Smoker Classification Project

Elvina Aliyeva, Asmar Sadikhova, Turana Rashidova

12 April, 2025

1 Dataset Observation and Analysis

1.1 Dataset Splitting

Our dataset contains **2** classes (*smoking* and *notsmoking*) and has been splitted into **3** subsets: train, test and validation:

```
C:\Users\turan\Desktop\ai-ml-final-project\dataset>tree
Folder PATH listing for volume OS
Volume serial number is 7C2A-624E
C:.
├───test
│   ├───notsmoking
│   └───smoking
├───train
│   ├───notsmoking
│   └───smoking
└───validation
    ├───notsmoking
    └───smoking
```

Figuur 1: Dataset Structure

1.2 Checking Train Dataset Balance

First, we examined the class distribution of the dataset to determine where there is any class imbalance or not. The distribution is represented in the Figure 2:

Figuur 2: Class Balance of the Dataset

As it stands from the figure above, there is no class imbalance, however, the number of images was insufficient to train an effective classification model. The F1 score of the initially trained model was so low, therefore, we decided to increase the number of images through image augmentation techniques.

1.3 Image Augmentation

We implemented various image augmentation techniques to improve the robustness of the classification model. Techniques included adding salt-and-pepper noise, Gaussian noise, and applying Gaussian blur.

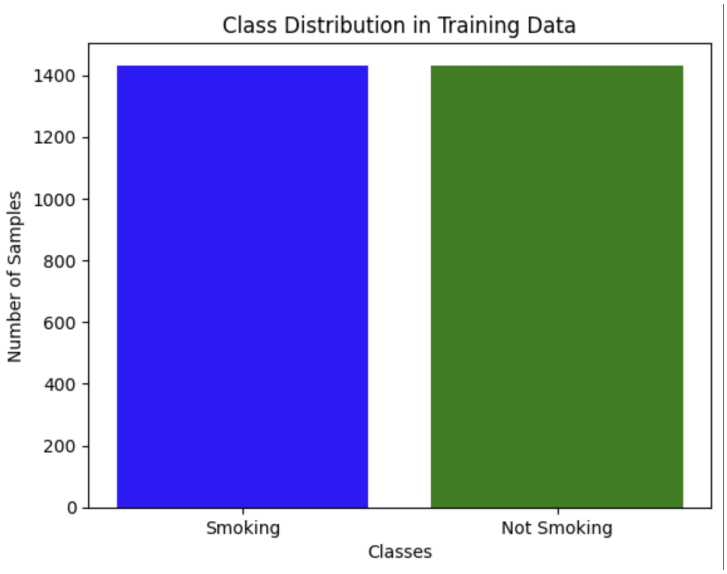
Figuur 3: Normal Image

Figuur 4: Blurred

Figuur 5: Gaussian Noise

Figuur 6: Salt and Pepper Noisy

We did not use other augmentation techniques such as adjusting brightness and contrast because the key visual indicators of smoking (e.g., a cigarette, smoke trails, hand position near mouth) are often small and subtle. Applying brightness/darkness changes globally can wash out or exaggerate these critical features, making it harder for the model to learn relevant patterns. Additionally, in real-world surveillance or detection settings (e.g., CCTV), lighting conditions are somewhat consistent. Brightness and darkness adjustments might create unrealistic scenarios that the model does not need to handle, reducing training effectiveness. After augmentation, we could achieve a higher class size of approximately 1400 samples, whose class distribution is shown below:



Figuur 7: Class Balance of the Augmented Dataset

2 Models

We used the ResNet18 and VGG16 architectures, implemented with the PyTorch library. For optimizing model performance, we employed the Adam and SGD optimizers, along with CrossEntropyLoss as the loss function.

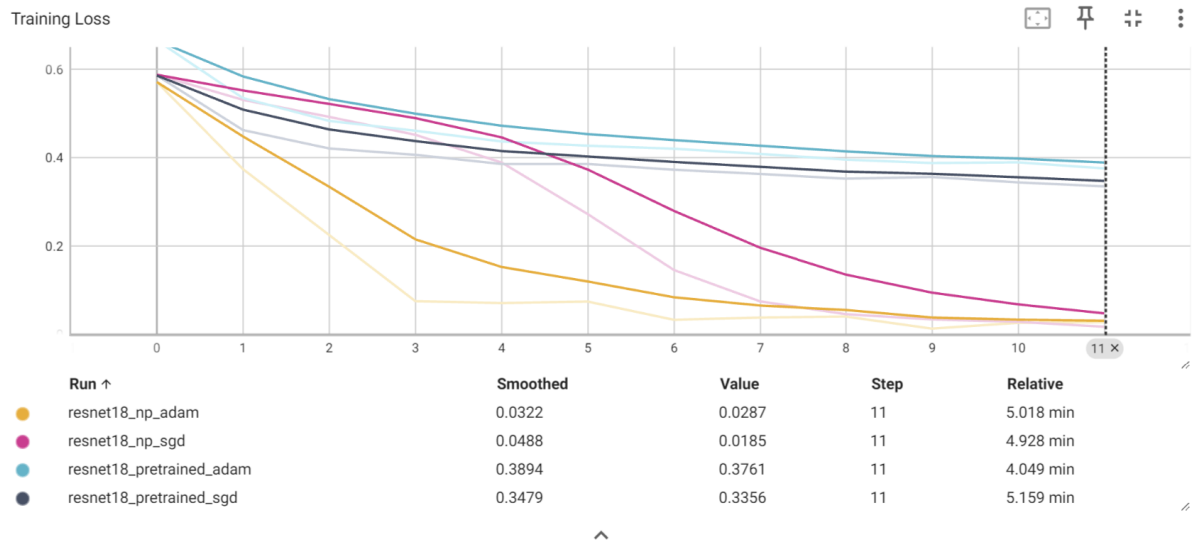
2.1 Evaluation Metrics for Model's Performance

