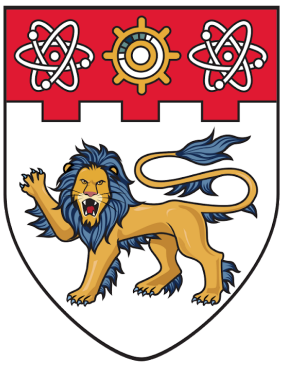

NANYANG TECHNOLOGICAL UNIVERSITY
SINGAPORE

CZ4042 Neural Network & Deep Learning

Final Project
(E) Material Recognition

Member	Matric No.	Email
Kenny Voo Tze Rung	U1921503K	kvoo001@e.ntu.edu.sg
Tan Ching Fhen	U1920787D	Ctan203@e.ntu.edu.sg
Chang Heen Sunn	U1920383H	ch0007nn@e.ntu.edu.sg

1. Objective

To train a convolutional neural network (CNN) to perform Material Recognition on Flickr Material Database (FMD) [1] and Materials in Context Database (MINC) [2].

2. Related Work/ Literature Review

To the best of our knowledge, the state-of-the-art of material recognition is the one in the CVPR 2021 publication titled “Deep Texture Recognition via Exploiting Cross-Layer Statistical Self-Similarity”, a joint research collaboration by South China University of Technology, China and National University of Singapore [3].

The paper presents a novel feature aggregation module called CLASS (Cross-Layer Aggregation of Statistical Self-similarity) for texture recognition. The researchers attempted to integrate CLASS into a ResNet backbone and developed CLASSNet, an effective deep model for texture recognition, which shows state-of-the-art performance in the experiments. The CLASSNet method with ResNet50 as the backbone achieved an mean accuracy of 86.2 and a standard deviation of 0.9.

Over the last few years, there have been a series of breakthroughs in the field of computer vision, better networks such as EfficientNet [4] have been introduced and vision transformers have been gaining popularity. Hence, with due respect to the aforementioned research, we will be using both ResNet50 and EfficientNet in this project.

Specifically, we investigate the effectiveness of “Noisy Student Training” [5][6], an iterative semi-supervised method, on EfficientNet. This method of training trains a teacher model to generate soft pseudo-labels on unlabeled images. Subsequently, a student model is trained on a combination of noisy labeled images and the pseudo labeled images. This process is iterated by treating the student as a teacher.

3. Overview

The general workflow of this project is illustrated as follows:

Figure 1: Project workflow

We will first build the base CNN models with ResNet and EfficientNet as backbone with their corresponding weights for both the FMD and MINC-2500 dataset, and evaluate the models’ performance by studying the mean accuracy along with its standard deviation. Next, we will perform a basic data augmentation on all the three base models built to see if we can improve on the performance of the models.

Subsequently, we will select the best performing model as our benchmark (baseline), to perform hyperparameter tuning where we will be studying the effect of implementing dropouts, tuning the network structure of the neural network, as well as tuning the batch size and learning rate accordingly. The aforementioned steps are not done simultaneously but in a chronological order, that is, once we found the best dropouts value that leads to the largest improvement in the model’s performance, we

will proceed to use that certain dropouts value to the next hyperparameter tuning step. The optimal parameters found in each stage will be brought forward to the subsequent stage until the whole process is complete.

After all the tedious procedures, we will then come to a set of hyperparameters that reached a local maximum of validation accuracy; and we will do the due diligence to rerun the baseline model with the optimal configuration settings to check if we can replicate the experiment results. Lastly, we will propose other approaches that could further improve the model's performance in future's study.

4. Data Exploration

In this project, we will be using the given Flickr Material Database (FMD) and a much larger dataset, Materials in Context Database (MINC) - 2500. Some information about the two datasets are shown in the table below . Note that there is 57.5 times difference between the size of the two datasets.

Dataset	Number of observations	Number of material category	Frequency of each category	Materials category
FMD	1000	10	100	fabric, foliage, glass, leather, metal, paper, plastic, stone, water, and wood.
MINC-2500	57500	23	2500	brick, carpet, ceramic, fabric, foliage, food, glass, hair, leather, metal, mirror, other, painted, paper, plastic, pol.stone, skin, sky, stone, tile, wallpaper, water, wood

5. Data Preprocessing

For all the experiments, the images were resized to 256x256 and it will be randomly cropped to 224x224 as the inputs to the backbone. Standard preprocessing is applied to the images based on the backbones. For Resnet, the images will be converted from RGB to BGR, then will zero-center each color channel with respect to the ImageNet dataset, without scaling. While for EfficientNet, image will be rescale in the range of [0,1] and shift it into a distribution centered around 0 with standard deviation 1.

6. Building the Base Model

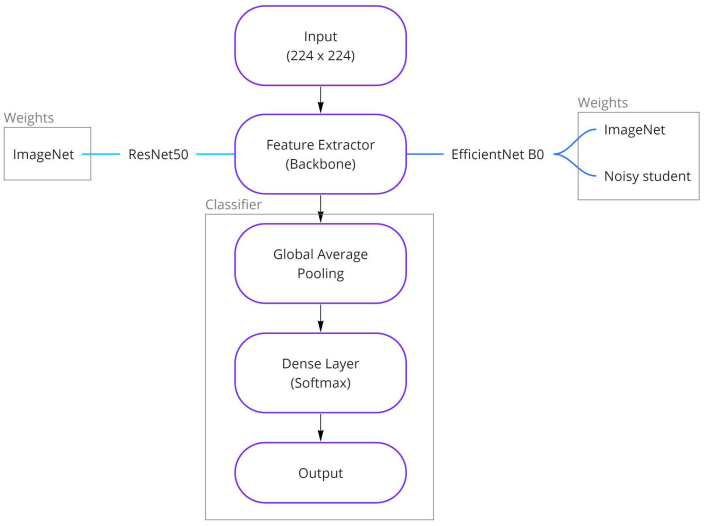

Figure 2: Building Base model

Training CNN models requires an extensive amount of time. For starters, we attempt to first come out with a simple and executable model for both the FMD and MINC-2500 dataset. Instead of training from scratch, we have performed transfer learning in these experiments. The last layer of the pretrained model is removed and connected to a classifier as shown in Figure 2. The pretrained model is freezed before fine tuning. Since MINC-2500 comes with 5 preset train-test split data; to ensure fair comparison, we will conduct 5-fold cross validation on the FMD dataset as well. We will take the mean validation categorical accuracies and the corresponding standard deviation as the metrics to evaluate the model performance.

As aforementioned, the configuration for our baseline models are as follows:

- Batch_size = 32 (FMD) , 256(MINC-2500)
- Learning_rate = 1e-03
- Epoch = 50
- Optimizer = Adam
- Callback = early-stopping -10 steps (MINC-2500 dataset); None for FMD

The results are as follows:

Model Backbone (Freezed)	Dataset			
	FMD		MINC-2500	
	Mean accuracy	Standard deviation	Mean accuracy	Standard deviation
Resnet 50 (imagenet)	0.841	0.0275	0.6682	0.0038
EfficientNet B0 (imagenet)	0.826	0.0213	0.6398	0.0057
EfficientNet B0 (noisy student)	0.869	0.0124	0.6495	0.0032

Figure 3: Performance of base models

To our surprise, the best performing model for FMD, the EfficientNet B0 with noisy student pretrained weight has already achieved an validation accuracy as high as 86.9 with standard deviation 0.01, which is a huge improvement from the results of the state-of-the-art model stated in the research paper at the beginning (86.2, std=0.9). However, Resnet50 with pre-trained imagenet weights performed the best among the rest for MINC-2500.

However, note that the current validation accuracy obtained is subject to overfitting. We can clearly see the problem by observing the accuracy vs epoch graphs as placed in the Appendix below. We shall try to reduce the problem of overfitting for the rest of the project moving forward.

7. Data Augmentation

In this section, we will be performing basic data augmentation to all three base models we built from the previous sessions. Some techniques we used for augmentation include random flip, random translation, random contrast, random rotation and random crop.

