Project SSY098

Cecilia Nyberg
cecnyb@chalmers.se
990106

***Abstract*—During the last decade, great advances has been made in the field of computer vision [Man24]. The usage of convolutional neural networks has become one of the most popular approaches to solving image recognition tasks. However, when designing a CNN there are many aspects to consider. In this report, the performance of CNNs with different configurations is compared when classifying images of trees. These images come from the TRUNK12 dataset and contain images of bark from 12 different tree types. First, a base model was created which reached an accuracy of 66%. Different improvements were then added and an ablation study was made for these improvements. The best improvements found were using residual connections, normalize the data and use batch normalization. The ablation study did not show strong improvements of using PReLU instead of regular ReLU. It did not show improvements of using early stopping either, but this could be because of the regularization techniques used or that the patience was too low. The highest received accuracy was 79.8%.**

I. Method

When designing the model, there were several aspects to consider, along with different trade-offs. If the model was too simple, it would fail to find patterns in the data, but if it was too complex it could become too specialized on the training data and fail to generalize on unseen data. Additionally, the higher the complexity, the more computationally demanding the training. Therefore, the plan was to start simple and increase the complexity of the model only if it was deemed necessary.

The network architecture used for the base model was a sequential Convolutional Neural Network (CNN) with multiple convolutional and pooling layers, implemented using PyTorch. ReLU was used as the activation function, since it helps mitigate the diminishing and/or exploding gradient problem [Mad22].

With the intention of creating a more stable and reliable model, normalization of the input data was used. Each color channel was normalized independently. Normalization helps the model find meaningful patterns across various features and prevents larger or higher-intensity features from dominating lower-intensity ones. The normalization was done using zero-centering and standard deviation 1. Zero-centering is particularly beneficial when using ReLU as the activation function, as it prevents gradient saturation and improves training dynamics [Mad22].

The model was trained using a training loop, which took a model, train loader, optimizer, criterion, and number of epochs as arguments. In each epoch, the loss and accuracy for the training data were calculated.

To improve the baseline model, several additions were made. The first one was batch normalization, which is a regularization technique [Sax24]. Like other regularization techniques, batch normalization prevents overfitting and makes the model converge faster. Batch normalization layers standardize and normalize output from the previous layer before it is sent to the next layer. This is done over each batch. Batch normalization also mitigates the diminishing or exploding gradient problem, where gradients deeper in the network either become really small or very large.

Another improvement that was added was changing the activation function. Instead of regular ReLU, parameterized ReLU, PReLU, was used. PReLU accounts for the dying ReLU problem, where negative inputs result in inactive neurons [Ola23]. PReLU uses a slope with a learnable parameter for negative inputs, instead of setting the gradient to zero for negative inputs as regular ReLU does. One downside to this is that it increases complexity since the model gets more learnable parameters. It was possible to have the model learn one slope value for each input channel, but because of computational complexity, only one slope value over all input channels was used.

Additionally, early stopping was used. Early stopping is another regularization technique that prevents overfitting by having the training terminate when the validation accuracy starts to decrease [Bro19]. Without this functionality, it can be difficult to know how many epochs to use when training the model. To use early stopping, a new training loop was implemented which compares validation accuracy over epochs. If the validation accuracy has not exceeded its highest value over a specific number of epochs, called the *patience*, it terminates. To use this new training loop, the training data had to be split into validation and training data. This was done using the torch function *random_split* with split size 80% for the training data and 20% validation data.

The last improvement added was residual connections, moving away from the completely sequential base model. Residual connections, or skip connections, let input skip a few layers and add it directly to the output from these skipped layers [T23]. This helps increase performance for deep models. Although the model used in this report may not have had a very large number of layers, it included multiple convolutional layers, batch normalization, activation functions, and a fully connected head. Therefore, it could still benefit from the effects of using residual connections. One benefit of residual connections is that they hinder the vanishing and/or exploding gradient problem. This was probably not a very big concern for the improved model since both normalization of the data, batch normalization, and PReLU were used, which all mitigate this problem. What was more important was that residual connections help with the degradation problem, where performance drops for deeper models. This is not because of over- or underfitting, but because, for example, useful information can get lost or weakened further down in the network, making it harder for deeper layers to learn relevant features.

An ablation study was made for the improved model to see which, or if all, improvements were beneficial.

The evaluation of the models was done using accuracy as evaluation metric.

A. *ID and OOD distributions*

To separate images into in-distribution (ID) and out-of-distribution (OOD) samples, feature extraction was performed on the last convolutional layer in the trained CNN. The distribution of these features was plotted in a histogram to identify any clear distinctions between the ID and OOD distributions. Following this, a principal component analysis (PCA) was conducted on the mean of the features over the spatial dimensions to determine which features had the highest variance. It would have been possible to not calculate the mean over the spatial dimensions, but this would have resulted in many more feature values to compare. Therefore, it was decided to first try to use the mean and see if it would give adequate distributions. The aim was to find features with the most spread-out distributions, potentially showing distinct distributions for the two classes. The PCA was performed using Sklearn's pre-built principal component analyzer.

Additionally, to mitigate the effect of outliers on the variance, a second PCA was performed after Winsorization had been applied to the data. This was achieved using the scipy function *winsorize*. Winsorize is a function that reduces the impact of outliers by capping extreme values. The parameter 0.05 was used with the winsorize function, which resulted in the data being capped at the 5th and 95th percentiles. For example, the lowest 5% of the data was set to the value at the 5th percentile, and the highest 5% was set to the value at the 95th percentile.

