This choice is taken as detrimental, as the same weights are used for two different purposes which is helping the attention process and preparing the vector to be fed to the classifier.

In the same image, we put this same problem in evidence by showing that inserting CLS later improves performance (at the middle).To the right side in the image, the CaiT architecture is proposed to freeze the patch embeddings when inserting CLS to save compute, so that the last part of the network (which is typically 2 layers) is fully devoted to summarizing the information to be fed to the linear classifier.

**CVT**

Kevin Clark has researched and mentioned in his paper that unsupervised representation learning algorithms improve the accuracy of many supervised NLP models, mainly because they take the advantage of large amounts of unlabeled text. Whereas supervised models only learn from task-specified labeled data during the main training phase. Therefore, finally a semi-supervised learning algorithm Cross-View Training (CVT) is proposed which improves the representations of a Bi-LSTM sentence encoder using a mix of labeled and unlabeled data. On labeled examples, standard supervised learning is used. On unlabeled examples, CVT teaches auxiliary prediction modules that see restricted views of input to match the predictions of the full model seeing the whole input (Clark, 2018).

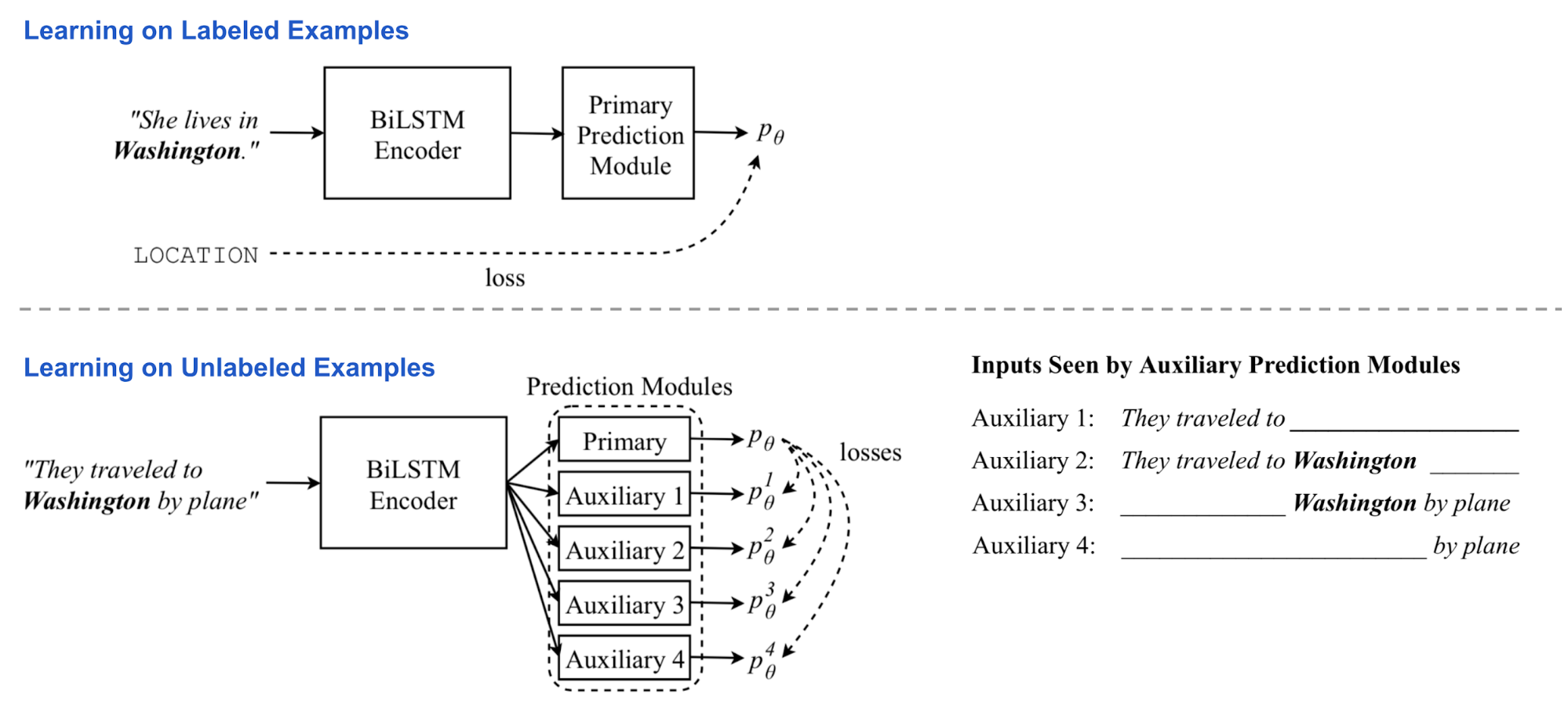


Fig:11 Inputs seen by Auxiliary Prediction Modules (Clark, 2018)

