Cell Growth Media Precipitation Quantification: CS7643 Final Project

Ashish Panchal
Georgia tech
apanchal33@gatech.edu

Amr El Gendy
Georgia tech
aelgendy6@gatech.edu

John Henderson
Georgia tech
jhenderson83@gatech.edu

Abstract

Precipitation in cultivated meat growth media is an important challenge to towards growth of this technology. This paper explores the use of Vision Transformer models for automated quantification of precipitation amounts in a transfer learning setting, covering extensive experimentation to understand the problem domain and how ViT could be easily adapted to novel problems. Our results show high potential for employing vision transformers in acute and imbalanced data classification tasks. We also show existing research does not always align with task hypotheses and model configurations are domain specific. For most parameters we found moderation to known SoA configurations could easily yield great results.

1. Introduction/Background/Motivation

Recent years have seen a tremendous progress towards sustainable development and lifestyle. As a curious race of creatures, we aim to explore the farthest of the places in the universe and create an ever evolving life without constraints! One of the most important factor to walk this road is a stable and sustainable food supply. Cellular agriculture makes this happen, without harming a single animal, growing animal products like meat, dairy and eggs in a controlled environment, which can be adjusted to ones' specific needs, allergies, diet plan ..etc.

A leading biotech company (Client confidentiality precludes the naming of the organization) makes this possible by developing cell growth media. However, during cell culturing in the growth media, they regularly observes formation of precipitation which is believed to impact cell growth and the resulting cell culture product. The company collects images of their cell media experiments, an efficient and low cost way to quantify the precipitation is to develop a software model which can assign a precipitation score to cell media image.

Usually this work would require an extensive exploration of sophisticated algorithms with back and forth validations to create a model which can be used effectively in production, entailing high development and production cost, and an active MLops pipline. Our work explores the capability of the state of the art vision transformer (ViT) models and their ability to transfer learn for faster adaptation [[2]]. ViTs uses a decoder only model, similar to GPT, and learn the 2D structures and patterns framing object and texture in an image.[4],[12],[9] have shed some light on how ViTs are able to learn deeper structures compared to Resnet like CNN models.

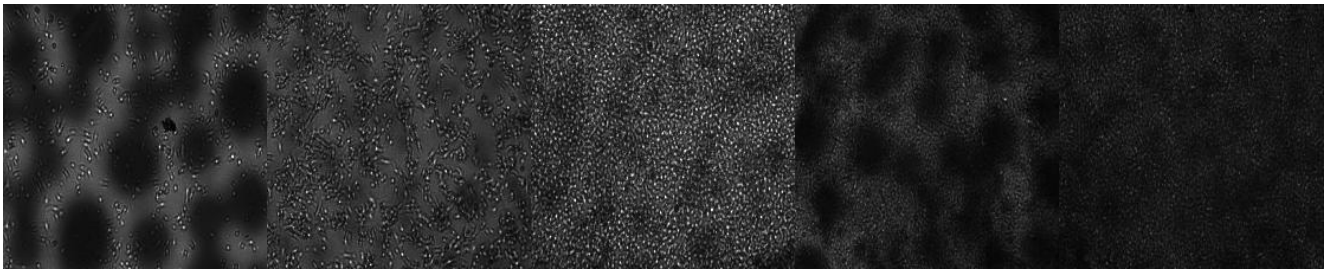
To best of our knowledge there has not been any published work on using ViTs for precipitation identification and classification in cellular agriculture. Our efforts are potentially an initial attempt for the specific application. We feel our models have set an initial benchmark achieving close to 96% validation accuracy. This is crucial of production level tasks, and we hope the application of ViT could be successfully extended to other critical domains.

1.1. Literature Review : Application

Due to lack of work in the domain, we evaluated cross domain and generalized problem statements. The similar nature of precipitation quantification and crowd counting, i.e. quantification of large number of similar objects, set our premise for further exploration. We hypothesized that the texture of precipitate could be localized in a similar manner, by fine-grain feature identifications. [17] [11] [19] [23] in their recent efforts showed how visual transformer models could be used to boost crowd counting. Apart from the model formulation to capture the nature of the problem, some work has been done to validate different loss functions and optimizers for this alignment. However, the majority of the research in crowd counting has been using CNNs [15] [7]. To better understand the nature of the problem, we limited ourselves to use vanilla ViT models.