II. Experimental Evaluation

A. *Baseline model*

The layout for the base-line model is presented in table **??**.

Table I
CNN Model Architecture

Layer	**Description**
Block One	Conv2d: Convolutional Layer with 3 input channels, 32 output channels, and kernel size of 5. ReLU: ReLU Activation Function. MaxPool2d: Max Pooling Layer with kernel size of 2.
Block Two	Conv2d: Convolutional Layer with 32 input channels, 64 output channels, and kernel size of 5. ReLU: ReLU Activation Function. MaxPool2d: Max Pooling Layer with kernel size of 2.
Block Three	Conv2d: Convolutional Layer with 64 input channels, 128 output channels, and kernel size of 3. ReLU: ReLU Activation Function. MaxPool2d: Max Pooling Layer with kernel size of 2.
Head	Flatten: Flatten Layer. Linear: Fully Connected Layer with input size 128×2×2 and output size 512. ReLU: ReLU Activation Function. Dropout: Dropout Layer with probability 0.5.
Output	Linear: Fully Connected Layer with input size 512 and output size 12.

A learning rate of 0.001 and 10 epochs were used to train the baseline model. The loss and training accuracy for these epochs are presented in Table II. While the accuracy on the training data provides an indication of how well the model is learning local patterns, it does not necessarily reflect the model's generalization ability. However, it is notable that the training loss decreases steadily over the epochs, showing useful weight adjustments.

It was difficult to decide the number of epochs without validation data. Training for too many epochs could lead to overfitting, but too few could lead to underfitting. Therefore, 10 epochs were chosen. The accuracy on the test set was 0.66, indicating that the model learned some meaningful patterns and performed significantly better than a random classifier would for the 12 classes.

To potentially improve performance, additional epochs could have been used, and their impact on the test data could have been evaluated. However, optimizing the model based on test data results is not a valid approach, as it would lead to overfitting to the test set. Therefore, this was not pursued.

Epoch	Loss	Accuracy
1/10	1.8606	0.3371
2/10	1.4704	0.4819
3/10	1.2863	0.5454
4/10	1.1205	0.6064
5/10	0.9966	0.6546
6/10	0.9218	0.6793
7/10	0.8389	0.7038
8/10	0.7617	0.7318
9/10	0.7036	0.7489
10/10	0.6551	0.7682

Table II
TRAINING RESULTS

B. *Improvements*

The improvements mentioned in the previous section were added to the model and the new model is presented in table III. A patience of 3 was used for this model.

The training and validation accuracy for this model is presented in figure 1. Both the validation and training accuracy was trending upwards even when the training terminated. A larger patience than 3 could maybe have resulted in an even better model. This is a balance since a higher patience could also result in the model starting to overfit.

The improved model reached an accuracy of 78 %, which is much higher than for the base-line model. This indicates that the improvements were beneficial. Which improvements were most useful is discussed in the next section about the ablation study.

A confusion matrix for the predictions is shown in figure 2. The confusion matrix shows that the chestnut had the highest number of misclassification and that oak tree overall was the most common label to missclassify as. Beech had the lowest number of misclassification.

Table III
RESNET MODEL ARCHITECTURE

Block One
Conv2d: Convolutional Layer with 3 input channels, 32 output channels, and kernel size of 5.
BatchNorm2d: Batch Normalization over out_channels.
PReLU: Parametric ReLU Activation Function.
Residual connection added to output
MaxPool2d: Max Pooling Layer with kernel size of 2.
Block Two
Conv2d: Convolutional Layer with 32 input channels, 64 output channels, and kernel size of 5.
BatchNorm2d: Batch Normalization over out_channels.
PReLU: Parametric ReLU Activation Function.
Residual connection added to output
MaxPool2d: Max Pooling Layer with kernel size of 2.
Block Three
Conv2d: Convolutional Layer with 64 input channels, 128 output channels, and kernel size of 3.
BatchNorm2d: Batch Normalization over out_channels.
PReLU: Parametric ReLU Activation Function.
Residual connection added to output
MaxPool2d: Max Pooling Layer with kernel size of 2.
Head
Flatten: Flatten Layer.
Linear: Fully Connected Layer with input size 3200 and output size 512.
PReLU: Parametric ReLU Activation Function.
Dropout: Dropout Layer with probability 0.5.
Output
Linear: Fully Connected Layer with input size 512 and output size 12.

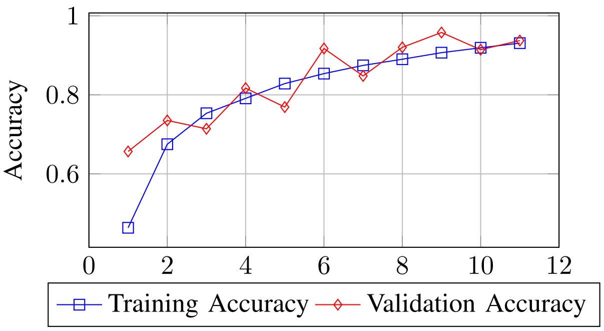


Figure 1. Training and Validation Accuracy improved model

C. *Ablation study*

1) Normalization removed: Even though the normalization was used for the base model as well, the improved model was run one time without the normalization to see its effect.

Figure 3 shows some normalized and unnormalized data as a comparison. The contrasts in the normalized data are much clearer than in the unnormalized even for smaller changes in pixel intensities.