The workflow of this section is similar to that when building the base model, except that we incorporated the step of data augmentation this time, as illustrated below:

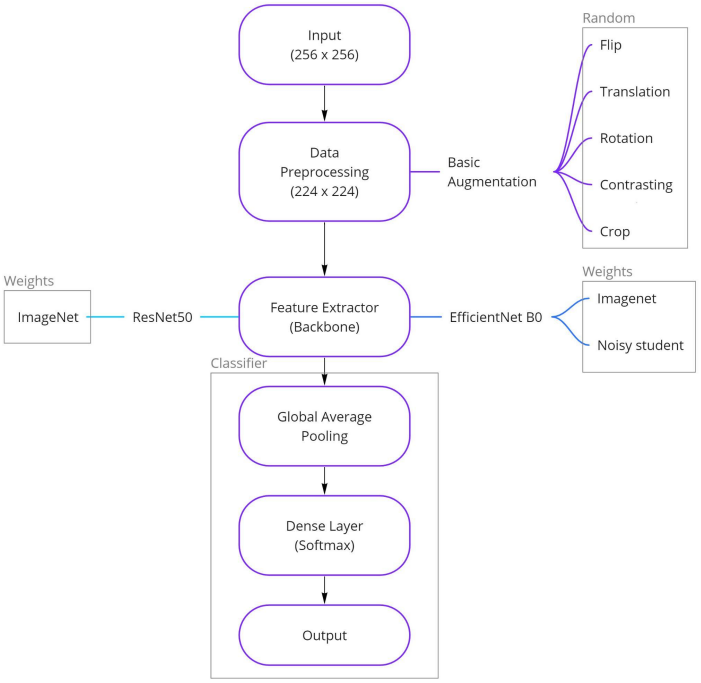

Figure 4: Performing Data Augmentation

We obtain the following result after performing the basic augmentation.

	Dataset			
Model Backbone (Freezed)	FMD		MINC-2500	
	Mean accuracy	Standard deviation	Mean accuracy	Standard deviation
Resnet 50 (imagenet)	0.834	0.0246	0.6328	0.0077
EfficientNet B0 (imagenet)	0.831	0.0213	0.5928	0.0048
EfficientNet B0 (noisy student)	0.863	0.004	0.5962	0.0034

Figure 5: Performance of models after augmentation

In overall, all the models are less overfitting as shown in the graphs at Appendix. The accuracy on the FMD dataset improved after performing basic augmentation. However, it's not the same case for the MINC-2500 dataset. The reason for this might be due to the early stopping. Based on the graphs shown in Appendix, the accuracy of EfficientNet (s) might improve with longer epochs.

8. Hyperparameter tuning

We will be tuning the EfficientNet B0 (noisy student) for FMD only based on the following parameters:

- Dropouts; [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8]
- Hidden layers
- Number of neurons in dense layers; [32, 64, 128, 256, 512]
- Batch size; [8, 16, 32, 64, 128]
- Learning rate; [1e-01, 1e-02, 1e-03, 1e-04, 1e-05]

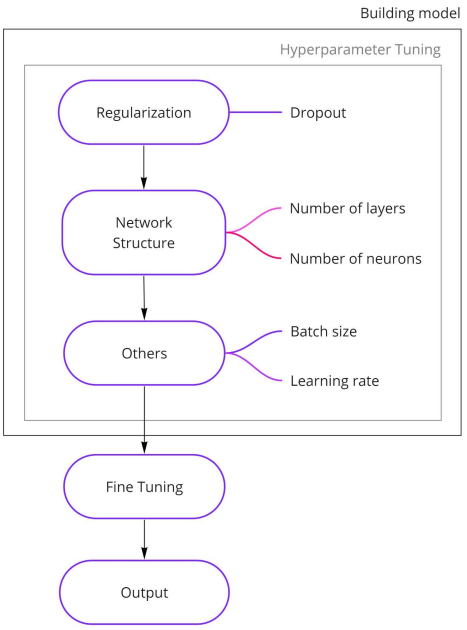

Figure 6: Hyperparameter tuning on EfficientNet B0 (noisy student)

EfficientNet B0 (Noisy student)	FMD	
Dropout	Mean accuracy	Standard deviation
None	0.863	0.004
0.2	0.873	0.0075
0.3	0.87	0.0148
0.4	0.873	0.0075
0.5	0.872	0.0117
0.6	0.862	0.0093
0.7	0.867	0.0093
0.8	0.869	0.0116

Figure 7: Regularization - dropouts

		FMD	
Model	Neurons	Mean accuracy	Standard deviation
Baseline*	None	0.873	0.0075
Additional one dense layer	32	0.869	0.0086
	64	0.872	0.0121
	128	0.87	0.0071
	256	0.869	0.0136
	512	0.869	0.0174

Figure 8: Changing network structure

EfficientNet B0 (Noisy student)	FMD	
Batch size	Mean accuracy	Standard deviation
4	0.866	0.0037
8	0.871	0.0066
16	0.87	0.0105
32	0.868	0.0068
64	0.866	0.0086
128	0.856	0.0066

Figure 9: Tuning batch size

EfficientNet B0 (Noisy student)	FMD	
Learning rate	Mean accuracy	Standard deviation
1e-01	0.852	0.0152
1e-02	0.858	0.0169
1e-03	0.871	0.0102
1e-04	0.828	0.0242
1e-05	0.527	0.0586

Figure 10: Tuning learning rate

The chosen optimal hyperparameter is highlighted in green as shown in the tables above. In general, we choose the optimal hyperparameter based on high validation accuracy, low standard deviation, and observing the accuracy vs epoch plots (to ensure overfitting is minimized) as attached in the

Appendix below. The tuning for learning rate is a special case; in the end we chose the value of 1e-04 despite a lower accuracy given its huge potential to upside (apparently 50 epochs are not enough for it to be trained fully).

9. Fine Tuning

Below are the optimal set of parameters for the best performing EfficientNet B0 with noisy student weights model:

- Dropouts = 0.4,
- No additional layers are added; using the same architecture of the baseline model,
- Batch_size = 32,
- Learning_rate = 1e-04,

We then perform the due diligence to rerun the model with the optimal configuration settings listed above and check if we can replicate the results of the experiments. This time since we are conducting the final assessment on the model, we will split the training and testing dataset into a ratio of 80:20, instead of using the 5-fold cross validation. We implemented the early stopping (es = 15) for the training process and removed the upper limit of epoch (was 50) in order to seek the best possible validation accuracy.

Once the model training converges or stops improving, fine tuning will be performed. Part of the efficientNet B0 model (last 16 layers) will be unfreezed and a lower learning rate (1e-5) will be set and the training will be continued until the validation accuracy no longer improves.

We obtained the following:

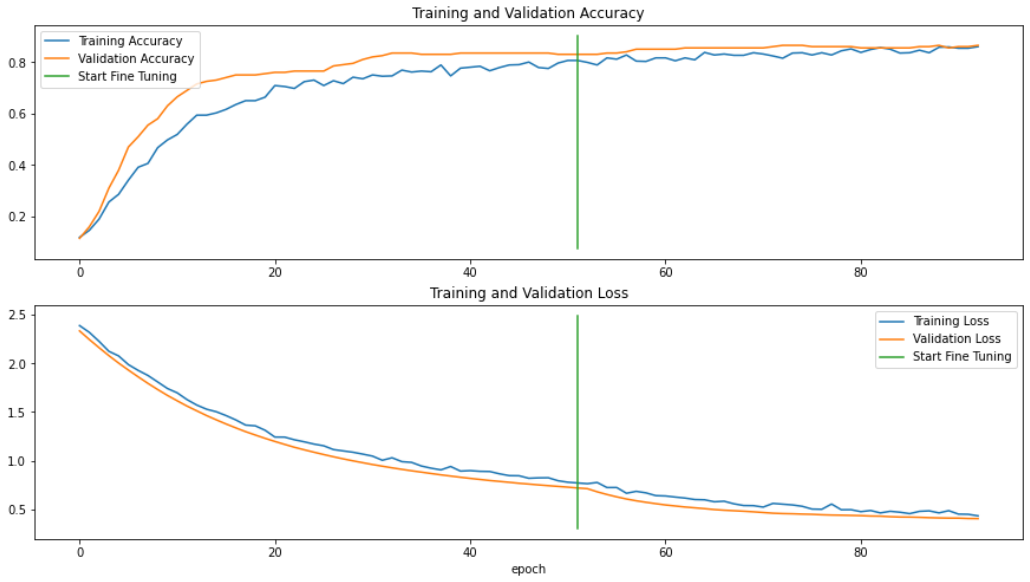

Figure 11: Performance of fine-tuned models, on FMD

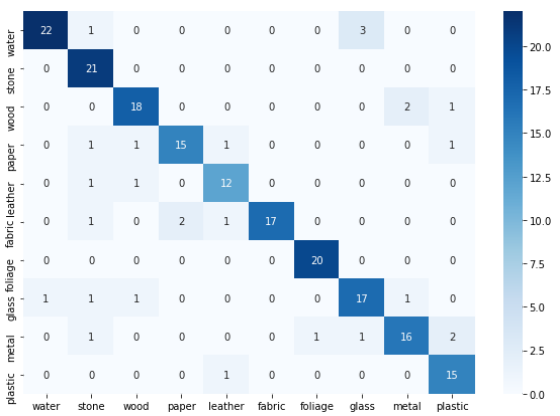

Figure 12: Confusion Matrix, FMD

ROC_AUC score : 0.99

	precision	recall	f1-score	support
water	0.96	0.85	0.90	26
stone	0.78	1.00	0.88	21
wood	0.86	0.86	0.86	21
paper	0.88	0.79	0.83	19
leather	0.80	0.86	0.83	14
fabric	1.00	0.81	0.89	21
foliage	0.95	1.00	0.98	20
glass	0.81	0.81	0.81	21
metal	0.84	0.76	0.80	21
plastic	0.79	0.94	0.86	16
accuracy			0.86	200
macro avg	0.87	0.87	0.86	200
weighted avg	0.87	0.86	0.86	200