To evaluate model's performance, we used both F1 Score (Macro) and Accuracy metrics. Although there is almost no class imbalance in our dataset, relying only on accuracy as performance measurement could be misleading. By applying Macro - Averaged F1 score along with accuracy, we ensure that both classes are treated equally, regardless of their size.

2.2 ResNet18

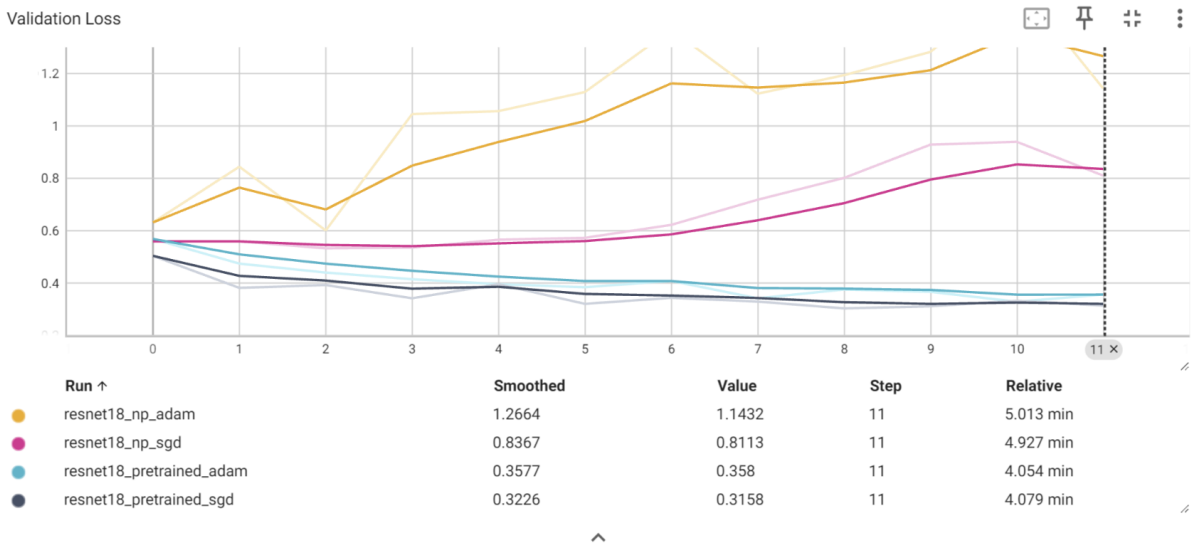
While training with ResNet18, we used 12 epochs for both pretrained and non-pretrained models.

The graphs below show the corresponding performance values of models on the dataset, in both training and validation stages:



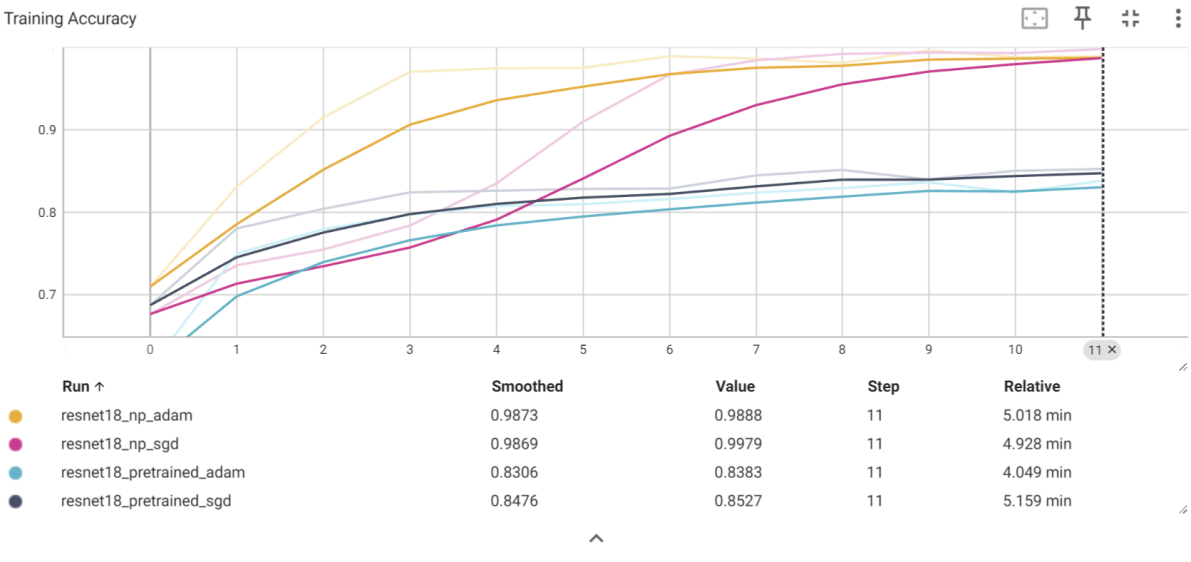
Run ↑	Smoothed	Value	Step	Relative
resnet18_np_adam	0.0322	0.0287	11	5.018 min
resnet18_np_sgd	0.0488	0.0185	11	4.928 min
resnet18_pretrained_adam	0.3894	0.3761	11	4.049 min
resnet18_pretrained_sgd	0.3479	0.3356	11	5.159 min

Figuur 8: Training Loss

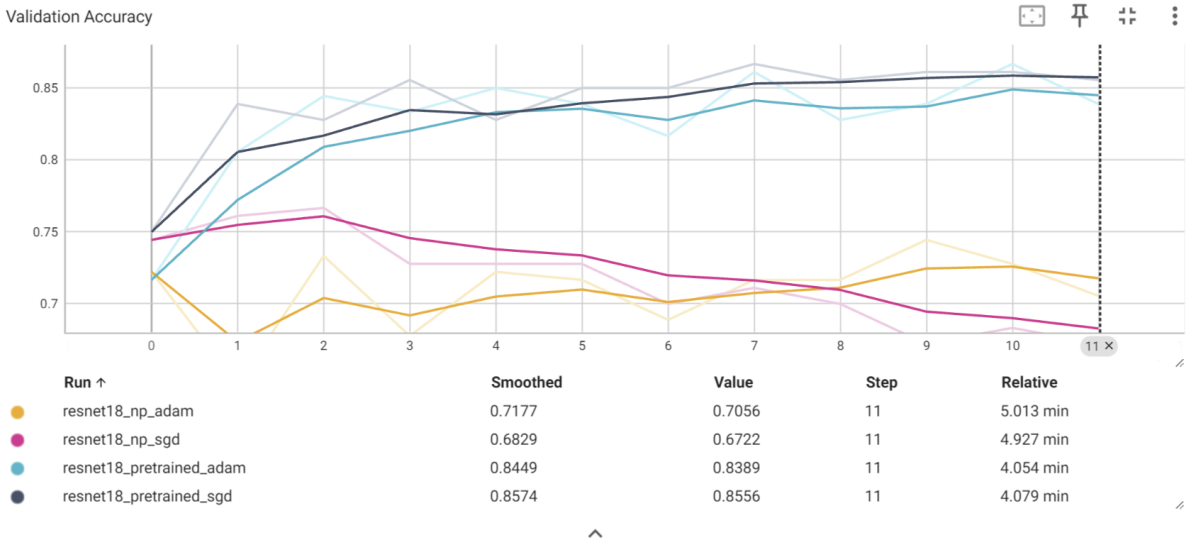


Run ↑	Smoothed	Value	Step	Relative
resnet18_np_adam	1.2664	1.1432	11	5.013 min
resnet18_np_sgd	0.8367	0.8113	11	4.927 min
resnet18_pretrained_adam	0.3577	0.358	11	4.054 min
resnet18_pretrained_sgd	0.3226	0.3158	11	4.079 min