The validation and training accuracy for this configuration over epochs are presented in figure 4. The test

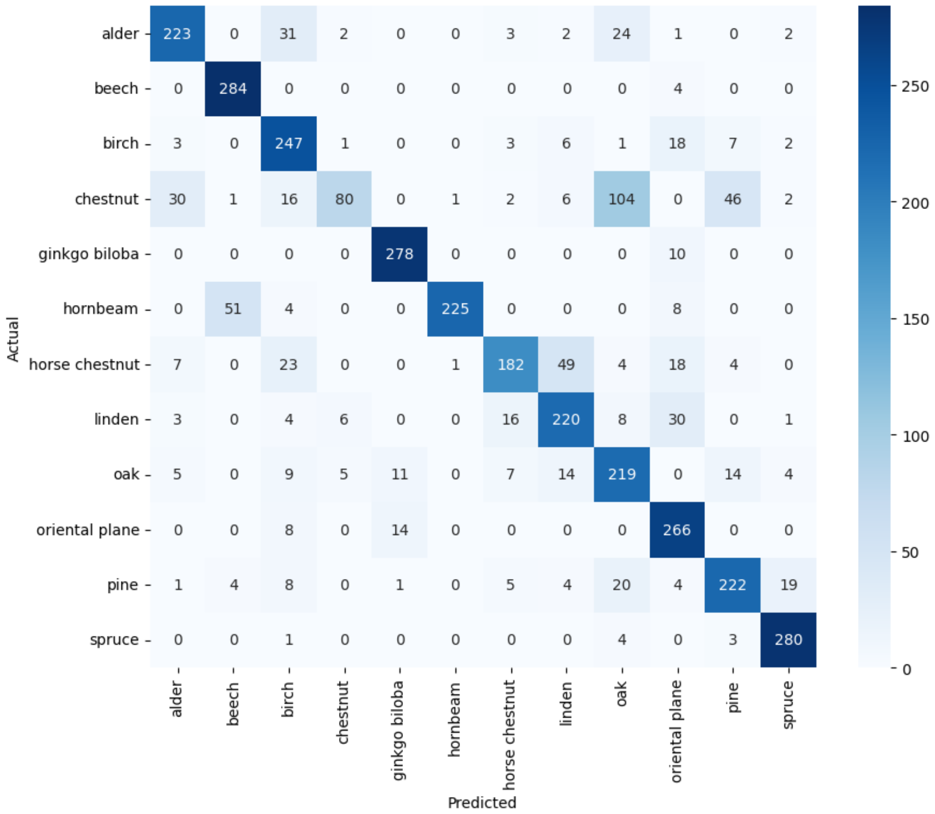


Figure 2. Confusion matrix improved model

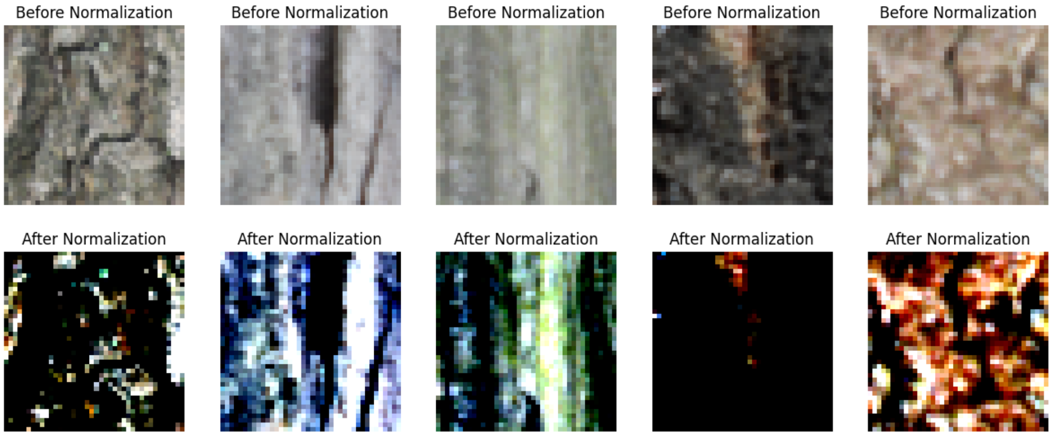


Figure 3. Visualization of normalization

accuracy received for this model was 0.34. As can be seen in the plot, the model starts to overfit very quickly and performs much worse than the model using normalization. This shows how beneficial using normalization is for this model with this data. Without it, the model struggles to generalize and find meaningful patterns in the data. The confusion matrix for the model without normalization is presented in figure 5 and it is very different from the one where normalization is used. A very high number of instances are classified as horse chestnut and oriental plane.

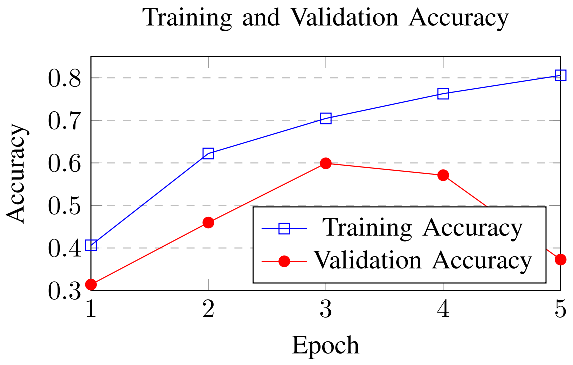


Figure 4. Training and Validation Accuracy without normalization

2) Batch Normalization removed: Secondly, the improved model was run without using batch normalization. The validation and training accuracy when training this model is shown in figure 6. The model ran for many more epochs before it terminated this time since the validation and training accuracy kept increasing. The model looks even better than with batch normalization, but when used on the test data the accuracy was only 72%, which is 4 percentage points lower than for the model using batch normalization. This indicates that the model becomes better at generalizing when batch normalization is used.