1.2. Dataset and transformation

A total of 57,137 greyscale images of cell growth experiments, labeled with a precipitation amount on a 5-point scale: 0, 0.25, 0.5, 0.75 and 1, was shared in two sets. The class distribution reflected the natural distribution of precip-



(a) 0.0 (b) 0.25 (c) 0.5 (d) 0.75 (e) 1

Figure 1: Precipitation Class image samples

itation occurance and therefore was heavily imbalanced, 1. Each image showed 1/9th part of a petri dish, with 45% having regions outside the dish as well. In addition to external object regions and imbalanced data, the variance in kind of precipitation pattern posed a bigger problem, which deviated highly from crowd counting, which followed a similar repeated structure for a class object. However, ViT model are found to be good at multitasking [22] and identify richer and deeper features than CNNs [11]. A subset of max 3000 image per set per label was used to balance the training dataset and keeping training times feasible.

Table 1: Counts of dataset images

Set	1	1	1	1	1	2	2	2	2	2
Precip	0.0	0.25	0.5	0.75	1.0	0.0	0.25	0.5	0.75	1.0
# count	38271	2937	1830	1136	867	2142	4158	3654	2016	126

1.2.1 Data transformation

The following data transformations were applied to align with pretraining data:

- Resize to 256
- Center-crop to 224
- Normalization to RGB per channel means of [0.485, 0.456, 0.406]
- Normalization to RGB per channel standard deviation of [0.229, 0.224, 0.225]

2. Approach

This section covers the methodology used for model identification and experiment design for transfer learning.

2.1. Literature review : ViT selection decision

2.1.1 Pre-training dataset

The choice of pre-trained weights for ViT models can have significant implications on transfer learning. According to [16], larger pre-training datasets enhance ViTs' ability to learn intermediate representations. In addition to data size, [16](table 5), outlines the importance of relevance of pre-training dataset to downstream task. With an assumption of texture based identification problem, experimented with nature imagery. [8] outlines texture bias of CNN models with such images, while ensuring non-stylized augmentation training. [13] highlights that high variation in pre-training data may improve the performance in downstream tasks, it is hypothesized, by decreasing overfitting to upstream tasks and learning more general features. One way to ensure this is with large number of classes.

Based on the above criteria 3 pre-training datasets were identified:

- **ImageNet 1k**: The original ImageNet dataset, consisting of 1,000 categories and thousands of images per category. 1k classes and 1,281,16 images.
- **ImageNet 21k**: An extension of the original ImageNet dataset, containing over 21k categories and 14,197,122 images.
- **LAION 2B**: large-scale dataset containing 2.3B images-text pairs. sourced from the internet.

While literature and the nature of the ViT models, would favor the LAION 2B pre-trained weights due to its size and diversity, our experiments on the ViT-b model proved otherwise. By fixing other parameters and varying only the pre-training data, the results shown in Fig 2 were observed. The ImageNet 21K pre-trained weights yielded the best results followed by the ImageNet 1k.

The LAION 2B pre-trained weights, however, did not behave as expected. This can be attributed to several reasons, arguably the most important is "Task Similarity". The LAION 2B dataset, although is bigger in size and has higher diversity, it's not particularly known for being used in image classification tasks. Therefore, the features extracted in its context might not be the most relevant to the task in hand and would then make much more sense to use another dataset that would extract features relevant to this task. For this role, ImageNet 21k fit well in this context.

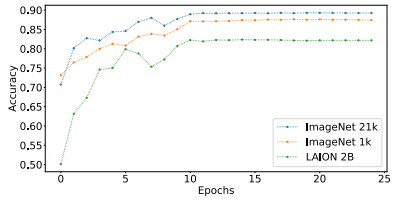


Figure 2: Validation Accuracy for different pre-training Datasets

2.1.2 Model architecture selection

In addition to larger and varied datasets, larger models have been found to show better target space expression and better performance,[20], [5]. Due to limited compute, we limit ourselves to explore only large and base model, and not XLarge and Huge models. Even though the downstream task has only 5 classes, they pose a serious challenge with high intra-class variane, however respecting the class count smaller models were also explored. [20],[18], Literature favours model with high network depth , highly parallelized, however due to the constrained availability of pretrained weights, we limit ourselves to standard model dimensions (D_k, D_v, D_ff, D_model,number of heads).