Figure 13: Classification report, FMD

On the other hand, although ResNet50 performed the best for MINC-2500, we will still perform fine tuning on the EfficientNet B0 with noisy student weight as studying the effect of the newer backbone is our objective. We unfreezed 16 layers on the EfficientNet B0 model for the MINC-2500 dataset. Some possible ways to improve the model is to train the model from scratch, remove early stopping, use a larger network (EfficientNet B1,B2) and also unfreeze more layers during fine tuning.

We obtained the following:

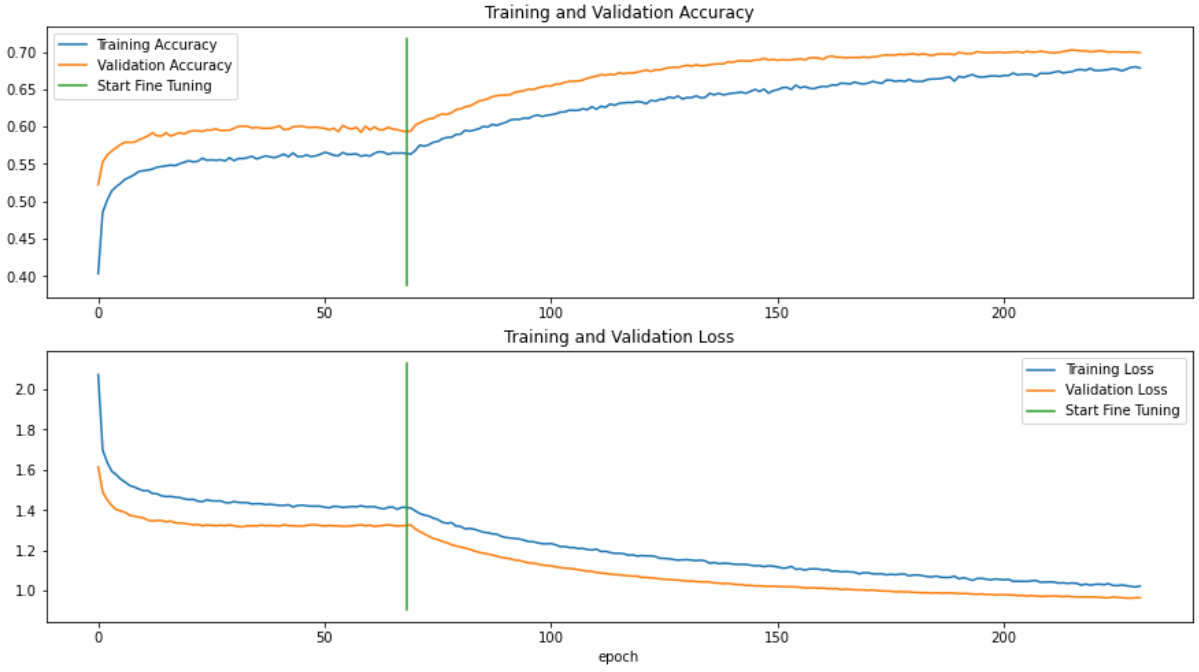

Figure 14: Performance of fine-tuned models, on MINC-2500

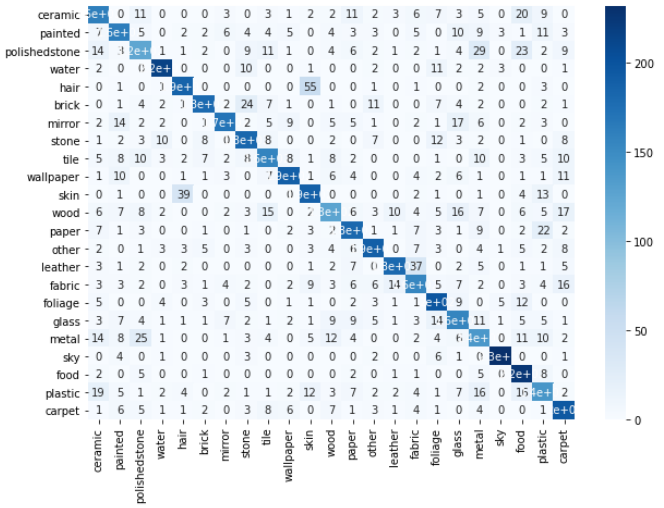

Figure 15: Confusion matrix, FMD

ROC_AUC score : 0.98

	precision	recall	f1-score	support
ceramic	0.63	0.65	0.64	250
painted	0.67	0.65	0.66	250
polishedstone	0.58	0.50	0.54	250
water	0.86	0.86	0.86	250
hair	0.76	0.74	0.75	250
brick	0.84	0.72	0.77	250
mirror	0.84	0.68	0.75	250
stone	0.69	0.73	0.71	250
tile	0.68	0.63	0.65	250
wallpaper	0.83	0.76	0.79	250
skin	0.66	0.76	0.71	250
wood	0.63	0.50	0.56	250
paper	0.69	0.73	0.71	250
other	0.78	0.76	0.77	250
leather	0.84	0.72	0.78	250
fabric	0.63	0.62	0.62	250
foliage	0.70	0.79	0.74	250
glass	0.61	0.62	0.62	250
metal	0.51	0.55	0.53	250
sky	0.95	0.93	0.94	250
food	0.66	0.90	0.76	250
plastic	0.57	0.56	0.57	250
carpet	0.67	0.78	0.72	250
accuracy			0.70	5750
macro avg	0.71	0.70	0.70	5750
weighted avg	0.71	0.70	0.70	5750

Figure 16: Classification report, MINC-2500

10. Future Improvement

In this section, we research and experiment two recent data augmentation techniques: CutMix [7] and AugMix [8], in an attempt to mitigate overfitting and improve model robustness and performance.

The objective of CutMix is to generate a novel training sample (x˜, y˜) by combining two original training samples (xA, yA) and (xB, yB). The new training sample (x˜, y˜) is used to train the model with its original loss function. $M \in \{0, 1\}^{WxH}$ is the binary mask that determines where to substitute the second image (Figure 1). λ is the combination ratio between the two original images, sampled from a beta distribution with parameter “alpha”(Figure 1).

$$\tilde{x} = \mathbf{M} \odot x_A + (\mathbf{1} - \mathbf{M}) \odot x_B$$
$$\tilde{y} = \lambda y_A + (1 - \lambda) y_B,$$

Figure 17: CutMix data augmentation operations

On the other hand, AugMix generates novel images by mixing multiple data augmentation chains with the original image - with combination ratio, m, sampled from a beta distribution (Figure 2). Additionally, the original authors coupled this scheme with Jensen-Shannon Divergence Consistency Loss to enforce smoother neural network responses - though we do not implement this loss in our project.

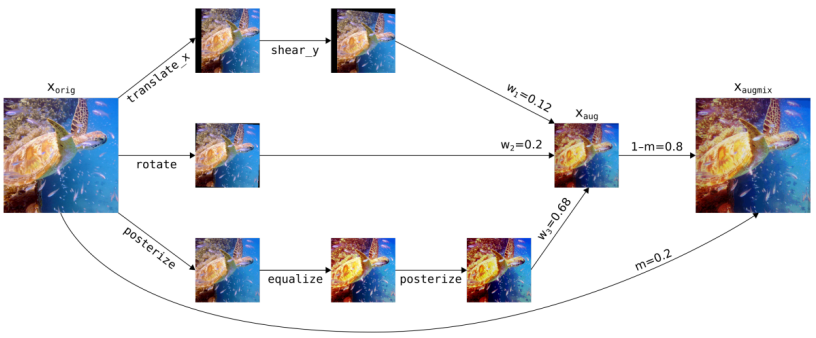

Figure 18: A realization of AugMix.