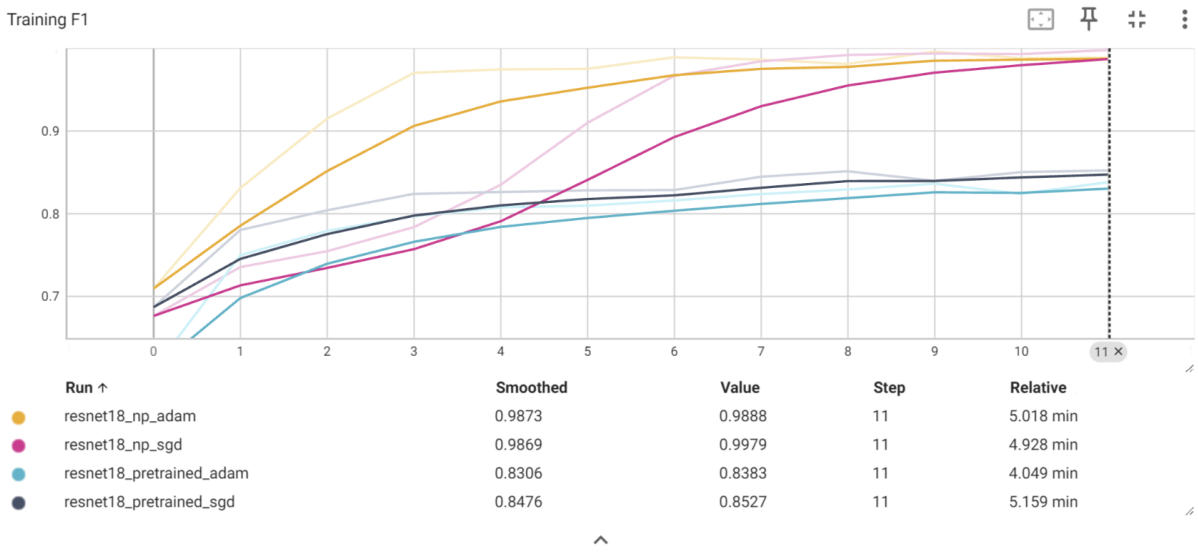
Figuur 9: Validation Loss



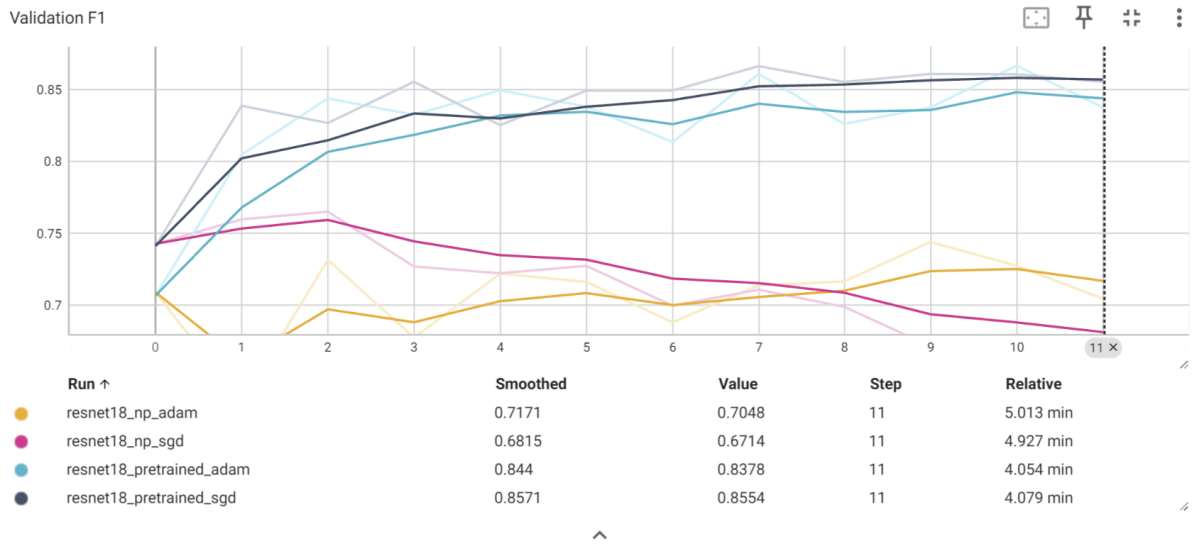
Figuur 10: Training Accuracy



Figuur 11: Validation Accuracy



Figuur 12: Training F1 Score



Figuur 13: Validation F1 Score

Training/Validation Loss: as it stands from Figure 8, the training loss decreases steadily over the epochs for all models. While the loss curves for models with pretrained weights are almost identical, those for non-pretrained models show a sharp decrease trend. The loss graph for the not pretrained weights with Adam optimizer specifically follows a dramatical drop. The validation loss graphs for pretrained models have a gradual decrease pattern while the not pretrained models rise up following an upward trend.