2.2. Experiment Design and methodology: Fine tuning model choice

The nature of our goal is to maximize the performance of a ViT model at a transfer learning task, by adjusting parameters of the learning process and molding the pretrained feature maps to the downstream task, with minimum changes. Given ViT model are compute heavy and have high training time complexities, we follow a parallel-agile process of building separate hyperparameters to study their effect individually and combining on the way. Furthermore the cost of compute is multiplied over range of experimentation, therefore we only aspire to ascertain the ideological impact of these parameters over a short range, initialized based on research references and tuned to task at hand. The following section covers the experiment and results for Layer Freezing Tuning (MHSA), Loss function, Learning Rate, Batch Size, Patch Size, Decoupled Weight Decay and Learning Rate Scheduler.

2.3. Experimental Setup

Graphics Optimized machine with 1 NVIDIA L4 GPU, 4 vCPUs N2 high memory, 16GB RAM was used for tuning experiments. Along with smaller compute experiments on NVIDIA GTX 1660ti and GTX 1650 super. Each experiment was run for 20-30 epochs, with maximum of 25 hrs per run. Learning curves and Loss curves for different hyper parameters were analyzed after every run to decide next experiment configuration. In addition inference on over all learning curve per class accuracy and over all bext accuracy were made, saving best model weights pickle files and tensorboards.

3. Experiments and Results

This section outlines in detail analysis of different training parameters as stated in 2.2. Cross entropy loss was used as a primary measure of model tuning, given the limited number of classes and associated prediction distributions. Absolute accuracy was used for higher level inference and easier depiction.

3.1. Multi-Head Attention Only Training

In terms of the training approach, using transfer learning is the most logical choice given the data-hungry nature of the vision transformer models. According to [21], one approach that can be used to fine tune the models to new training data is freezing the feed forward layers and only fine-tune Multi head attention layers. This is advantageous as it reduces the time and compute complexity of the training as well as memory consumption. It also allows the reuse of existing FFN weights, which extract important dominant features that typically need a larger dataset to tune.

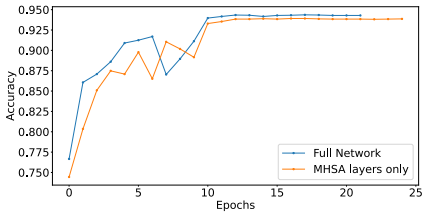


Figure 3: Validation Accuracy with different fine-tuning approaches

To verify this, an experiment was conducted comparing two approaches, Fine-tuning only the multi-head attention layers of a vit-b model with patch 8 and fine-tuning the whole network for the same model. Looking at the results in Fig 3, it's observed that fine-tuning the whole network actually has a slight edge in performance. However, the difference is very subtle, thus, in a practical setting, the merits mentioned above might outweigh the subtle difference in performance. However, given that we are currently interested purely in model performance and accuracy, the full network fine-tuning approach was generally adopted to squeeze out every last bit of performance.

3.2. Loss Function

As precipitation labels are a numerical measurement, an important design decision is whether to formulate the task as a regression problem or a classification problem. Both formulations were experimentally tested. In the classification formulation, a cross-entropy (CE) loss function was applied. In the regression formulation, softmax was applied to normalize the single logit between 0 and 1 and then a Mean Squared Error (MSE) loss function was applied.

Figure 5 shows the results of the trained model on the validation set. The validation accuracy was lower when using MSE loss function than when using the CE loss function. Although in some cases discretization of the output space can lead to a loss of accuracy [10], in the task at hand the reduction of the optimization space may have been useful for the optimization, especially given that the input data labels were themselves in a discrete set.