We investigate the implementation of CutMix at varying “alpha” parameters and AugMix at varying augmentation severities and number of augmentation chains (width). Despite our best efforts, both augmentation techniques do not improve on mean 5-fold cross validation performance (Figure 17&18) over the baseline, with no data augmentation. However, it is observed that both methods are effective at mitigating overfitting; validation accuracies (dotted lines) exceed training accuracies (bold lines) in all cases.

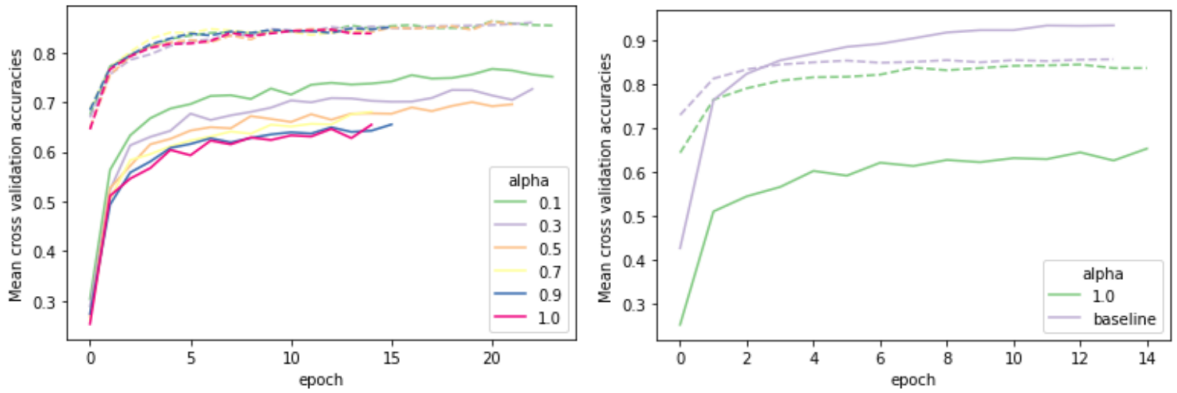

Figure 19: CutMix mean cross validation accuracies (dotted lines) over varied “alpha” parameters

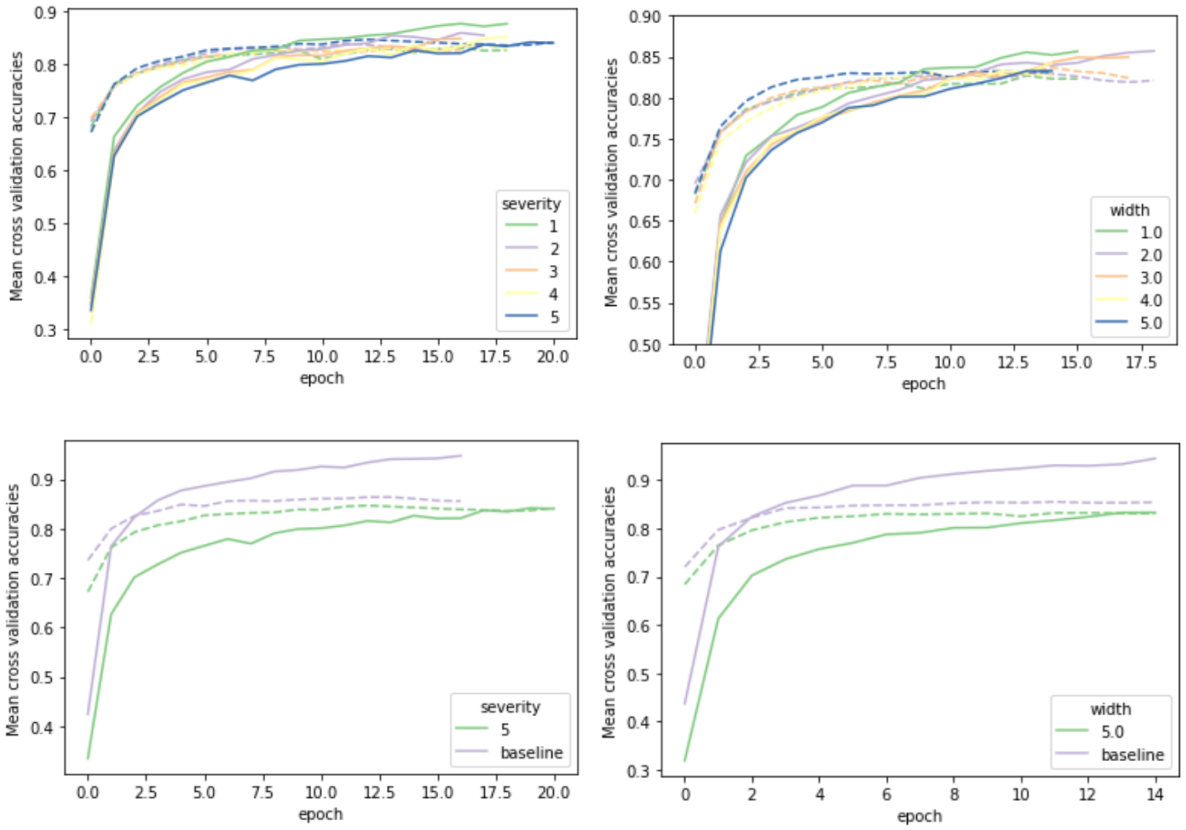

Figure 20: AugMix mean cross validation accuracies (dotted lines) over varied “severity” and “width” parameters.

In our future work, we could also implement Jensen-Shannon Divergence Consistency Loss of AugMix [9], which we had left out. Furthermore, we could further investigate the use of vision transformers- a transformer that is targeted at vision processing tasks such as image recognition.

11. Conclusion

In summary, given the constraint in real life (computational power, deadlines, etc.), it does not make any economic sense for us to brute force search all the permutations of hyperparameters and look for the global optimal set of configuration that leads to the highest possible validation accuracy for the CNN model. Keep in mind that there will be randomness and uncertainty associated during the training process of the model.

All in all, we have successfully achieved a final accuracy of 86% for FMD and 70% for MINC-2500. The experiments showed that a simple classifier head with an up to date backbone (EfficientNet) can perform as well as the CLASSNet [3] on a simpler dataset (FMD) but failed to perform as well on a more complex dataset. This experiment proved the need of these feature aggregation modules in tackling material recognition.

12. References

[1] Liu, L. Sharan, E. H. Adelson, and R. Rosenholtz, “Exploring features in a Bayesian framework for material recognition,” in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2010

[2] S. Bell, P. Upchurch, N. Snavely, K. Bala, “Material recognition in the wild with the materials in context database,” in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015

[3] Chen, Zhile, et al. "Deep Texture Recognition via Exploiting Cross-Layer Statistical Self-Similarity." *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021.

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

[5] Mingxing Tan and Quoc V. Le. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. arXiv:1905.11946, 2019.

[6] Qizhe Xie, Minh-Thang Luong, Eduard Hovy and Quoc V. Le. Self-training with Noisy Student improves ImageNet classification. arXiv:1911.04252, 2019.

[7] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe and Youngjoon Yoo. CutMix: Regularization Strategy to Train Strong Classifiers with Localizable Features. arXiv:1905.04899, 2019.

[8] Dan Hendrycks, Norman Mu, Ekin D. Cubuk, Barret Zoph, Justin Gilmer and Balaji Lakshminarayanan. AugMix: A Simple Data Processing Method to Improve Robustness and Uncertainty. arXiv:1912.02781, 2019.

[9] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, Neil Houlsby. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. arXiv:2010.11929, 2020.