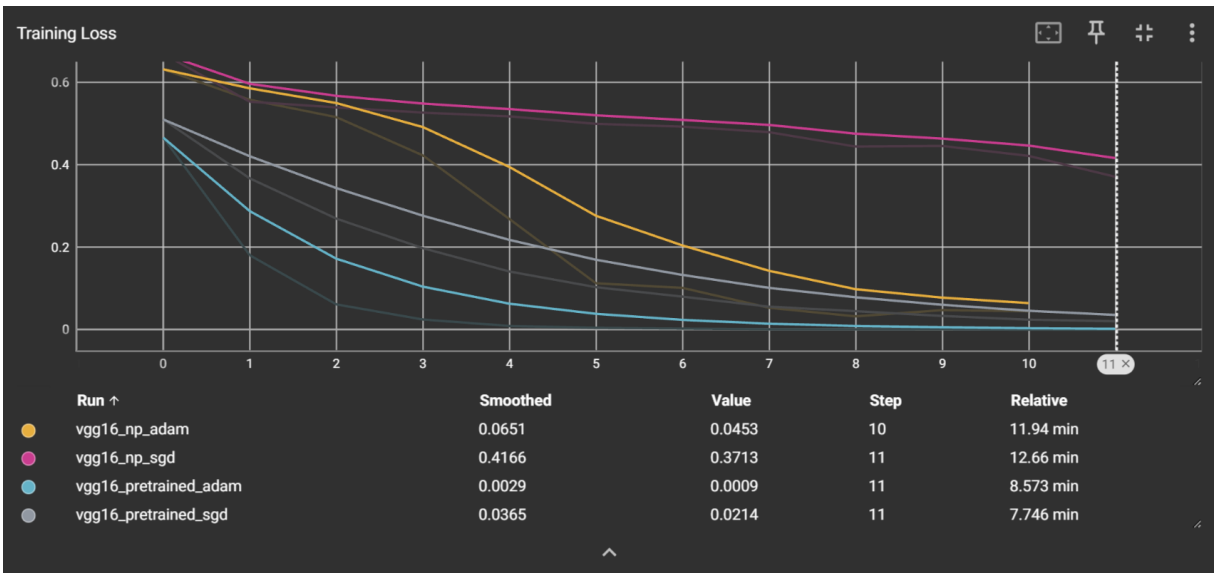
Training/Validation Accuracy/F1 Scores: both training F1 scores and accuracy metrics almost follow the similar pattern over the epochs for all models. While models with pretrained weights show a moderate and smooth increase over the epochs, we observe a sharp trend in not pretrained models for both metrics. Regarding validation metrics, non-pretrained models follow a fluctuated drop while the same fluctuation in pretrained models with an increased pattern, for both accuracy and F1 score. Since the accuracy and macro F1 score are almost the same, we can conclude that having fewer image classes does not impact the model's performance.

If we were to choose the best model from these four models, the pretrained ResNet18 with the SGD optimizer would be the top one. Pretrained models benefit from being trained on a large and diverse dataset, enabling them to generalize better and perform well on variations in input data. On the other hand, non-pretrained models, trained only on the specific dataset, might be more likely to misclassify irrelevant images, such as presence of objects resembling cigarettes or hands near the mouth mimicking the smoking gesture, as smokers, due to their limited exposure to different types of images.

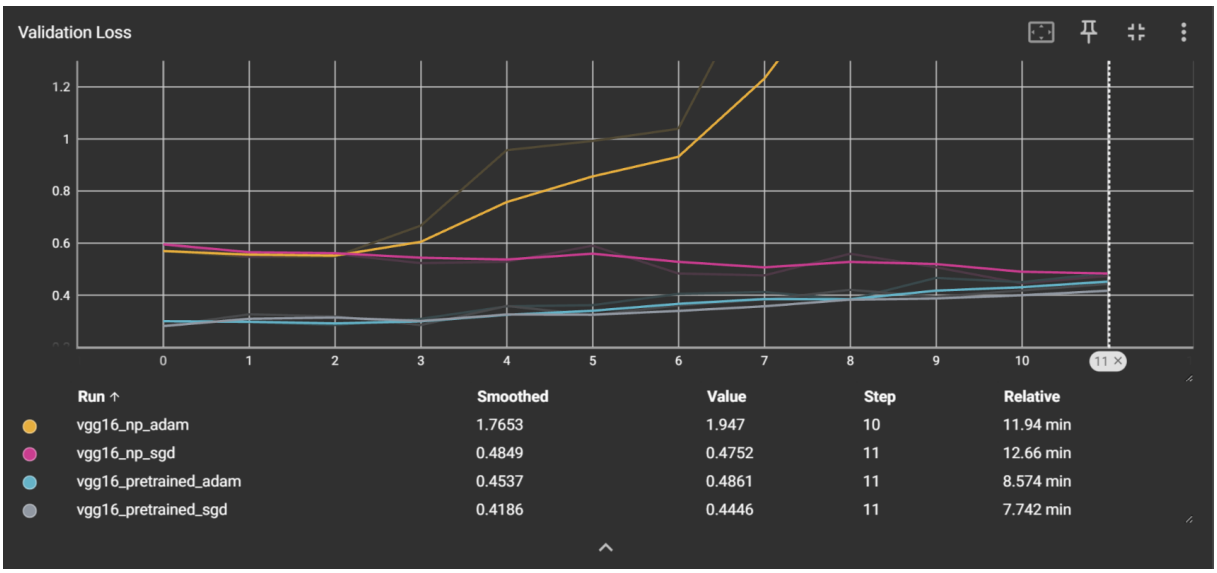
2.3 VGG16

While training with VGG16, we used 12 epochs for both pretrained and non-pretrained models.