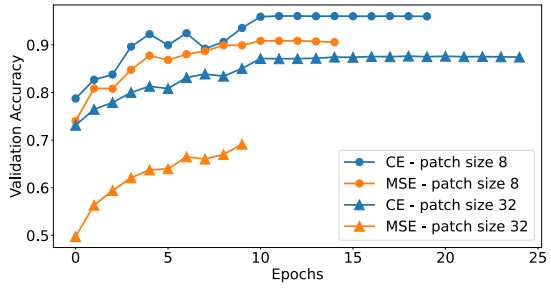


Figure 5: Validation Accuracy with different Loss Functions

3.3. Learning Rate

The learning rate chosen during initial exploration was 0.0001. As it was found to be perform adequately, further experiments were performed to search around this value for superior learning rates. Although this learning rate is lower than values typically used for Vision Transformer finetuning[6], testing significantly higher learning rates was not tried because a lower learning rate is advised when observing training-validation divergence[14].

Note that learning rate specified here was the starting learning rate. For all of the experiments it was reduce by a factor of 0.1 after the tenth epoch.

Figure 4a shows that although decreasing the learning rate from 0.0001 reduced accuracy, increasing it did not notably improve performance.

3.4. Batch Size

One of the parameters considered is the batch size. Batch size can have several effects on the training process and the resulting model performance. In terms of training speed, a lower batch size would slow down the training overall since smaller batches result in more frequent weight updates. On the other hand, working with lower batch sizes has a positive effect on the model's generalizability, which in turn, is reflected on the performance on the validation set, as they encourage the model to learn more robust features.

In this experiment, 4 independent models were trained with 4 different batch sizes, namely 4, 8, 16 and 32 while fixing all other hyper-parameters and using the Vit-b model with patch size of 8 pre-trained on ImageNet 21K. Lack of resources and GPU memory unfortunately were a bottleneck for this experiment which limited the range of experimentation to only a maximum batch size of 32.

Fig 4b shows the results of the trained models on the validation set. It is observed that the batch size of 4 has the highest validation accuracy scoring around 96% validation accuracy. One explanation that might fit within the conditions of the training is that using smaller batch sizes introduce more stochasticity into the training process, which can sometimes help the model escape local minima and find better solutions in addition as well as avoid over-fitting.

3.5. Patch Size

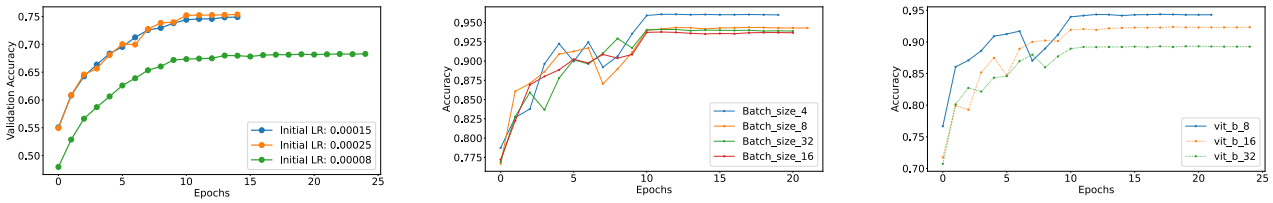
One of the more relevant factors in the ViT models is the patch size chosen. It generally determines the granularity of information that could be captured from the input. A smaller patch size would lead to more information being captured but more patches to process, which would reflect on the computational costs. A larger patch size, on the other hand, would fewer patches and lower computational costs in exchange for potentially losing some detail compared to the lower patches. The choice generally depends on the problem in hand, images with finer textures benefit more from smaller patches while repetitive textures might work well with slightly bigger patches.

To determine the suitable patch size for this problem, an experiment was conducted on 3 versions of the ViT-b model with 3 different patch sizes, small (8), medium (16) and large (32).

Results, shown in Fig 4c, indicate that this problem relies more on fine textures as patch size 8 had the highest performance on the validation set followed by the 16 patch size and the 32 patch size.