The confusion matrix for the improved model without batch normalization is presented in figure 7. It is quite similar to the one when batch normalization was used. However, one difference is that pine was the most common tree to misclassify as. Otherwise, chestnut still had the highest number of incorrect classifications and beech the fewest.

3) PReLU removed: The third removed improvement was PReLU. Instead, regular ReLU was used as activation function. The validation and training accuracy for this run is shown in figure 8. It looks like the model is converging slower and the validation accuracy starts to stabilize at 0.9, which is lower than when PReLU was used. However, when evaluated on the test data the model reached an accuracy of 77.9% which is almost the same as the model using PReLU. This indicates that the model did not suffer from the dying ReLU problem. The additional complexity introduced by using PReLU instead of regular ReLU might therefore not be worth it.

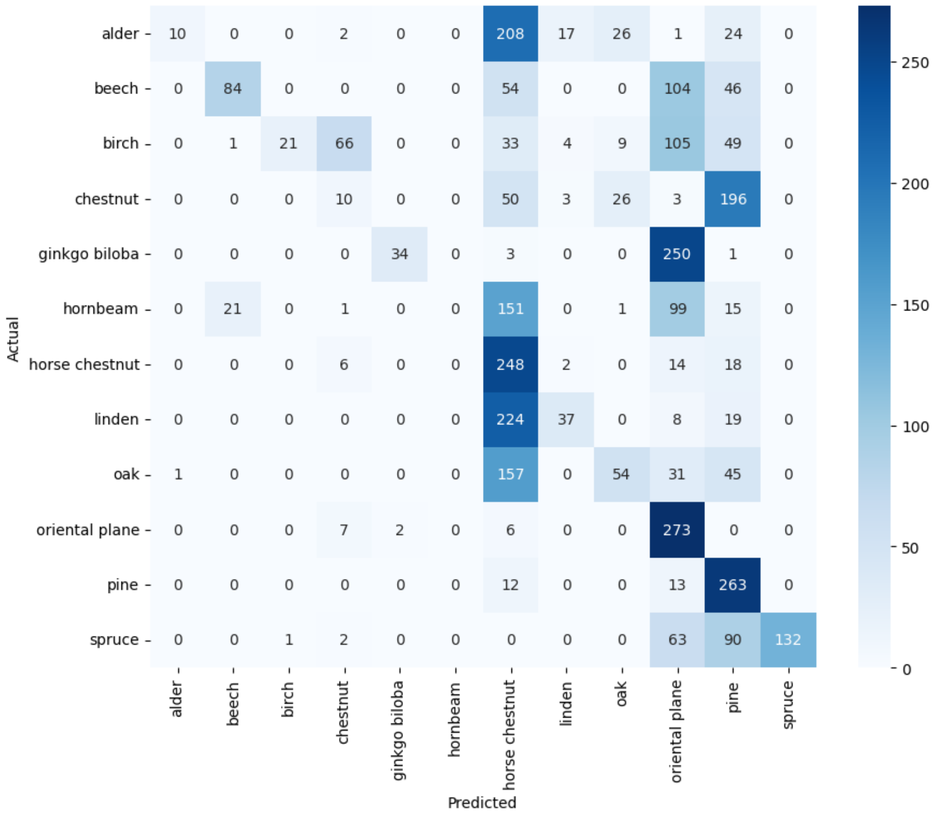


Figure 5. Confusion matrix for improved model without data normalization

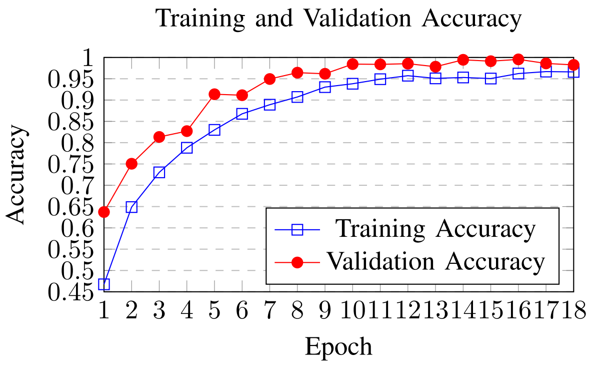
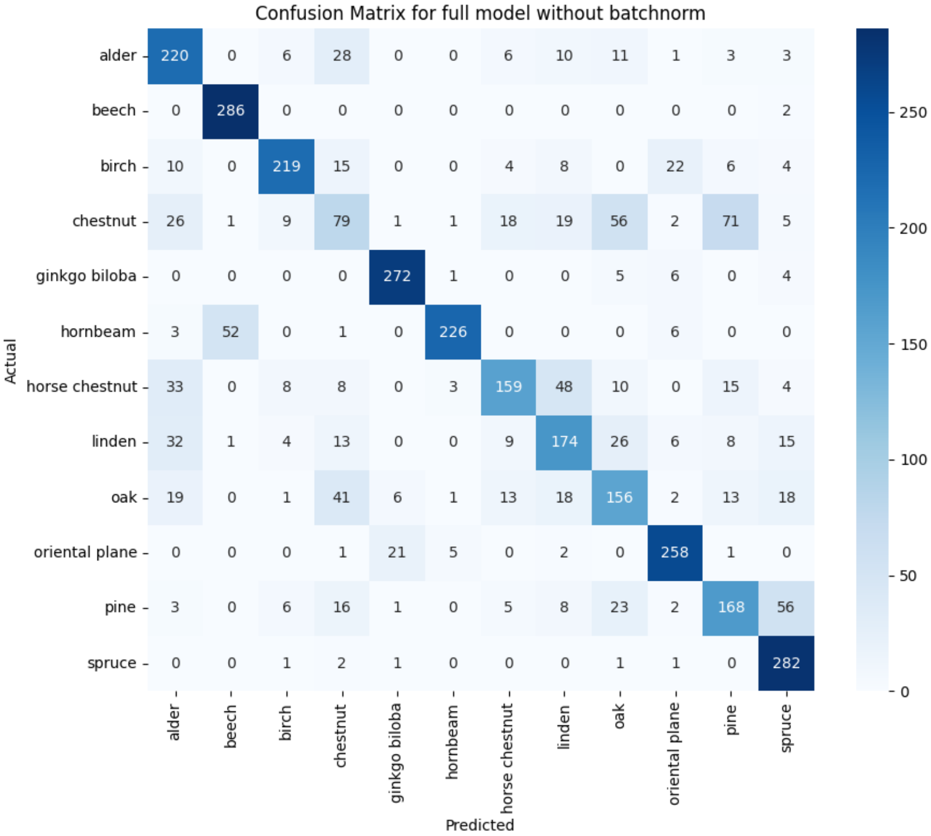