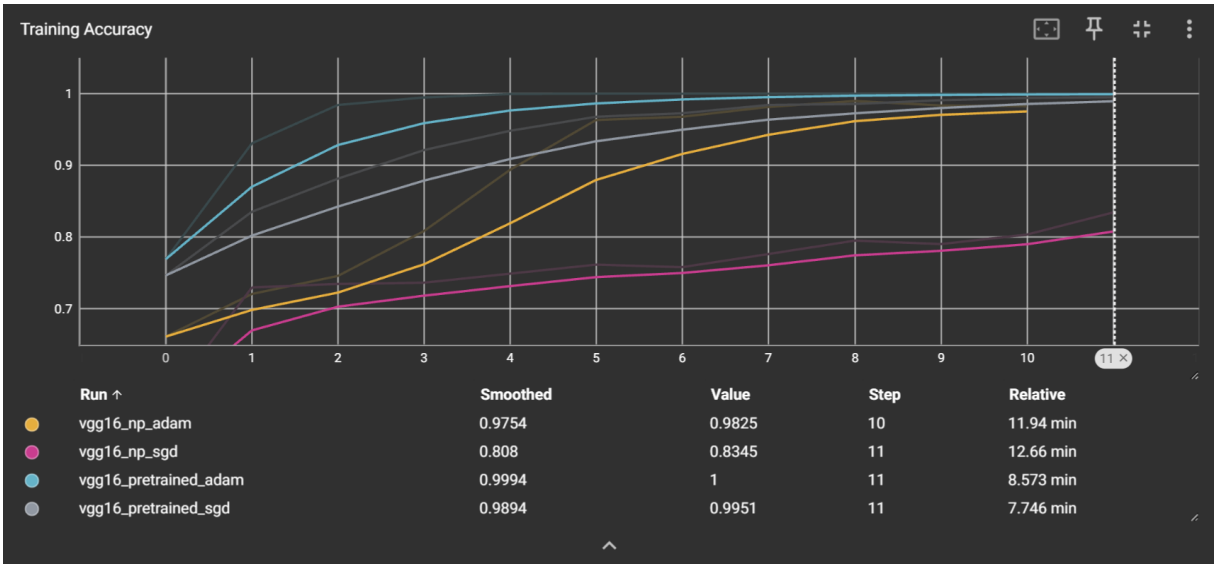
The graphs below show the corresponding performance values of models on the dataset, in both training and validation stages:



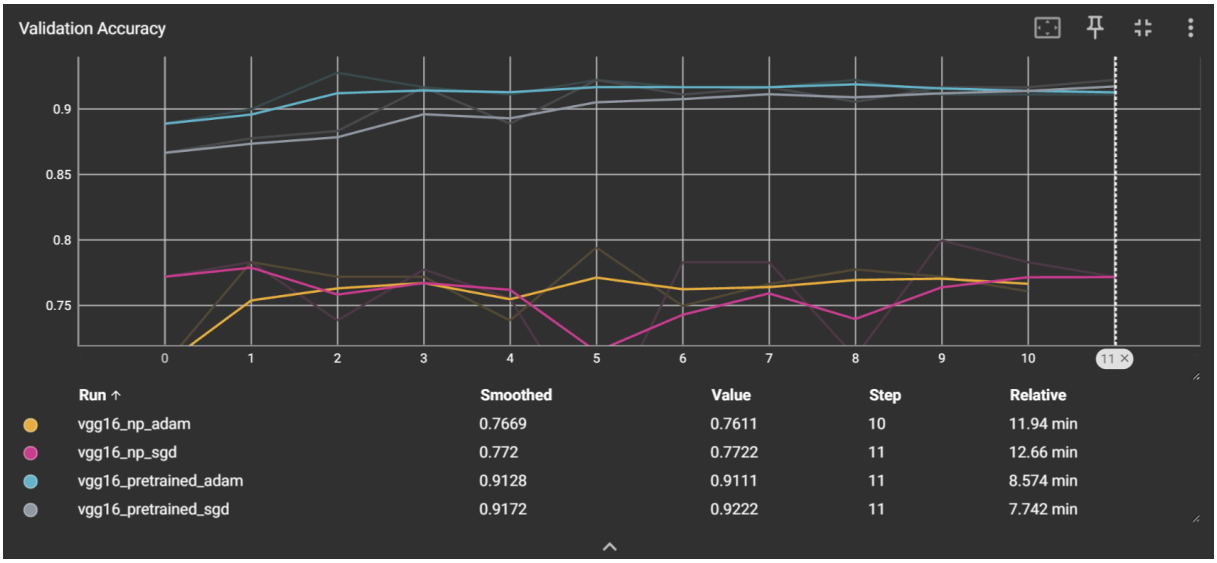
Figuur 14: Training Loss



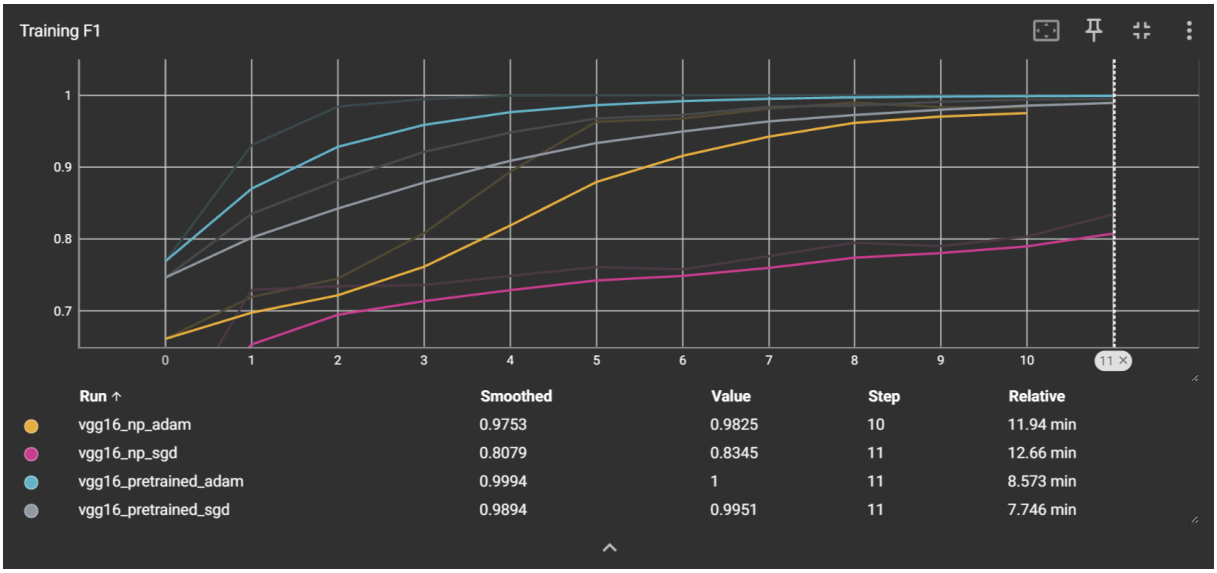
Figuur 15: Validation Loss



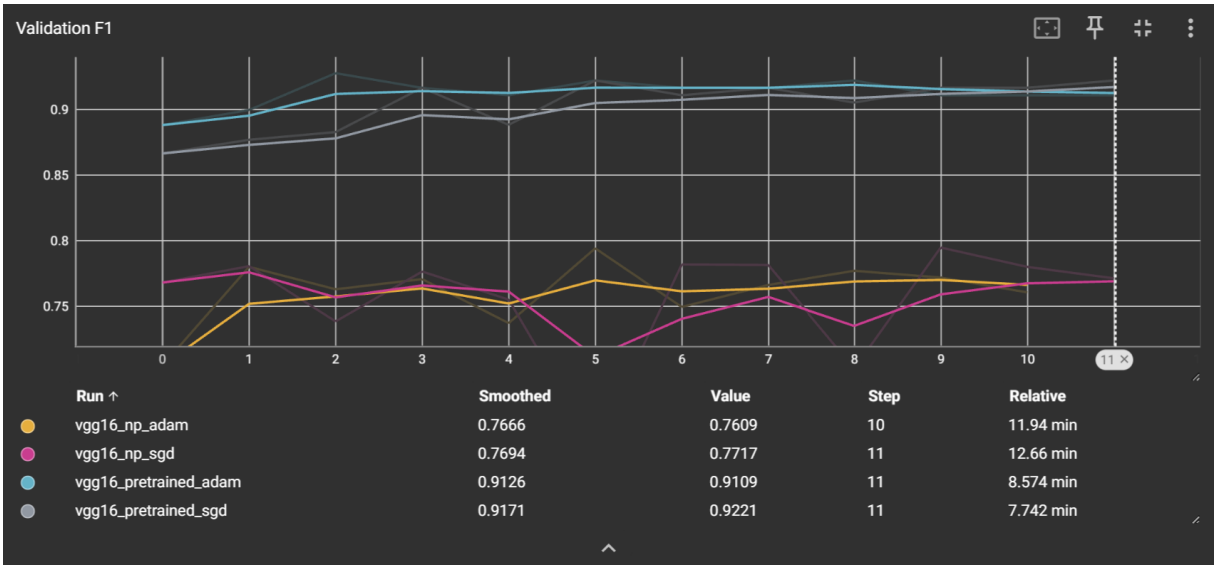
Figuur 16: Training Accuracy



Figuur 17: Validation Accuracy



Figuur 18: Training F1 Score



Figuur 19: Validation F1 Score

Training/Validation Loss: as it stands from Figure 14 and 15, the training loss decreases steadily over the epochs for all models. The loss curves for models with pretrained weights are almost similar. The validation loss graph not pretrained model with Adam optimizer shows a sharp increase, meaning that the model starts overfitting; while the corresponding graph with SDG optimizer shows a slight downward trend over the epochs. Pretrained models also show overfitting, meaning that further steps should be taken to prevent this process, such as tuning hyperparameters (learning rate, model size, batch size, etc.) or adding more data.