**Methodology**

The problem is simple, given these 38,788 images and labels, train a model to predict disease activity from 0, 1, 2, 3, to 4. With 0 being the most mild, and 4 being the most severe. This is not just an ordinary multiclass classification problem; it is an ordinal classification problem. It is better to guess closer to the severity than farther away. Ex. better to guess 3 when it is truly a 4, than to guess a 1.

Initially we ignored the ordinal component of the classes and modeled the problem as a simple image classification problem. We used the CrossEntropy function in pytorch, which requires that the output be a vector of floats of length 5, one float for each class. The target is a single integer.

| No DR = 0 | Mild = 1 | Moderate = 2 | Severe = 3 | Proliferative DR = 4 |
| --- | --- | --- | --- | --- |

Model output = <p0, p1, p2, p3, p*4*>}where p*i* is a real number. With categorical cross entropy Loss.

There is a trick, however, that allows us to encode the labels in a way that gives the model A trick that I used is to do a special hot encoding of the labels, like so:

| No DR  [1,0,0,0,0] | Mild  [1,1,0,0,0] | Moderate  [1,1,1,0,0] | Severe [1,1,1,1,0] | Proliferative DR  [1,1,1,1,1] |
| --- | --- | --- | --- | --- |

Model output = <p0, p1, p2, p3, p*4*> where 0 <= pi <= 1. With binary cross entropy loss.

Then binary cross entropy was used as the loss function, but to do that we need to ensure that the model outputs are between 0 and 1, so a sigmoid function is added to the final layer of the model.

This encoding trick didn’t improve the performance of the transformer models much, since there were likely other limitations holding the model back, like the size of the images and the class imbalance. We tested dozens of different models with each of the above methods, but only the CNN type models were able to benefit from the new encoding.