Figure 6. Training and Validation Accuracy without batch normalization

The confusion matrix when PReLU was not used is shown in figure 9. It is somewhat similar to the original one, with a few exceptions. For example, beech is now the tree with the lowest number of misclassifications. The misclassification overall are more spread out in comparison with the original case where a few trees had quite a lot of misclassification. If a more generalized and stable model is desired, it might be more beneficial to use regular ReLU instead of PReLU in this case.

4) Early stopping removed: The next improvement to remove was early stopping. The model was run on 20 epochs instead to see if it would overfit. The loss and training accuracy are presented in table IV and as can be seen, the loss steadily decreased over the epochs. The test accuracy after the 20 epochs was 79.8%, which is higher than when early stopping was used. This indicates that a too low patience might have been used so that the improved model was underfitted. But it could also be that the model without early stopping was trained on more data since no validation set was used. Additionally, since several other regularization techniques were already used, the model might be quite robust to overfitting even if it is run for a high number of epochs.

The confusion matrix for the model without early stopping is presented in figure 10. It is somewhat similar to the original confusion matrix, having many incorrect classifications of chestnut, where multiple chestnut images were classified as oak trees, although much fewer than for the original.



	alder	beech	birch	chestnut	ginkgo biloba	hornbeam	horse chestnut	linden	oak	oriental plane	pine	spruce
alder	220	0	6	28	0	0	6	10	11	1	3	3
beech	0	286	0	0	0	0	0	0	0	0	0	2
birch	10	0	219	15	0	0	4	8	0	22	6	4
chestnut	26	1	9	79	1	1	18	19	56	2	71	5
ginkgo biloba	0	0	0	0	272	1	0	0	5	6	0	4
hornbeam	3	52	0	1	0	226	0	0	0	6	0	0
horse chestnut	33	0	8	8	0	3	159	48	10	0	15	4
linden	32	1	4	13	0	0	9	174	26	6	8	15
oak	19	0	1	41	6	1	13	18	156	2	13	18
oriental plane	0	0	0	1	21	5	0	2	0	258	1	0
pine	3	0	6	16	1	0	5	8	23	2	168	56
spruce	0	0	1	2	1	0	0	0	1	1	0	282

Figure 7. Confusion matrix for improved model without batch normalization

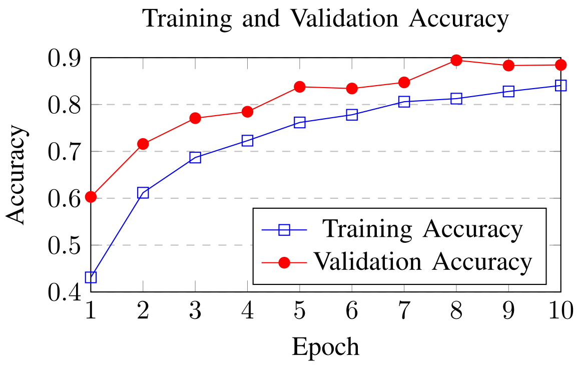


Figure 8. Training and Validation Accuracy over Epochs without PReLU

Epoch	Loss	Accuracy
1	1.6499	0.4516
2	0.9177	0.6787
3	0.7288	0.7425
4	0.5826	0.7938
5	0.5019	0.8224
6	0.4291	0.8490
7	0.3651	0.8712
8	0.3202	0.8853
9	0.2786	0.9060
10	0.2399	0.9147
11	0.2326	0.9188
12	0.1976	0.9308
13	0.1888	0.9367
14	0.1515	0.9461
15	0.1580	0.9456
16	0.1345	0.9529
17	0.1553	0.9488
18	0.1465	0.9520
19	0.1272	0.9579
20	0.1014	0.9672