13. Appendix

Building the base model:

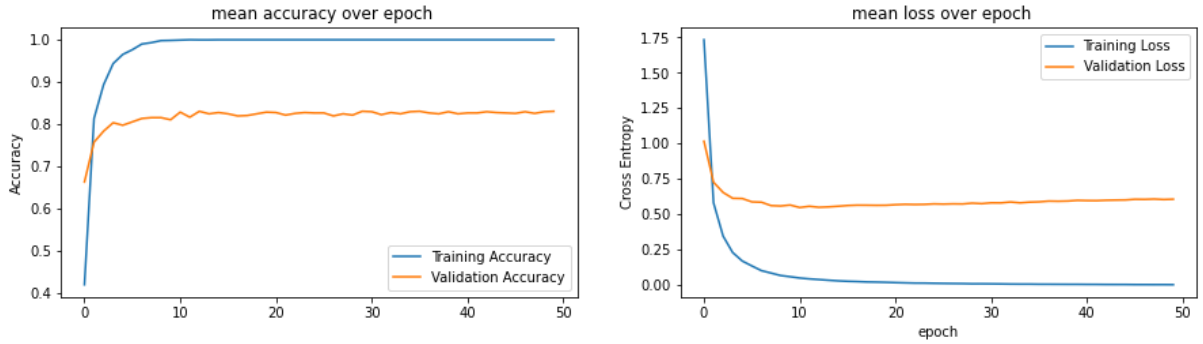

Figure: ResNet50 (Imagenet), FMD

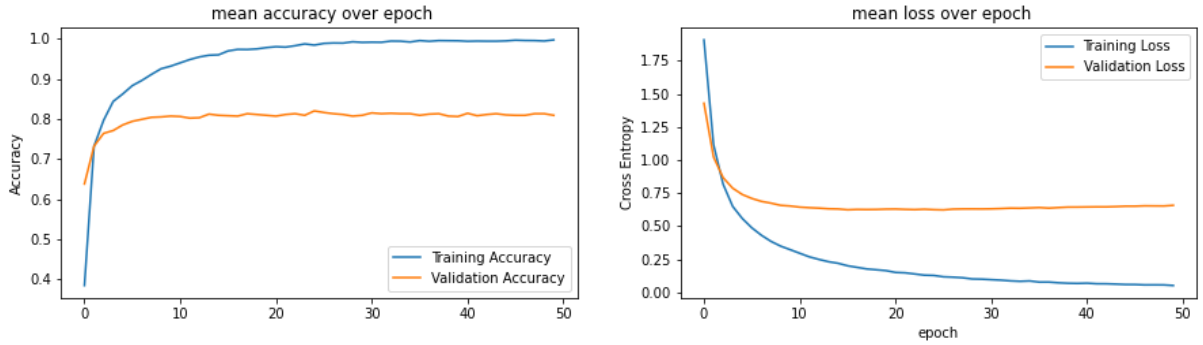

Figure: EfficientNet B0 (Imagenet), FMD

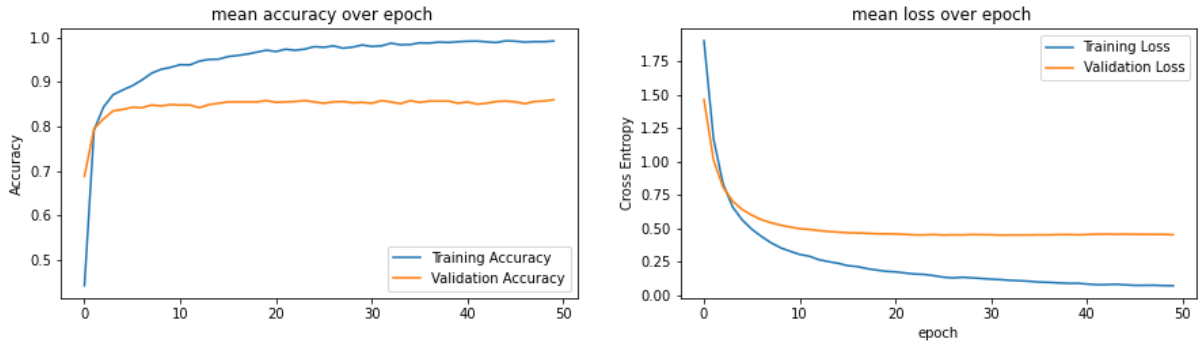

Figure: EfficientNet B0 (noisy student), FMD

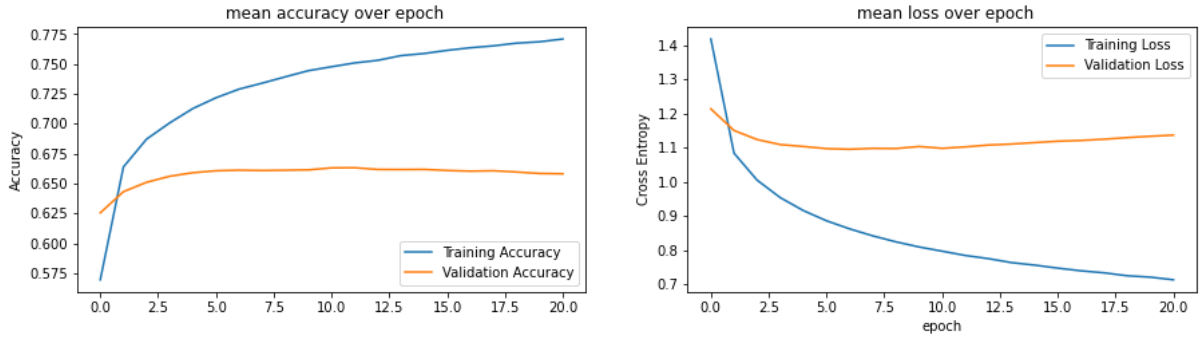

Figure: ResNet50 (Imagenet), MINC-2500

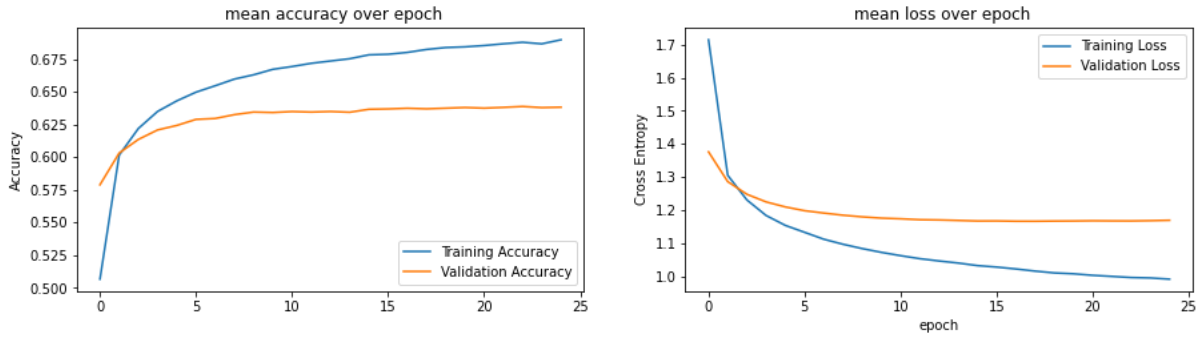

Figure: EfficientNet B0 (Imagenet), MINC-2500

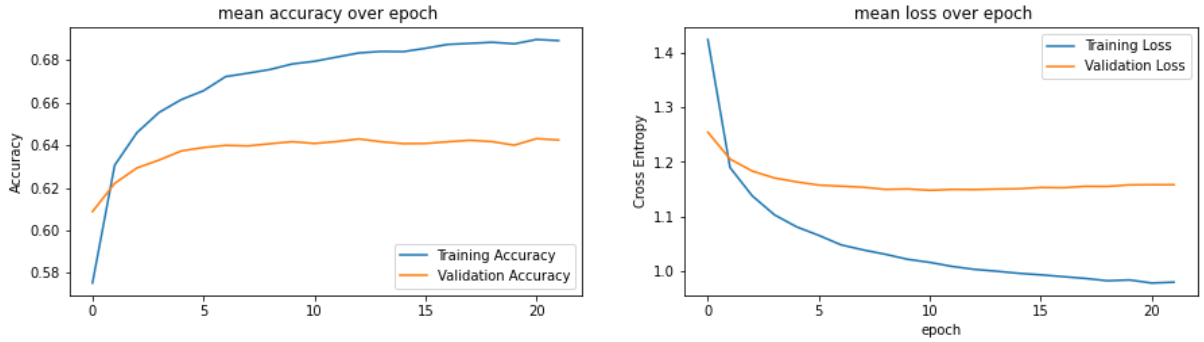

Figure: EfficientNet B0 (noisy student), MINC-2500

Data Augmentation:

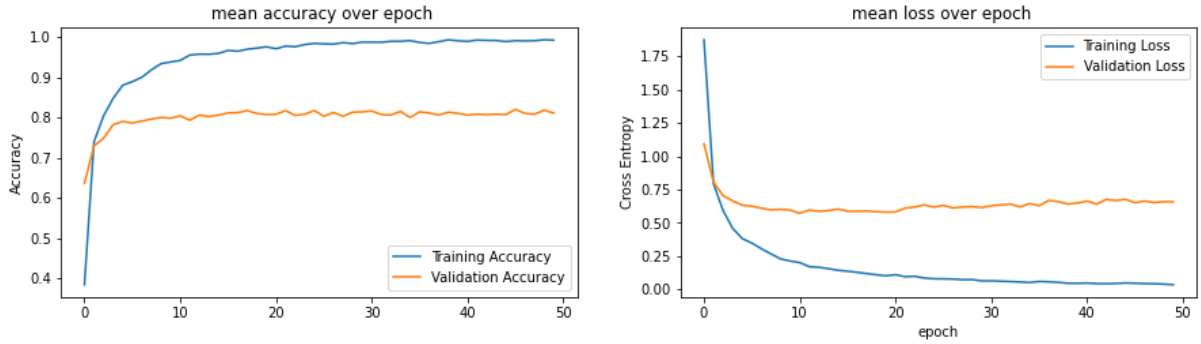

Figure: ResNet50 (Imagenet), FMD

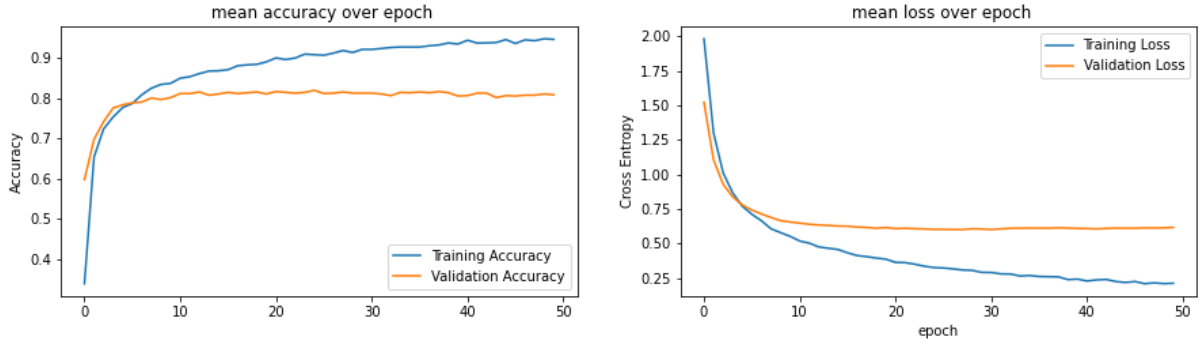

Figure: EfficientNet B0 (Imagenet), FMD

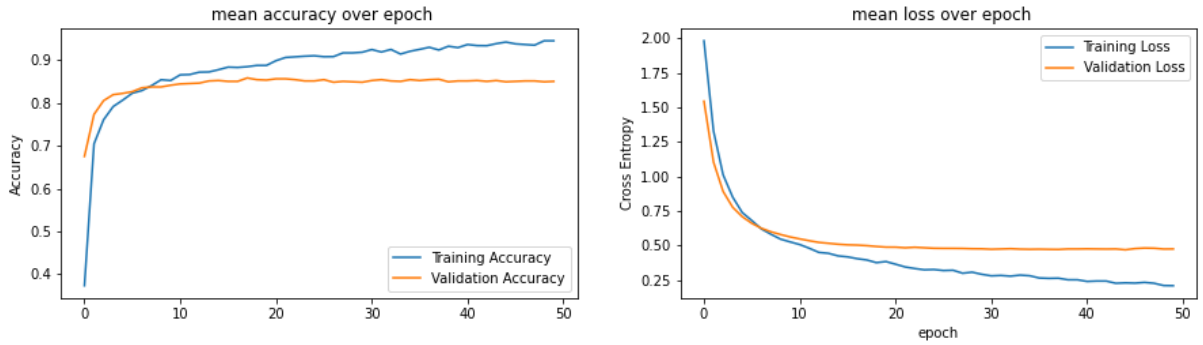

Figure: EfficientNet B0 (noisy student), FMD

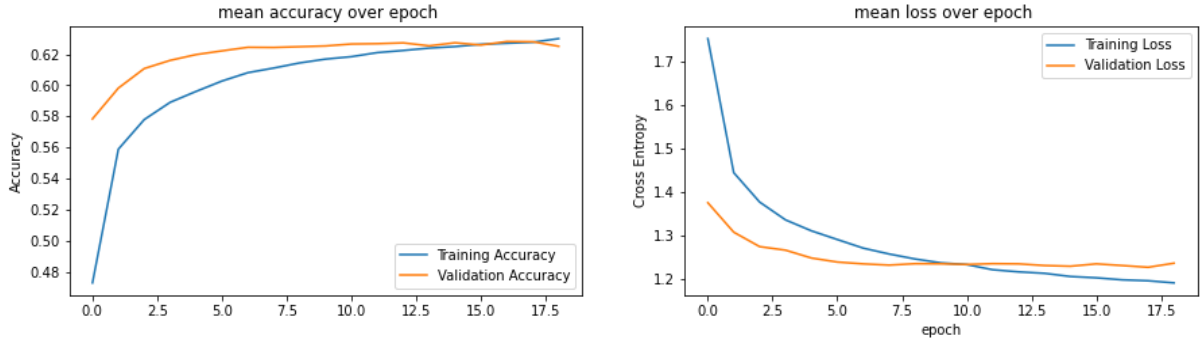

Figure: ResNet50 (Imagenet), MINC-2500

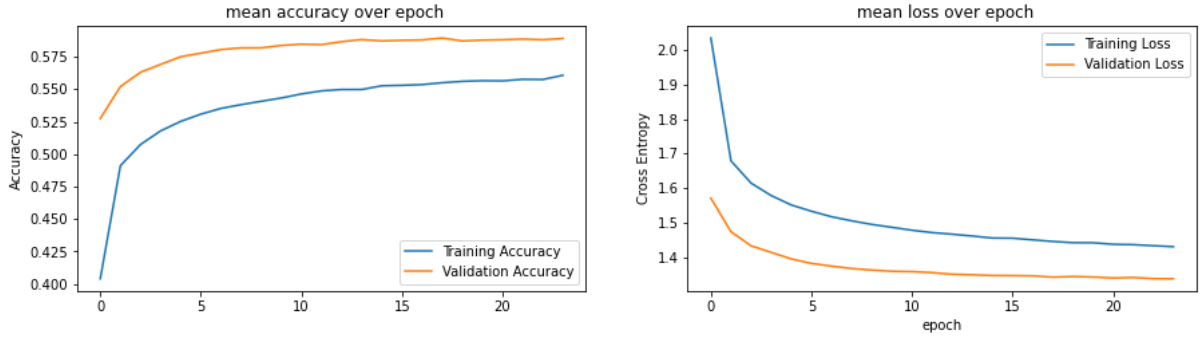

Figure: EfficientNet B0 (Imagenet), MINC-2500

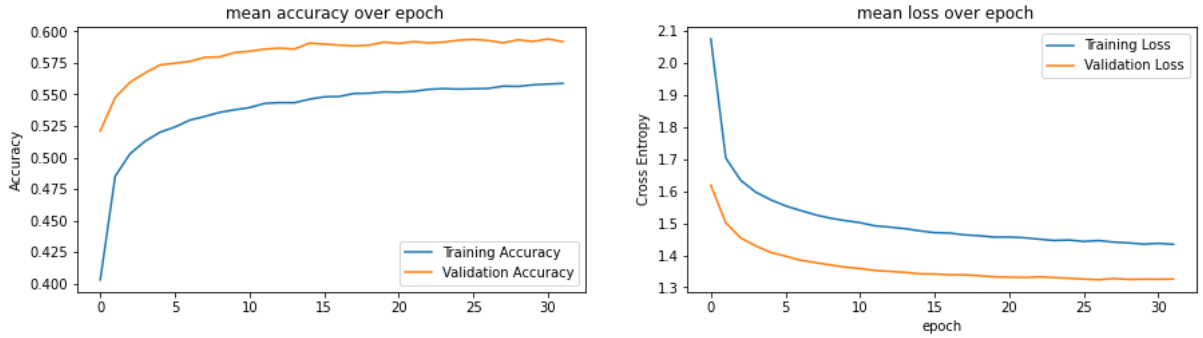

Figure: EfficientNet B0 (noisy student), MINC-2500

Hyperparameter Tuning:

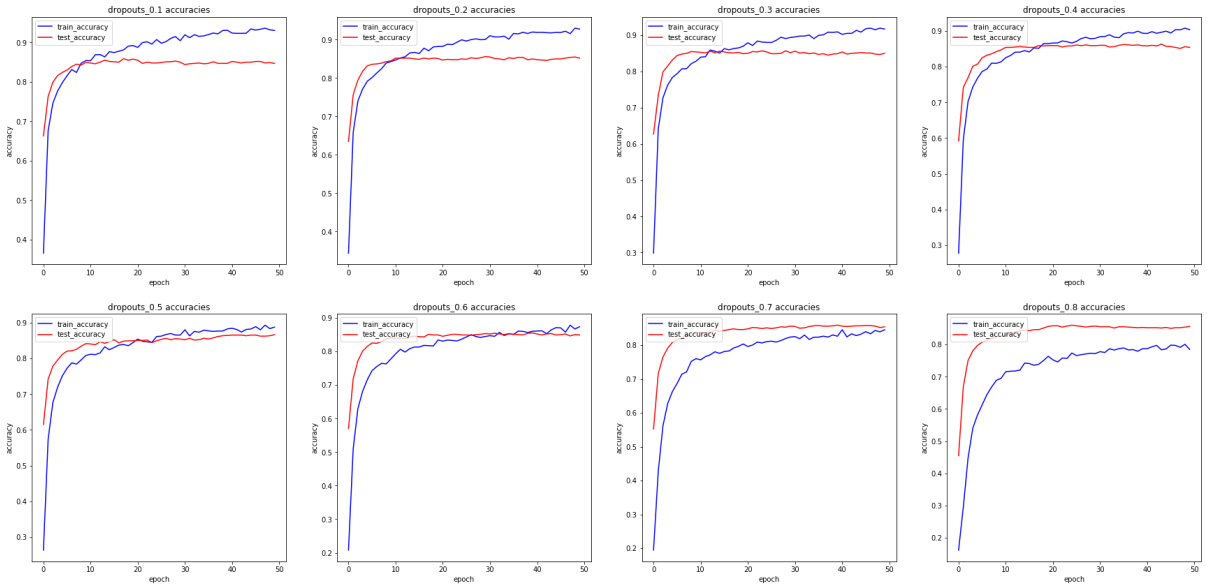

Figure: EfficientNet B0 (Noisy student), FMD, dropouts

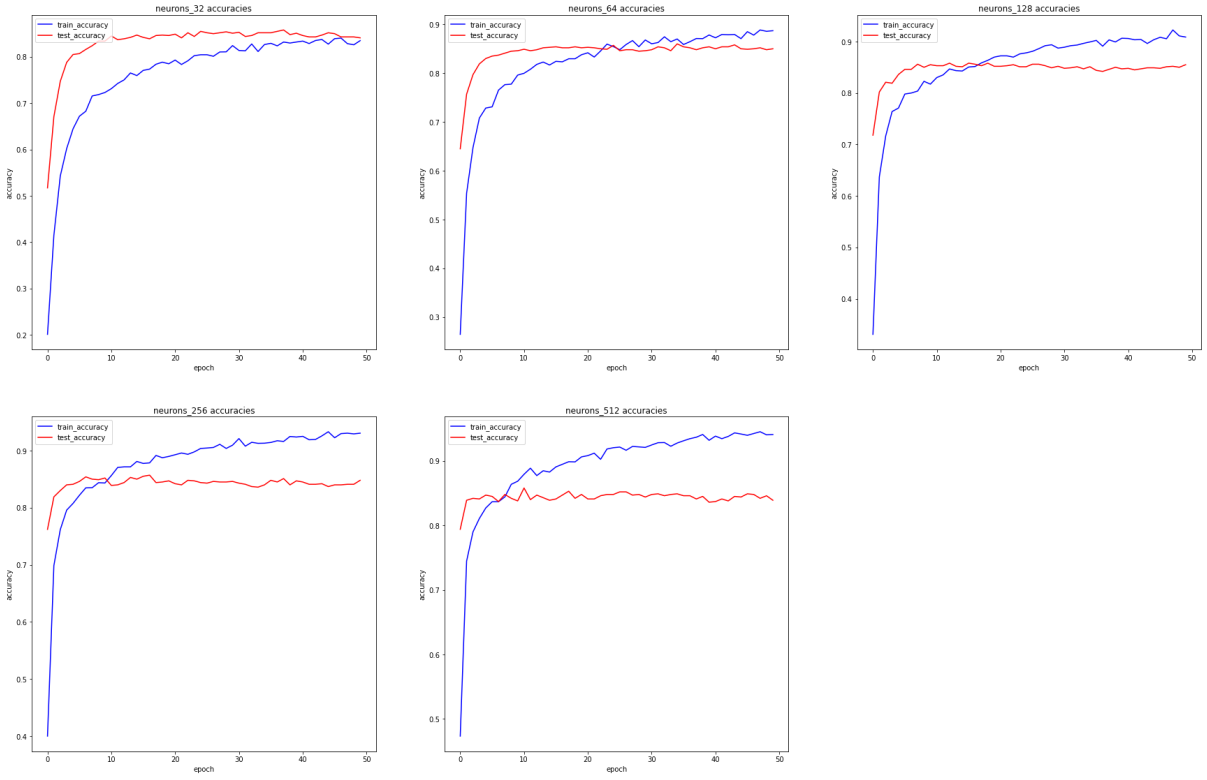

Figure: EfficientNet B0 (Noisy student), FMD, neurons

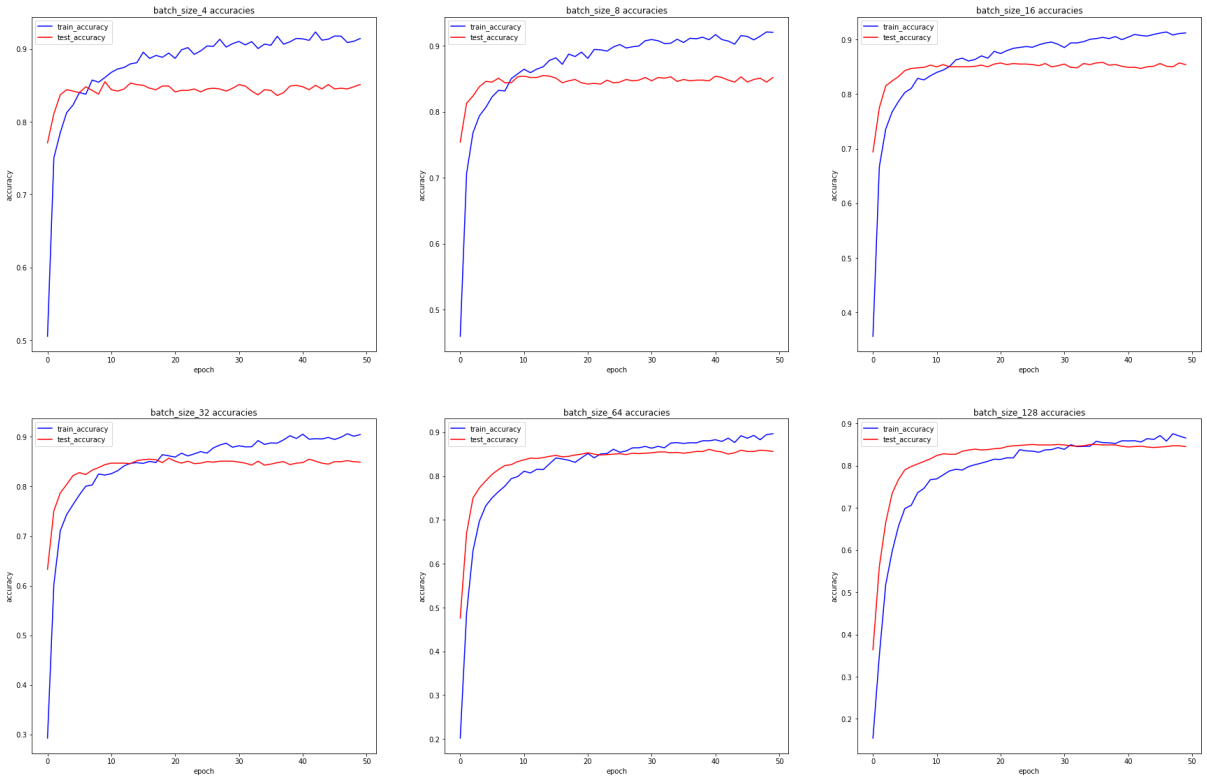

Figure: EfficientNet B0 (Noisy student), FMD, batch_size

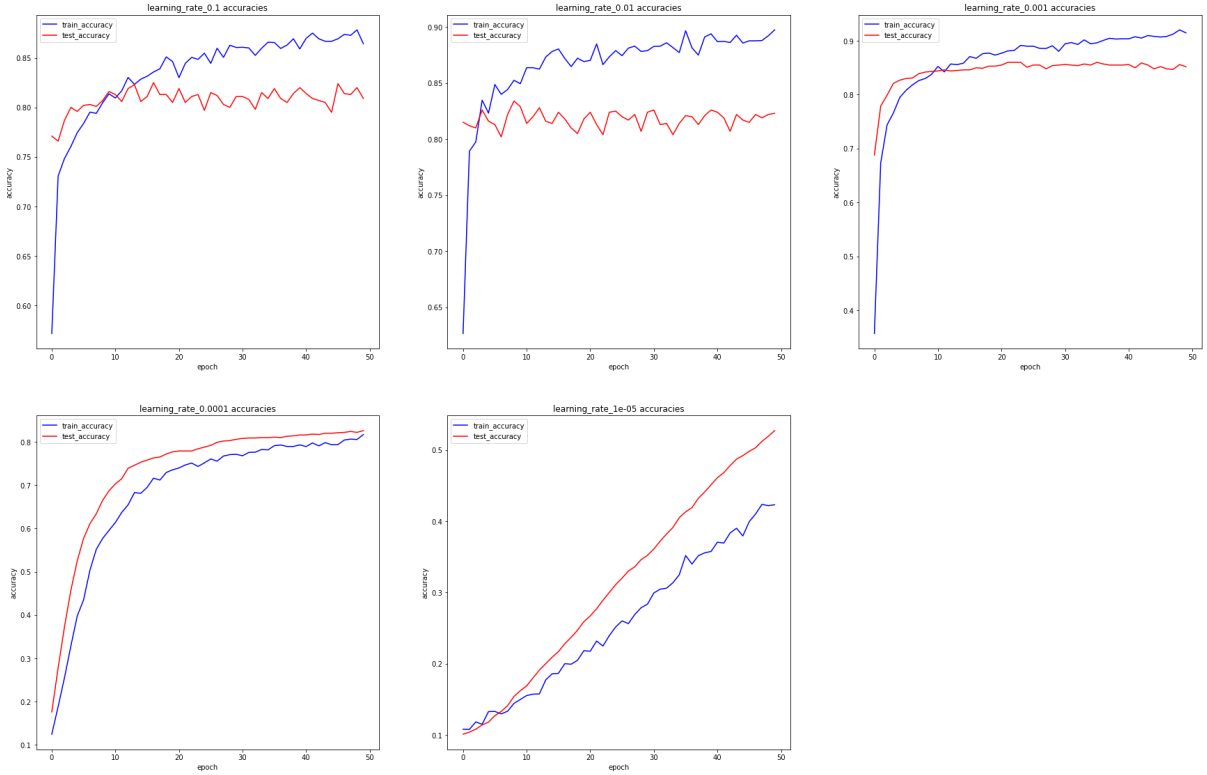

Figure: EfficientNet B0 (Noisy student), FMD, learning rate

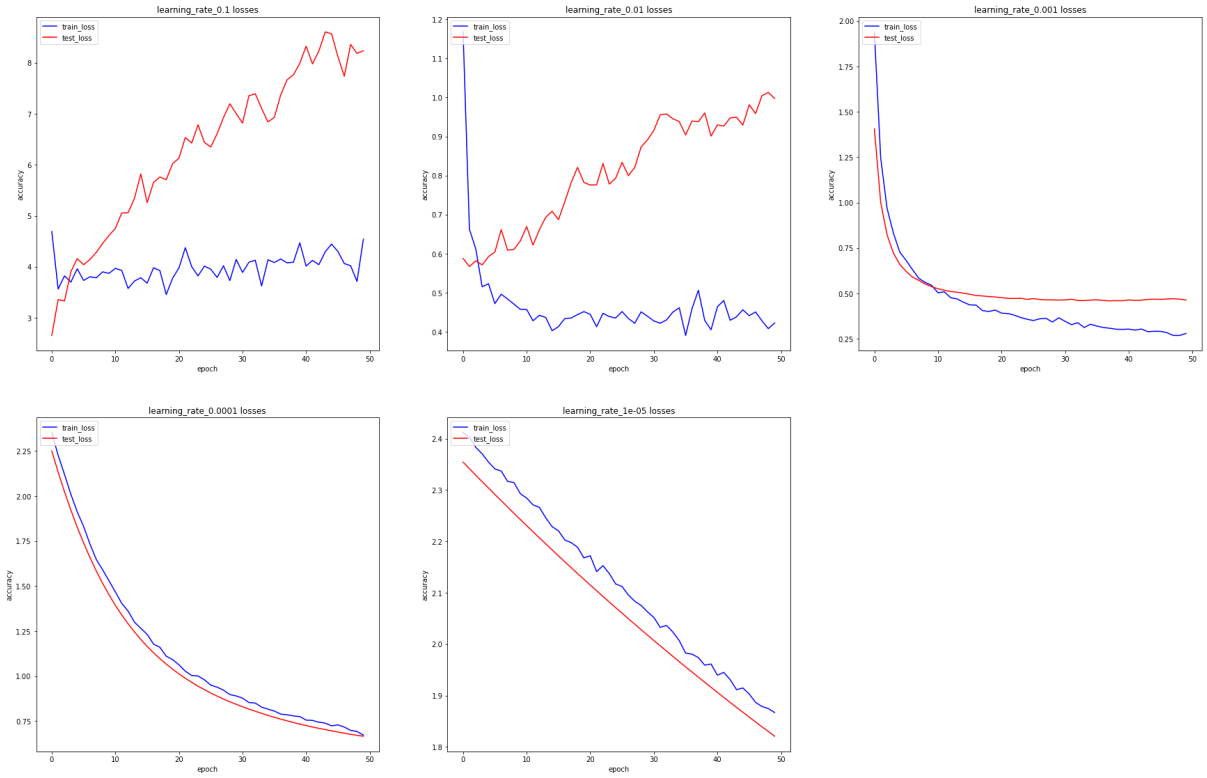

Figure: EfficientNet B0 (Noisy student), FMD, learning rate, loss

Over the last few years, there have been a series of breakthroughs in the field of computer vision, especially with the introduction of deep convolutional neural networks. In this project, we are interested in two particular neural networks, ResNet [1] and EfficientNet [2].