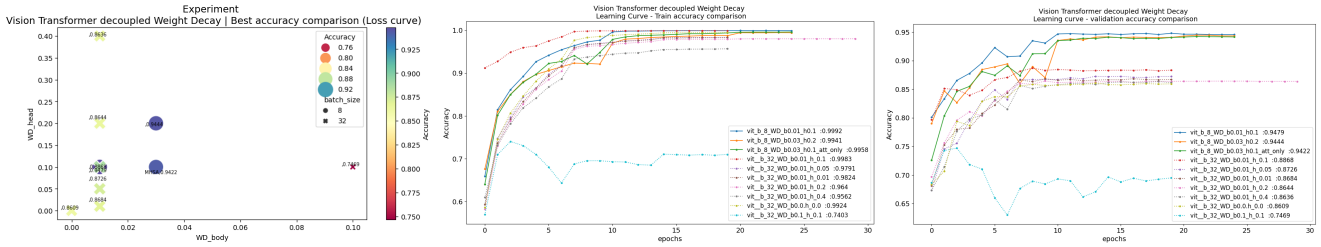
3.6. Decoupled Weight Decay - WD

This section covers the results of decoupled weight decay (WD) experiments. Zhai et al [5] established the benefit of using WD in adaptation in low data regime, during transfer learning. As our data is imbalanced, the effective sample per class is low, therefore, which deems exploration of WD useful. our results support Zhai's work, fig.(6a), i.e higher head H-WD accompanied with low body B-WD results in as better performce. We hypothesize the lower B-WD results in fine changes in the features maps wrt to our data, and a higher H-WD forces the model to find wider margins between classes. We found this to be true for lower (8) as



(a) Validation Accuracy across Learning Rates (b) Validation Accuracy across different batch sizes (c) Validation Accuracy with different Patch sizes

Figure 4: Learning Curve : Experiments - Learning Rate, Batch size & Patch size



(a) Weight decay : Loss curve

(b) Weight decay: Learning Curve Trainingγ vs size

(c) Weight decay: Learning Curve Validation

Figure 6: Decoupled Weight Decay - head vs Body — VIT Patch 8 performance Comparison

batch_size	WD_body	WD_head	att	0	0.25	0.5	0.75	1	avg_per_class	Total_Accuracy
8	0.010000	0.100000		0.964049	0.952066	0.920430	0.936893	1.000000	0.954688	0.947900
	0.030000	0.200000		0.953817	0.948651	0.940810	0.917829	0.974093	0.947040	0.944400
		0.100000	MHSA	0.960194	0.943303	0.919831	0.925865	0.995215	0.948882	0.942200
		0.100000		0.909953	0.869779	0.874608	0.872093	0.982955	0.901878	0.886800
		0.050000		0.892891	0.881245	0.839080	0.843854	0.971591	0.885732	0.872600
	0.010000	0.010000		0.902293	0.870759	0.828304	0.834697	0.979798	0.883170	0.868400
32		0.200000		0.897537	0.860371	0.837838	0.829073	0.951456	0.875255	0.864400
		0.400000		0.898578	0.853399	0.833856	0.840532	0.965909	0.878455	0.863600
	0.000000	0.000000		0.894317	0.857027	0.830385	0.821604	0.984848	0.877636	0.860900
	0.100000	0.100000		0.801896	0.783784	0.634274	0.704319	0.920455	0.768946	0.746900

Figure 7: Decoupled Weight Decay — Per Class accuracy

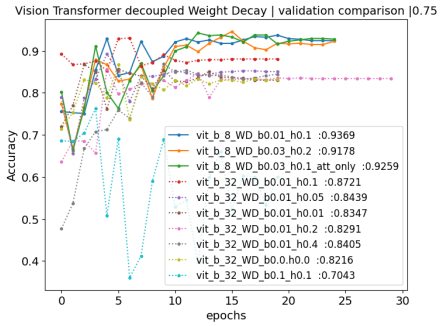


Figure 8: WD — learning curve - Class 0.75

Figure 9: WD — Extended experiment

well as higher (32) batch size, however the effect was limited on latter. The weight decay for our problem, though relatable, but were much lower than [5]'s finding, where we posit the a scheme of lower B-WD and higher H-WD follows the paper's conclusions, but is rations are problem dependent.

Additionally,fig.(6c), we found, when tuned, even a higher batch size of 32 performed comparable to lower batch size initially, (h-wd=0.1,b-wd=0.01) reaching 16.5% better on avg compared to a model without WD in only its 1st epoch, reaching 79.7% validation accuracy. We also found very high WD could result in degenerate behaviour, resulting in catastrophic interference.