Table IV
TRAINING LOSS AND ACCURACY WITHOUT EARLY STOPPING

5) Residual connections removed: The last improvement to remove was the residual connections. The validation and training accuracy is presented in figure 11. As can be seen in the figure, the model only ran for five epochs before it terminated because the validation accuracy started to decrease. This indicates that the residual connections are an important improvement for this model. When evaluated on the test data, an accuracy of 63% was received, which is much lower than when residual connections where used. The model might therefore suffer from degradation problem even if it is not very deep. Additionally, residual connections help in preserv-

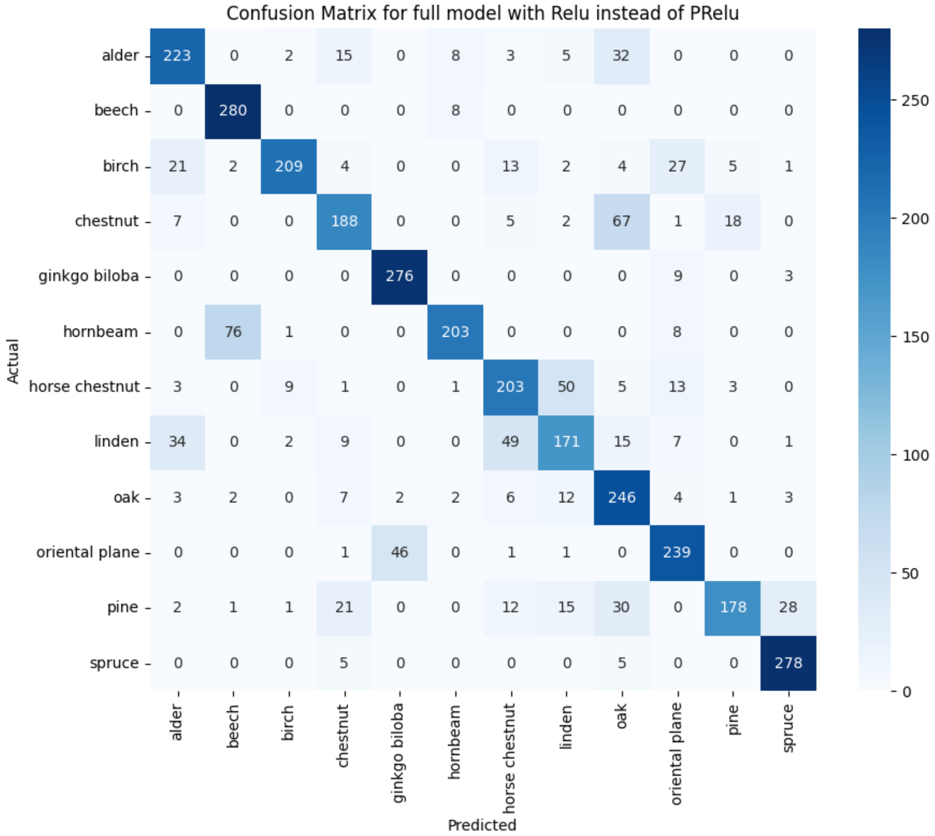


Figure 9. Confusion matrix for improved model without PReLU

ing important features and gradients during training, which is beneficial given the small details and subtle differences in the data.

The confusion matrix when residual connections were removed is presented in figure 12. It differs a bit from the original case, having many images incorrectly classified as chestnut, especially images of oak trees. On the other hand, for the original model many images of oak trees were classified as chestnut instead. Additionally, oriental plane was wrongfully classified as ginkgo biloda multiple times.

D. *Visualize subset of images*

Some example classifications are presented in figure 13. Two incorrect classifications show the common mixup between oak tree and chestnut. These two tree types seem to have similar features.

E. *ID and OOD results*

To be able to classify images as ID or OOD, the features from the last convolutional layer in the CNN were extracted. A histogram of the frequencies of feature values for all features combined was created and is presented in figure 14. This distribution does not give a clear distinction of the two classes, which might not be very surprising. This is because some features might be characterized by having low values for ID samples and high values for OOD or vice versa. When all these values are combined, they can outweigh each other so that there is no distinction between ID and OOD samples. Therefore, a principal component analysis was made with the goal of finding individual features characterized by having distinct values for ID and OOD samples. The ten features with the highest variance were features 68, 89, 100, 60, 127, 25, 11, 6, 90 and 81 in order. The plot in figure 15 shows the result of the PC analysis for the features in the same order they were

