Deep Learning

Lab 04

ROUBACHE Islam
i.roubache@esi-sba.dz

2024 - 2025

Contents

1 Introduction

This report is about the forth lab of the Deep Learning course. The lab is about transfer learning. We will be using shufflenet and efficientnet for this lab. The performance metrics of different models in the context of transfer learning are presented. Transfer learning involves leveraging knowledge gained from training on one task or dataset to improve learning performance on a related but different task or dataset.

1.1 OurData

The data is Chest X-rays from kaggle. It contains 21165 samples. We used startified hold out for this lab to maintain class balance.

2 Architectures Summaries

2.1

2.1.1 EfficientNet

depth = 0

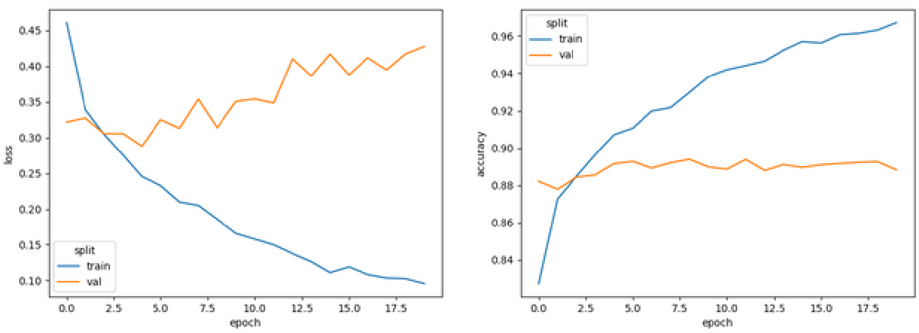


Figure 1: Accuracy And Loss

==

Total params: 5,846,144
Trainable params: 1,838,596
Non-trainableparams: 4,007,548

Input size (MB): 0.57
Forward/backward pass size (MB): 173.68
Params size (MB):22.30
Estimated Total Size (MB): 196.55

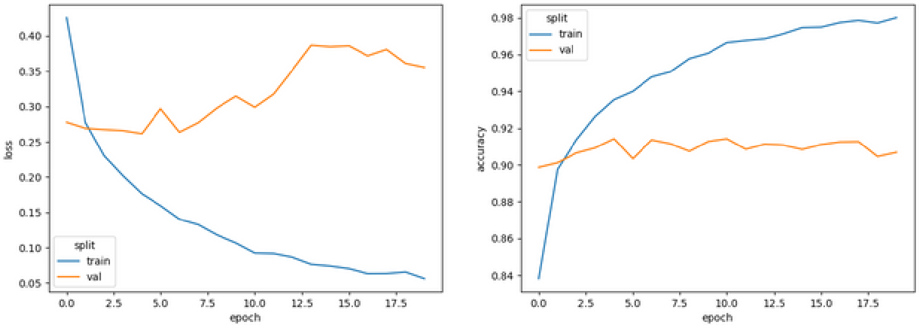


Figure 2: Accuracy And Loss

2.1.2 depth = 1

==

Total params: 5,846,144

Trainable params: 2,250,756

Non-trainableparams: 3,595,388

Input size (MB): 0.57

Forward/backward pass size (MB): 173.68

Params size (MB):22.30

Estimated Total Size (MB): 196.55

2.1.3 depth = 2

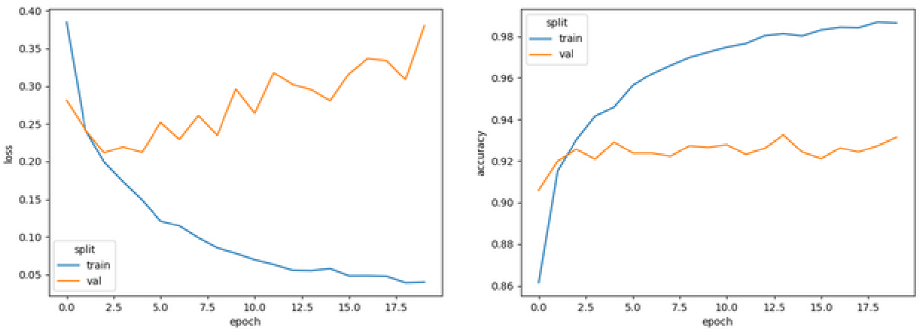


Figure 3: Accuracy And Loss

==

Total params: 5,846,144

Trainable params: 2,967,988

Non-trainableparams: 2,878,156

Input size (MB): 0.57
Forward/backward pass size (MB): 173.68
Params size (MB): 22.30
Estimated Total Size (MB): 196.55