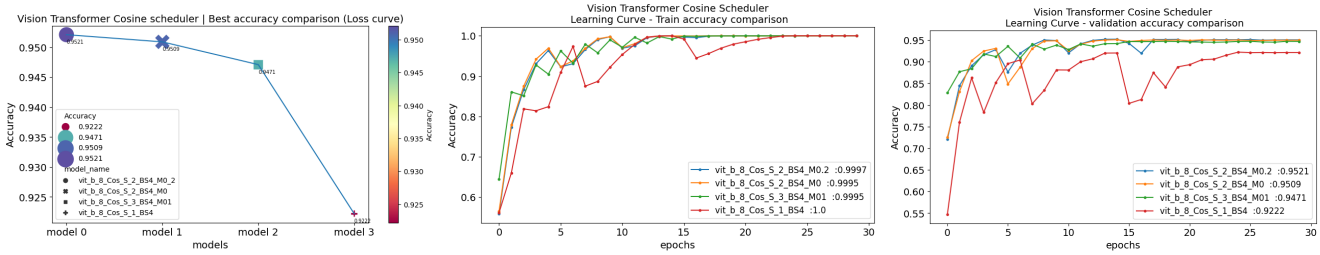
On average, WD only works to improve above the base accuracy, and so lower batch size (8) performed better than 32. Even for these models, fig.(7), different WD configurations forced the model to a different local minima, performing differently for different classes, this can help us adjust a model wrt class importance, for example (HWD 0.2 , BWD 0.01) performed 2% better than best model on class 0.5.

Lastly, this effect is carry forward to different periods during training, as seen in fig (8), Batch 32 (HWD 0.1 , BWD 0.01), performed better than batch 8 in the initial runs.

Through our experiments, we found the model tuned quickly to training data (100% training accuracies), resulting in restricted improvements on validation set. By controlling learning parameters, LR, and WD, and restarting the tuning multiple times, while using a step linear scheduler SLS, we forced the model to generalize well over validation set, and not overfit the training data. We achieved a max of 97% validation accuracy,fig (9), and a convergence accuracy of 96.27%. For fast adaption, in eary stages, we kept WD high and along with larger LR, and as the accuracies plateaued, we forces a fine grain learning, by decreasing WD and LR. The idea of tuning restart, is derived from semulated anealing with restarts, were for each run SLS initiates with a Big LR and gradually decrease it to convergence.

3.7. Learning Rate Scheduler

This section specifically explore the effects of Cosine annealing learning rate scheduler with warm restart. [[5]] in their experiment, empirically proved that for a known number of training epochs, all learning rate schedules performed equally well, and so for major part of our study we used a step linear rate scheduler. However our extended experiment, fig (9), pointed towards potential value in exploring a learning rate which forces the model to explore the hypothesis space better, by coming out of the local minima. Park et al, in their work [[1]] supports this idea where they



(a) Cosine Scheduler : Loss curve (b) Cosine Scheduler : Learning Curve Trainingγ vs size (c) Cosine Scheduler : Learning Curve Validation

Figure 10: Cosine Scheduler — VIT Patch 8 performance Comparison

							0	0.25	0.5	0.75	1	avg_per_class	Total_Accuracy
exp	lr	momentum	T_0	T_mult	multiplier	warmup_iters							
2b	0.000500	0.100000	5	1	0.750000	8000	0.962100	0.957600	0.930800	0.946000	0.985600	0.956420	0.952100
2	0.000500	0.100000	5	1	0.750000	8000	0.957100	0.952300	0.935500	0.948400	1.000000	0.958660	0.950900
3	0.000500	0.100000	2	1	0.900000	2000	0.946100	0.947600	0.929900	0.954600	1.000000	0.955640	0.947100
1	0.002000	0.900000	1	2	0.750000	4000	0.932600	0.896900	0.919400	0.938900	0.984000	0.934360	0.922200

Figure 11: Cosine Scheduler — Per Class accuracy

found a better performance on SGD and AdamW optimizers with cosine scheduler. For simplicity and comparability with our previous experiments, we use SGD over patch 4 VIT base model. Similer to our observations, Loshchilov et al [[3]], propose a cosine annealing scheduler with warm restart CAWR, however the LR for this scheduler starts high. [[5]] emphasize slow warm up in their experiments, and so we design a derivative CAWR with intial warmup steps CAWR_S.