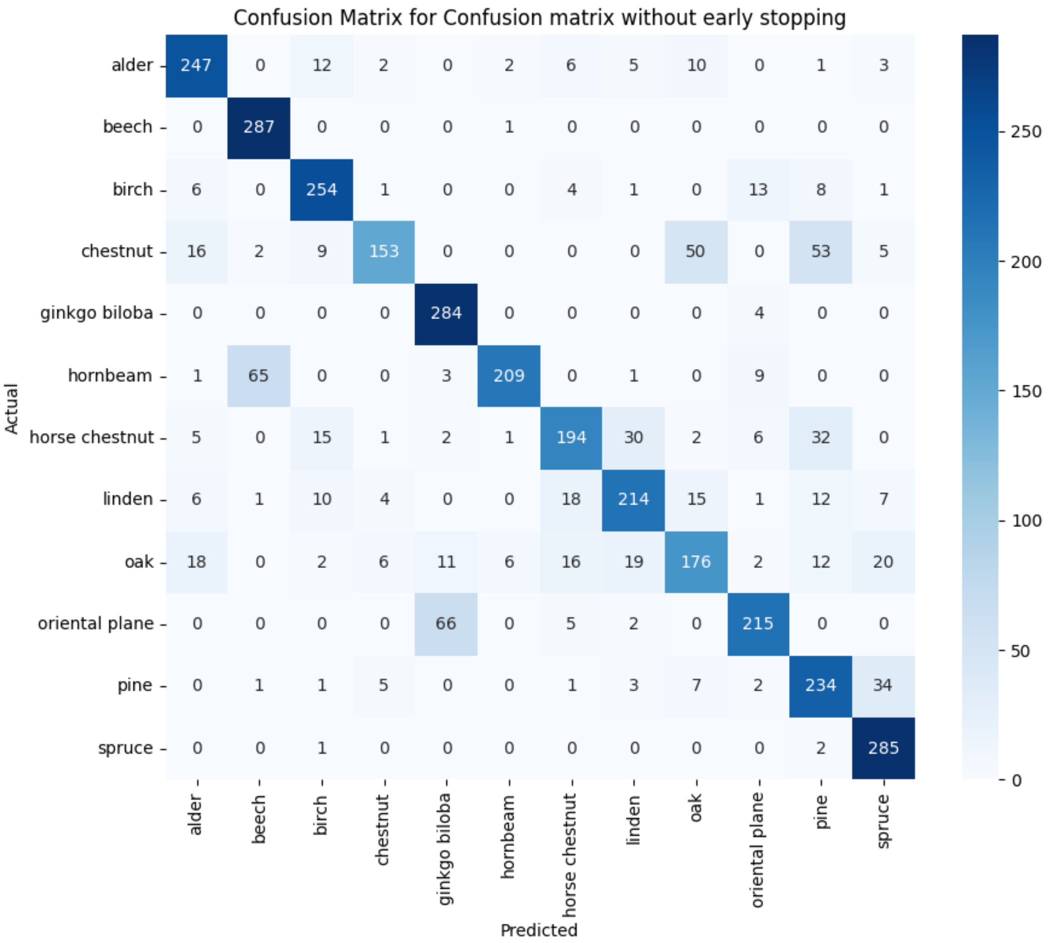


Figure 10. Confusion matrix for improved model without early stopping

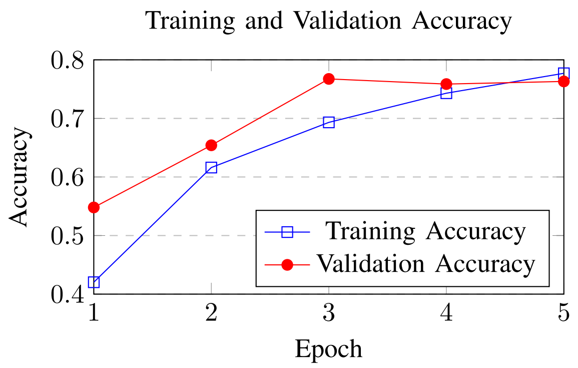


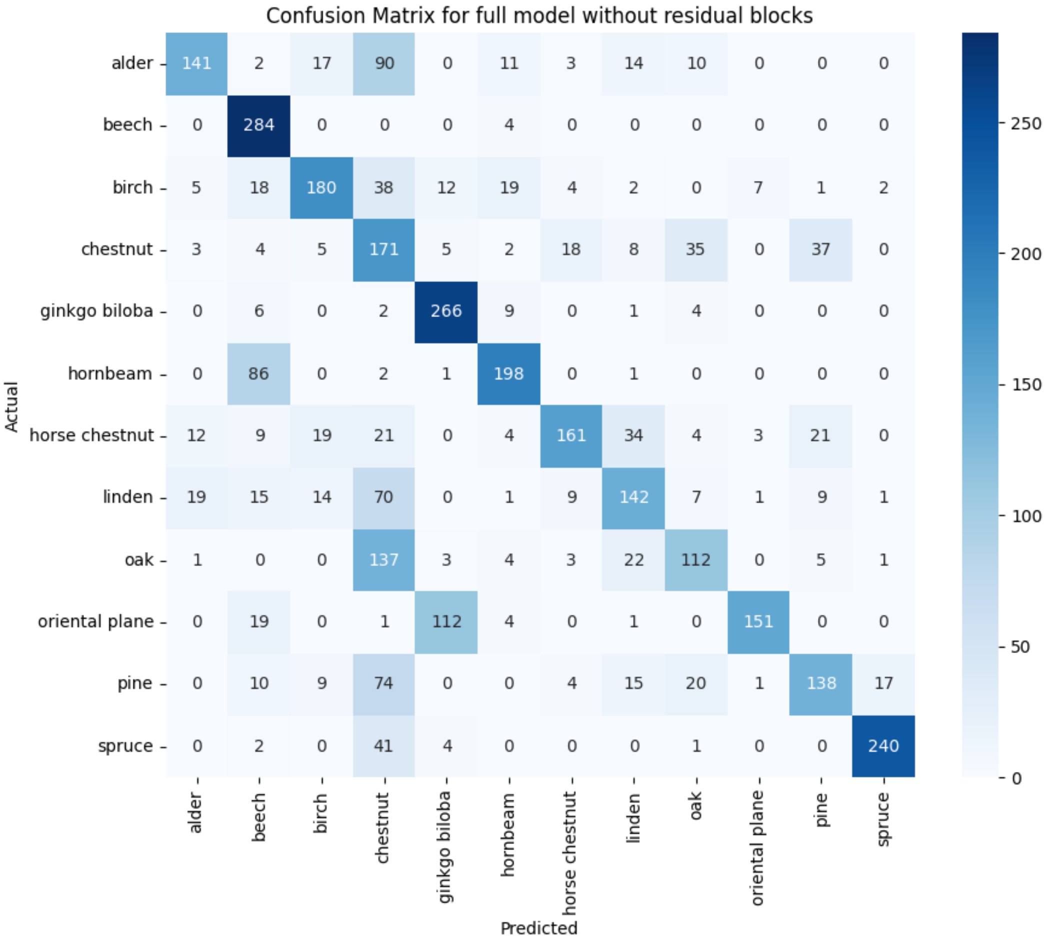
Figure 11. Training and Validation Accuracy without residual connections

presented. The plot shows that feature 68 had the highest variance.