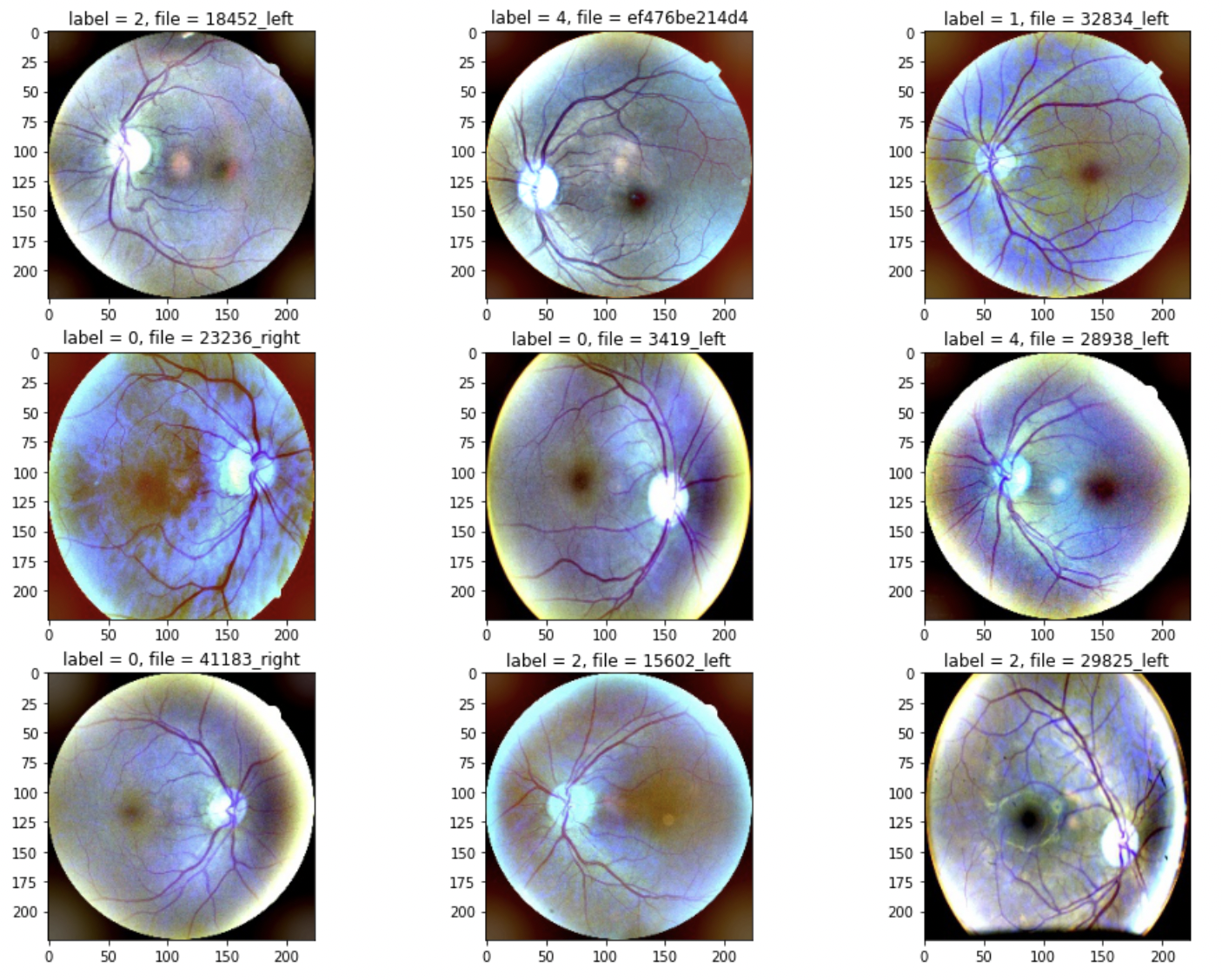


Fig: Cropped, resized, and gaussian blurred images

**Training and Evaluation**

In order to monitor the health of the training process, a 5% validation set of images was held out from the dataset. It was stratified so that the distribution of classes in the validation set matched the training data.

A significant challenge associated with the dataset is the massive class imbalance.

| No DR  71.2 % | Mild  16.2% | Moderate  7.2% | Severe  2.7% | Proliferative DR  2.5% |
| --- | --- | --- | --- | --- |

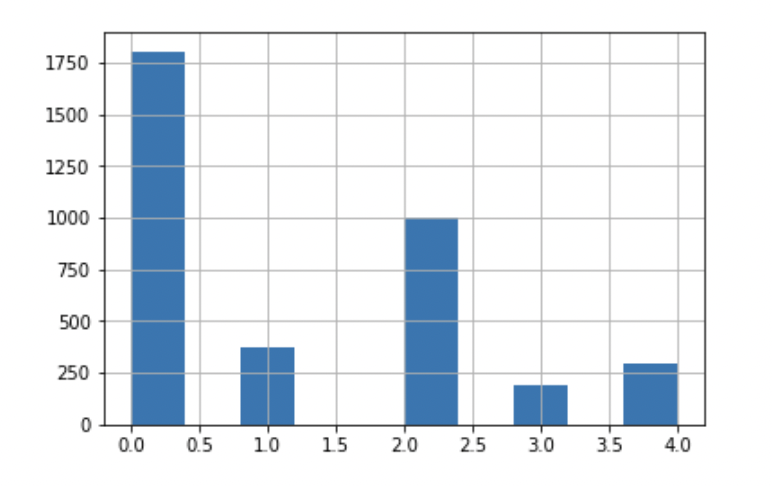


Figure: Number of images in each class. The classes are highly imbalanced.