ResNet, stands for Residuals Network, alleviates the problem of training very deep networks through the use of residual blocks (Figure 1). These residual blocks contain “skip-connections”, which solves the problem of vanishing gradient because they enable an alternative shortcut path for the gradient to flow through.

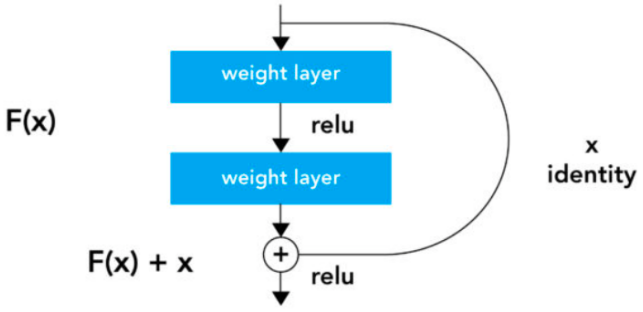

Figure 1: Residual block

On the other hand, EfficientNet is a neural network that scales up ResNet using compound scaling to scale up the depth (d), width (w) and resolution (r) in a balanced way:

$$\begin{aligned}
\text{depth: } & d = \alpha^{\phi} \\
\text{width: } & w = \beta^{\phi} \\
\text{resolution: } & r = \gamma^{\phi} \\
\text{s.t. } & \alpha \cdot \beta^2 \cdot \gamma^2 \approx 2 \\
& \alpha \geq 1, \beta \geq 1, \gamma \geq 1
\end{aligned}$$

Figure 2: Compound Scaling method. Compound coefficient φ uniformly scales up d, w and r.