2.2 ShuffleNet

2.2.1 depth = 0

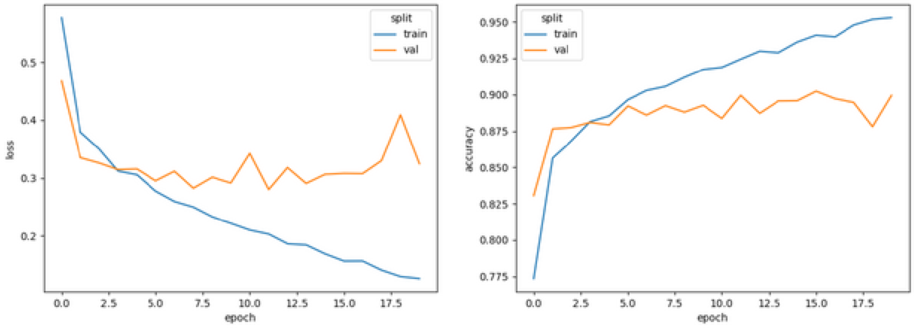


Figure 4: Accuracy And Loss

==
Total params: 2,830,056
Trainable params: 1,576,452
Non-trainable params: 1,253,604

Input size (MB): 0.57
Forward/backward pass size (MB): 47.97
Params size (MB): 10.80
Estimated Total Size (MB): 59.34

2.2.2 depth = 1

==
Total params: 2,830,056
Trainable params: 2,053,636
Non-trainable params: 776,420

Input size (MB): 0.57
Forward/backward pass size (MB): 47.97

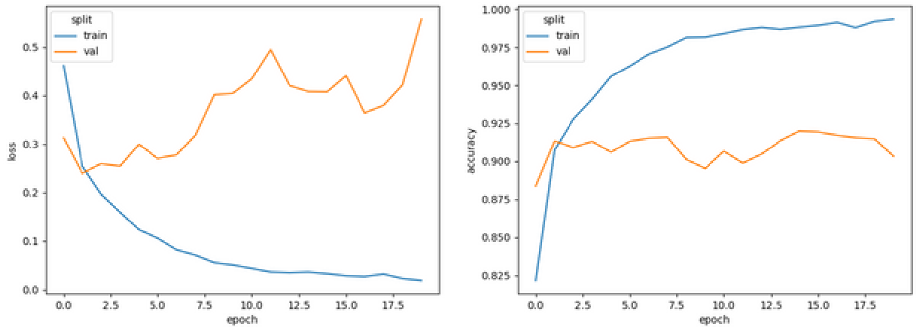


Figure 5: Accuracy And Loss

Params size (MB):10.80
Estimated Total Size (MB): 59.34

2.2.3 depth = 2

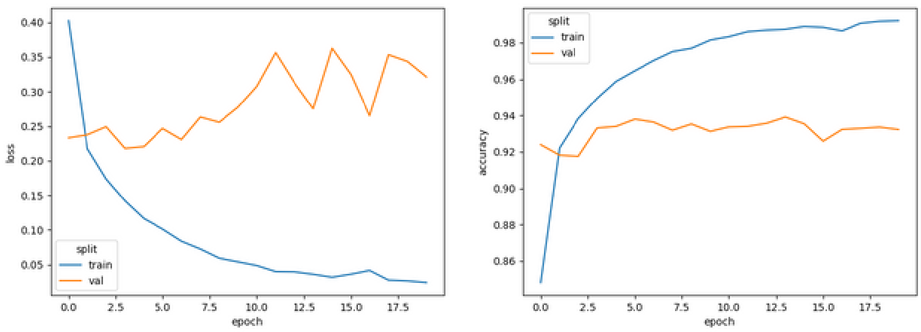


Figure 6: Accuracy And Loss

==

Total params: 2,830,056
Trainable params: 2,554,988
Non-trainableparams: 275,068

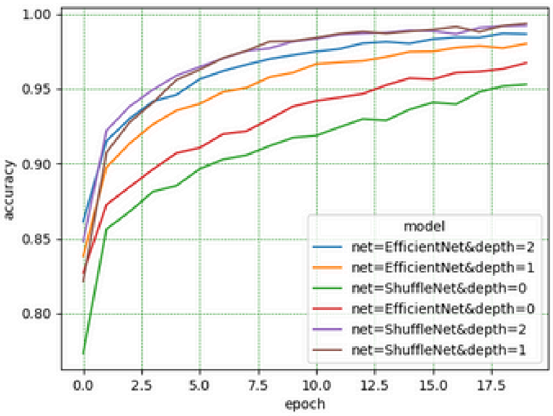
Input size (MB): 0.57
Forward/backward pass size (MB): 47.97
Params size (MB):10.80
Estimated Total Size (MB): 59.34