To study the impact of this scheduler in isolation, WD was not used. 3 different configurations over 30 epochs were tested, table (11). where we revaluated the best model. Most of previous experiments used as momentum of 0.9, however as CAWR_S forces model to out of local minima using a schedule and so a high momentum is not needed, this can be seen in the results as well, table (11) and fig (10a). Additionally we found increasing number of iterations for restart T_i using T_multi with a high LR ,0.002, gave the model less opportunities to improve with drastic changes in the current model weights, as seen in fig (10b,10c). In the second experiment, a decreased LR and with a fixed restart interval T_0 of 2 epochs and a high LR multiplier of 90% was used. which resulted in very frequent and non-convergent of learning rate, however substantially improving the accuracy over base experiment. Lastly to stabilize this learning, larger T_0 of 5 epochs along with LR multiplier of 75% was used, giving a 95.1% validation accuracy in 1st run and 95.2% in 2nd.

Using Cosine scheduler, resulted in 2 major findings. 1 using CAWR could potentially improve the pace of learning and generalization, as all but 1 run reached more than 90% validation accuracy in 3 epochs. This could have application adaptability impact, where the state of environment changes rapidly,fig (10b,10c). Second, the difference in the converged local minima has a much stronger emphasis on different class accuracies, further giving a venue for usage in class critical applications. As seen in fig(11) exp 2b,2 and 3 show higher propensities for class (0,0.25),0.5,0.75 respectively. Moveover, even though exp 2b is good overall, exp 2 has a more balanced performance across classes, as shown by avg per class accuracy, and could be useful from application point of view.

4. Discussion, Limitations & Conclusion

In addition to potentially SoA results, our model sheds lights on important aspects on ViT model transfer learning and its relationship with downstream tasks, not aligning with existing research, specifically: 1. larger data is not always better, 2. smaller batch sizes could be better too, 3. smaller patch sizes convert a texture identification task to fine grain object identification and performance better, 4. Decoupled Weight decay works but ratios are task specific, 5. Not all schedulers are equal for a given fixed number of epochs.

Furthermore our experiments guide future implementers when making model choices based on application criticality and correct problem characterization, by loss function, learning rate and scheduler choices. In the future, we aim to explore larger models and study the impact of features on saliency feature maps for our problem.

5. Work Division

Refer to Table 2.

References

[1] Bai, yutong, et al. ”are transformers more robust than cnns?.” advances in neural information processing systems

Student Name	Contributed Aspects	Details
Ashish Panchal	Literature review, Implementation, Analysis and Code review	performed indepth literature review to understand existing work on similar problem domains, along with in-depth literature review for making design choices. Implemented and analyzed Decoupled weight decay along with extended experiment, Learning Rate scheduler and code modularization.
Amr El Gendy	Implementation, Analysis, Machine setup and Code review	Extensive expriments and analysis on weight freeze tuning (MHSA), Batch size, model size and patch size. active code review and brain storming.
John Henderson	Initial ideation, Implementation, Analysis, Data management and Code review	Implementation of learning rate and loss function comparison experiments. Active version control and Repository management.

Table 2: Contributions of team members.

34 (2021): 26831-26843. 5

[2] Dosovitskiy, alexey, et al. "an image is worth 16x16 words: Transformers for image recognition at scale." arxiv preprint arxiv:2010.11929 (2020). 1

[3] Loshchilov, ilya, and frank hutter. "sgdr: Stochastic gradient descent with warm restarts." arxiv preprint arxiv:1608.03983 (2016). 6

[4] Raghu, maithra, et al. "do vision transformers see like convolutional neural networks?." advances in neural information processing systems 34 (2021). 1

[5] Zhai, xiaohua, et al. "scaling vision transformers." proceedings of the ieee/cvf conference on computer vision and pattern recognition. 2022. https://doi.org/10.48550/arxiv.2106.04560. 3, 4, 5, 6