Initially we tried to train on the whole dataset, without using any class imbalance mitigation strategies. The results were good, but often the models — the transformers in particular — would regress to only predicting a single class. The models would exclusively predict the class 0 (sometimes 2) in order to minimize the loss function in the best way that it could. This is an example of a local minimum that we would like to avoid — especially when evaluating the model with the quadratic weighted kappa, since it would heavily punish the strategy of always predicting a single class.

At first we tried upsampling the dataset, which means that we just augmented the dataset with extra versions of the underrepresented classes. We didn’t have to create copies of the images, though, we just supplied the training loop with certain images multiple times in each epoch.

The results of the upsampling were promising, but quickly we realized that we did not have enough computational resources to process even more images than usual, so the logical next step was to downsample.

Downsampling allowed us to make sure that each class was equally represented in the training data, so the model was not incentivized to only predict one class. The results were promising on the densenet, but the sampling was still not enough to get the vision transformers out of their local minimums. Even with the balanced training data, many of the visual transformers still only predicted a single class on the training data.

Confusion Matrix of Best Model (densenet) on Holdout Data.

| No DR | Mild | Moderate | Severe | Proliferate |
| --- | --- | --- | --- | --- |
| 33 | 16 | 4 | 1 | 0 |
| 3 | 63 | 6 | 0 | 0 |
| 0 | 8 | 57 | 4 | 2 |
| 0 | 1 | 8 | 60 | 1 |
| 0 | 0 | 3 | 8 | 47 |



Fig: y = Binary Cross Entropy, x = Training Epochs, of best densenet.

**Conclusion**

Our best model mimicked the kind of model that won the competition. The densenet was a successful application of a deep pre-trained neural network. Transfer learning seemed to be the key to doing well in this problem space.

The success of transformers in natural language processing is fueled by transfer learning. BERT, one of the most successful NLP models, is trained on 3.3 Billion words from BooksCorpus and English Wikipedia (Devlin et. al.,2019). The densenet that we used to get our best score was pre trained on ImageNet, a massive open source image classification dataset. The densenet had already learned a rich set of features that can be applied to a diverse range of image classification tasks.

Unlike in NLP there aren't many pretrained options for visual transformers, so we needed to train our transformer models from scratch. With such a small data set (in comparison to ImageNet) we were unable to train a useful ViT classifier, despite using many state of the art methods to augment the training process. However, the densenet we trained was able to achieve excellent results. The densenet is a similar model to the winning solution, but in our case we had computational limitations that forced us to use small 224x224 images and only one model. The winning solution was able to use an ensemble of 8 models, each using images larger than ours. This kind of model is effective, but it requires significant computational resources.

What we learned from this project is that for computer vision, transfer learning is king. If you have access to a model that is pretrained, you will almost always perform better on a computer vision task. While state of the art methods like visual transformers are able to perform well on datasets like ImageNet, with millions of images, and massive parallel computing resources, you will not be able to train a robust model.

**References**

Agrahari, R. (2018, February 2). *Implementing DenseNet on Mura using pytorch*. Medium. Retrieved December 10, 2021, from https://pyaf.medium.com/implementing-densenet-on-mura-using-pytorch-f39e92566815.