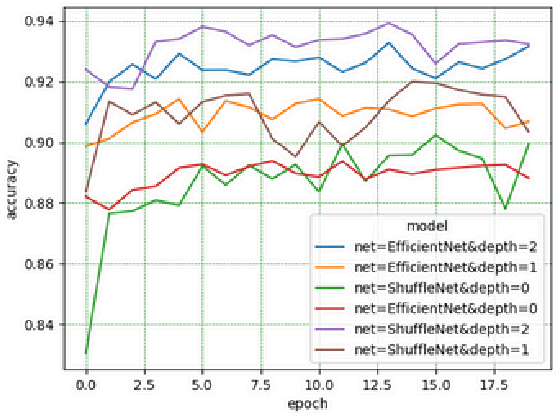
----

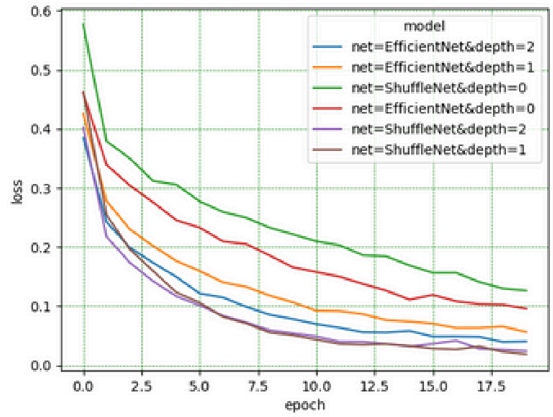
3 Results

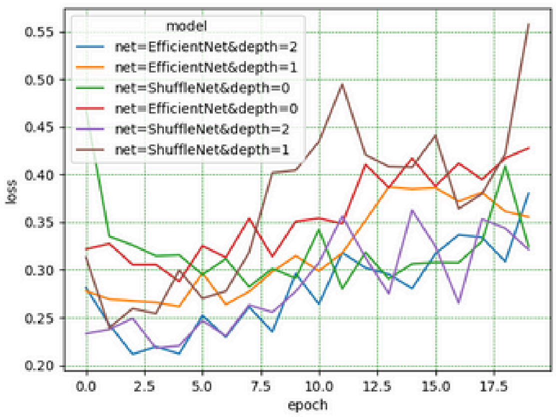
The following table shows the performance metrics of the different models:

- depth 0: (the last three fully connected)
- depth1:(oneconvolutionallayers+thelastthreefullyconnected)
- depth2:(twoconvolutionallayers+thelastthreefullyconnected)









Metric	ShuffleNetV2x0.5_0	ShuffleNetV2x0.5_1	ShuffleNetV2x0.5_2
Accuracy	0.8995	0.9036	0.9323
F1-Score	0.9063	0.9104	0.9390
Precision	0.9143	0.9134	0.9335
Recall	0.8988	0.9131	0.9448
AUC	0.9810	0.9829	0.9899

Metric	EfficientNet-B0_0	EfficientNet-B0_1	EfficientNet-B0_2
Accuracy	0.8885	0.9068	0.9313
F1-Score	0.8994	0.9159	0.9376
Precision	0.9073	0.9095	0.9459
Recall	0.8920	0.9231	0.9304
AUC	0.9769	0.9848	0.9878

Table 1: Performance Metrics of Different Models

3.1 Discussion

- Accuracy: and F1-Score: Models with depth of 2 consistently achieve the highest accuracy and F1-Score values, surpassing 0.93 and 0.937, respectively, indicating strong overall performance.

- PrecisionandRecall:Modelswithdepthof2demonstratehighprecisionandrecallvalues,suggesting effective identification of positive instances with minimal false positives.

- AreaUndertheCurve(AUC):AllmodelsexhibitAUCvaluesexceeding0.97,indicatingstrongdiscrimination ability, with Models with depth of 2 consistently achieving the highest values, surpassing 0.987.

ShuffleNet2 and EfficientNet2 emerge as the top performers across all metrics, showcasing their robustness and effectiveness for the given task.

4 conclusion

Increasing the number of fine-tuned layers adds complexity to the model. This complexity may allow the model to capture more intricate patterns in the data, potentially leading to better performance, especially if the task at hand is complex and requires high-level representations. However, overly complex models may also be prone to overfitting, especially if the dataset is small or noisy. And increasing the required computational resources.