[6] 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, June 2021. arXiv:2010.11929 [cs]. 4

[7] Guangshuai Gao, Junyu Gao, Qingjie Liu, Qi Wang, and Yunhong Wang. Cnn-based density estimation and crowd counting: A survey. *arXiv preprint arXiv:2003.12783*, 2020. 1

[8] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A Wichmann, and Wieland Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. *arXiv preprint arXiv:1811.12231*, 2018. 2

[9] Byeongho Heo, Sangdoo Yun, Dongyoon Han, Sanghyuk Chun, Junsuk Choe, and Seong Joon Oh. Rethinking spatial dimensions of vision transformers. pages 11936–11945, 2021. 1

[10] Stéphane Lathuilière, Pablo Mesejo, Xavier Alameda-Pineda, and Radu Horaud. A Comprehensive Analysis of Deep Regression. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(9):2065–2081, Sept. 2020. arXiv:1803.08450 [cs]. 3

[11] Dingkang Liang, Xiwu Chen, Wei Xu, Yu Zhou, and Xiang Bai. Transcrowd: weakly-supervised crowd counting with transformers. *Science China Information Sciences*, 65(6):160104, 2022. 1, 2

[12] Muhammad Muzammal Naseer, Kanchana Ranasinghe, Salman H Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang. Intriguing properties of vision transformers. *Advances in Neural Information Processing Systems*, 34:23296–23308, 2021. 1

[13] Martin Popel and Ondřej Bojar. Training tips for the transformer model. *arXiv preprint arXiv:1804.00247*, 2018. 2

[14] Martin Popel and Ondřej Bojar. Training Tips for the Transformer Model. *The Prague Bulletin of Mathematical Linguistics*, 110(1):43–70, Apr. 2018. arXiv:1804.00247 [cs]. 4

[15] Pauliina Salmi, Marco Calderini, Salli Pääkkönen, Sami Taipale, and Ilkka Pölönen. Assessment of microalgae species, biomass, and distribution from spectral images using a convolution neural network. *Journal of Applied Phycology*, 34(3):1565–1575, 2022. 1

[16] Andreas Steiner, Alexander Kolesnikov, Xiaohua Zhai, Ross Wightman, Jakob Uszkoreit, and Lucas Beyer. How to train your vit? data, augmentation, and regularization in vision transformers. *arXiv preprint arXiv:2106.10270*, 2021. 2

[17] Guolei Sun, Yun Liu, Thomas Probst, Danda Pani Paudel, Nikola Popovic, and Luc Van Gool. Boosting crowd counting with transformers. *arXiv preprint arXiv:2105.10926*, 3:3, 2021. 1

[18] Yi Tay, Mostafa Dehghani, Jinfeng Rao, William Fedus, Samira Abnar, Hyung Won Chung, Sharan Narang, Dani Yogatama, Ashish Vaswani, and Donald Metzler. Scale efficiently: Insights from pre-training and fine-tuning transformers. *arXiv preprint arXiv:2109.10686*, 2021. 3

[19] Ye Tian, Xiangxiang Chu, and Hongpeng Wang. Cctrans: Simplifying and improving crowd counting with transformer. *arXiv preprint arXiv:2109.14483*, 2021. 1

[20] Hugo Touvron, Matthieu Cord, Alaaeldin El-Nouby, Jakob Verbeek, and Hervé Jégou. Three things everyone should

know about vision transformers. In *European Conference on Computer Vision*, pages 497–515. Springer, 2022. 3

[21] Hugo Touvron, Matthieu Cord, Alaaeldin El-Nouby, Jakob Verbeek, and Hervé Jégou. Three things everyone should know about vision transformers, 2022. 3

[22] Yangyang Xu, Xiangtai Li, Haobo Yuan, Yibo Yang, and Lefei Zhang. Multi-task learning with multi-query transformer for dense prediction. *IEEE Transactions on Circuits and Systems for Video Technology*, 2023. 2

[23] Shaopeng Yang, Weiyu Guo, and Yuheng Ren. Crowdformer: An overlap patching vision transformer for top-down crowd counting. 1:2, 2022. 1