Alyoubi, W. L., Shalash, W. M., & Abulkhair, M. F. (2020). Diabetic retinopathy detection through Deep Learning Techniques: A Review. *Informatics in Medicine Unlocked*, *20*, 100377. https://doi.org/10.1016/j.imu.2020.100377

Clark, K., Luong, M.-T., Manning, C. D., & Le, Q. (2018). Semi-supervised sequence modeling with Cross-View Training. *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. https://doi.org/10.18653/v1/d18-1217

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv [cs.CL]*. Opgehaal van <http://arxiv.org/abs/1810.04805>

El Achi, H. E., & Khoury, J. D. (2020). Artificial Intelligence and digital microscopy applications in diagnostic hematopathology. *Cancers*, *12*(4), 797. https://doi.org/10.3390/cancers12040797

Graham, B., El-Nouby, A., Touvron, H., Stock, P., Joulin, A., Jégou, H., & Douze, M. (2021). LeViT: a Vision Transformer in ConvNet’s Clothing for Faster Inference. *arXiv [cs.CV]*. Opgehaal van http://arxiv.org/abs/2104.01136

Huang, G., Liu, Z., van der Maaten, L., & Weinberger, K. Q. (2018). Densely Connected Convolutional Networks. *arXiv [cs.CV]*. Opgehaal van <http://arxiv.org/abs/1608.06993>

Karim, R. (2021, November 29). *Illustrated: Self-attention*. Medium. Retrieved December 10, 2021, from <https://towardsdatascience.com/illustrated-self-attention-2d627e33b20a>.

Kaselimi, M., Voulodimos, A., Daskalopoulos, I., Doulamis, N., & Doulamis, A. (2021). ForestViT: A Vision Transformer Network for Convolution-free Multi-label Image Classification in Deforestation Analysis. *ICML 2021 Workshop on Tackling Climate Change with Machine Learning*. Opgehaal van <https://www.climatechange.ai/papers/icml2021/40>

Mitra, D. (2020, May 22). *Blindness detection (diabetic retinopathy) using deep learning on eye retina images*. Medium. Retrieved December 10, 2021, from https://towardsdatascience.com/blindness-detection-diabetic-retinopathy-using-deep-learning-on-eye-retina-images-baf20fcf409e.

Pal, L. (2019, September 13). Image classification: A comparison of DNN, CNN and Transfer Learning Approach. Medium. Retrieved December 10, 2021, from <https://medium.com/analytics-vidhya/image-classification-a-comparison-of-dnn-cnn-and-transfer-learning-approach-704535beca25>.

Poostchi, M., Silamut, K., Maude, R. J., Jaeger, S., & Thoma, G. (2018). Image analysis and machine learning for detecting malaria. *Translational Research*, *194*, 36–55. https://doi.org/10.1016/j.trsl.2017.12.004

Saha, S. (2018, December 17). *A comprehensive guide to Convolutional Neural Networks - the eli5 way*. Medium. Retrieved December 10, 2021, from https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53.

Tan, M., & Le, Q. V. (2020). EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. *arXiv [cs.LG]*. Opgehaal van <http://arxiv.org/abs/1905.11946>

Touvron, H., Cord, M., Sablayrolles, A., Synnaeve, G., & Jégou, H. (2021). Going deeper with Image Transformers. *arXiv [cs.CV]*. Opgehaal van <http://arxiv.org/abs/2103.17239>

Tsang, S.-H. (2019, March 20). *Review: DenseNet - dense convolutional network (image classification)*. Medium. Retrieved December 10, 2021, from https://towardsdatascience.com/review-densenet-image-classification-b6631a8ef803.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017, December 6). *Attention is all you need*. arXiv.org. Retrieved December 10, 2021, from https://arxiv.org/abs/1706.03762.

Wang, S., Li, B. Z., Khabsa, M., Fang, H., & Ma, H. (2020). Linformer: Self-Attention with Linear Complexity. *arXiv [cs.LG]*. Opgehaal van <http://arxiv.org/abs/2006.04768>

Wu, B., Xu, C., Dai, X., Wan, A., Zhang, P., Yan, Z., … Vajda, P. (2020). Visual Transformers: Token-based Image Representation and Processing for Computer Vision. *arXiv [cs.CV]*. Opgehaal van <http://arxiv.org/abs/2006.03677>