Training/Validation Accuracy/F1 Score: in all VGG16 models, we observe a notable growth in the training accuracy and F1 score curves. Among those curves, the pretrained models start with significantly higher values, which gradually improve over the epochs and stays almost constant during the latest ones. We observe sharp fluctuations in validation accuracy and F1 score graphs of not pretrained models (the values themselves are so low as well) while validation curves for pretrained models show a gradual increase over the epochs starting from very high values.

Again, we can say that having fewer image classes does not impact the model's performance because accuracy and macro - weighted F1 scores are almost equal. We can choose the pretrained VGG16 model with Adam optimizer as the best choice, due to its robustness and superior accuracy and F1 score, both in training and validation.

Model	Pretrained / Not Pretrained	Optimizer	Loss		Accuracy		F1 Score	
			Train	Validation	Train	Validation	Train	Validation
ResNet18	Pretrained	SDG	0.3356	0.3158	0.8527	0.8556	0.8527	0.8554
ResNet18	Pretrained	Adam	0.3761	0.358	0.8383	0.8389	0.8383	0.8378
ResNet18	Not Pretrained	SDG	0.0185	0.8113	0.9979	0.6722	0.9979	0.6714
ResNet18	Not Pretrained	Adam	0.0287	1.1432	0.9888	0.7056	0.9888	0.7048
VGG16	Pretrained	SDG	0.0214	0.4446	0.9951	0.9222	0.9951	0.9221
VGG16	Pretrained	Adam	0.0009	0.4861	1	0.9111	1	0.9109
VGG16	Not Pretrained	SDG	0.3713	0.4752	0.8345	0.7722	0.8345	0.7717
VGG16	Not Pretrained	Adam	0.0453	1.947	0.9825	0.7611	0.9825	0.7609

Figuur 20: Model Performance on Dataset — Training and Validation

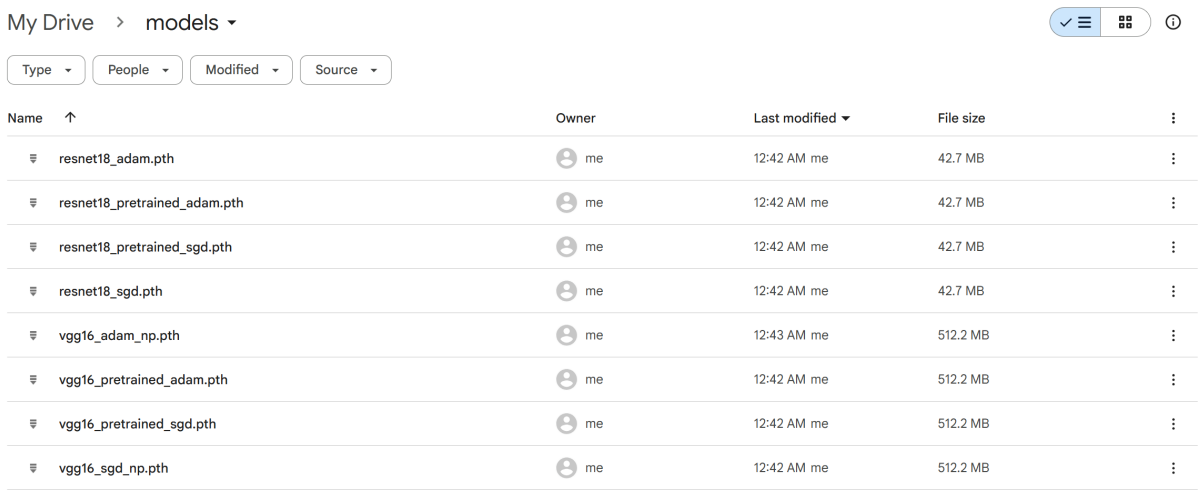
2.4 Test Scores for Models

Model	Pretrained / Not Pretrained	Optimizer	Accuracy	F1 Score (Macro)
ResNet18	Pretrained	SDG	0.8214	0.8212
ResNet18	Pretrained	Adam	0.8304	0.8303
ResNet18	Not Pretrained	SDG	0.6652	0.6602
ResNet18	Not Pretrained	Adam	0.6875	0.6792
VGG16	Pretrained	SDG	0.8571	0.8569
VGG16	Pretrained	Adam	0.8438	0.843
VGG16	Not Pretrained	SDG	0.75	0.7487
VGG16	Not Pretrained	Adam	0.7098	0.7077

Figuur 21: Model Evaluation Results — Accuracy and F1 Score

3 Saving Models

We have saved the trained models in the specified folder for future use:



Figuur 22: Saving Models