The distributions for the found features are presented in figure 16. Many histograms in figure 16 are not obviously separated into two distinct distributions. It does however look like feature 11, 90 and 81 could have two distributions since they all have two separated peaks. These features were therefore chosen and a threshold was decided by looking at their histograms, threshold 0.5 for feature 11, 0.6 for feature 90 and -0.1 for feature 81. The results for these are presented in table V. Feature 81 had its distribution the other way around, so the accuracy is 0.92 if the labels are switched. All features received quite high accuracies and proved useful for separating instances as belonging to ID and OOD.

Feature	Threshold	Accuracy
11	0.5	0.9115
90	0.6	0.9100
81	-0.1	0.0815

Table V
ACCURACY ID AND OOD FOR DIFFERENT FEATURES WITH CORRESPONDING THRESHOLDS



Actual \ Predicted	alder	beech	birch	chestnut	ginkgo biloba	hornbeam	horse chestnut	linden	oak	oriental plane	pine	spruce
alder	141	2	17	90	0	11	3	14	10	0	0	0
beech	0	284	0	0	0	4	0	0	0	0	0	0
birch	5	18	180	38	12	19	4	2	0	7	1	2
chestnut	3	4	5	171	5	2	18	8	35	0	37	0
ginkgo biloba	0	6	0	2	266	9	0	1	4	0	0	0
hornbeam	0	86	0	2	1	198	0	1	0	0	0	0
horse chestnut	12	9	19	21	0	4	161	34	4	3	21	0
linden	19	15	14	70	0	1	9	142	7	1	9	1
oak	1	0	0	137	3	4	3	22	112	0	5	1
oriental plane	0	19	0	1	112	4	0	1	0	151	0	0
pine	0	10	9	74	0	0	4	15	20	1	138	17
spruce	0	2	0	41	4	0	0	0	1	0	0	240

Figure 12. Confusion matrix for improved model without residual connections

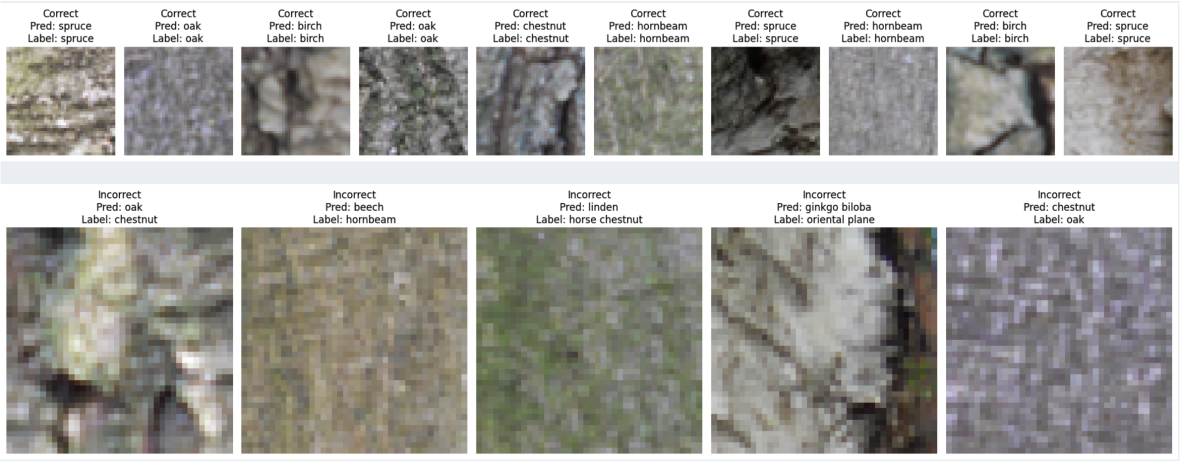


Figure 13. Example classifications for subset of images

A downside of using principal component analysis is that it only compares isolated features. In many cases, combining features and analyzing several features together gives much better results. It would be possible to manually add features together and then do a principal component analysis on combined features, but the number of features to consider would increase very rapidly with this approach. In some cases, it can be worth it but in this case, the separate features already performed fairly well. To still get an

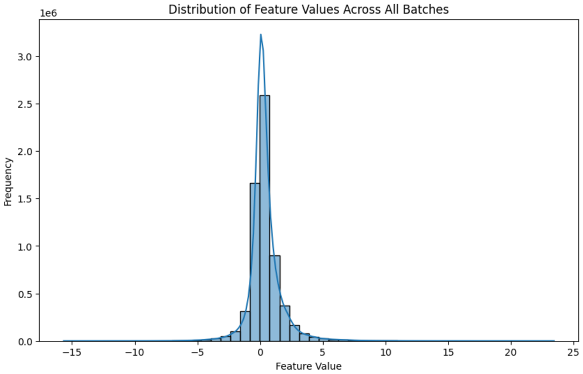


Figure 14. Histogram of distribution of all features

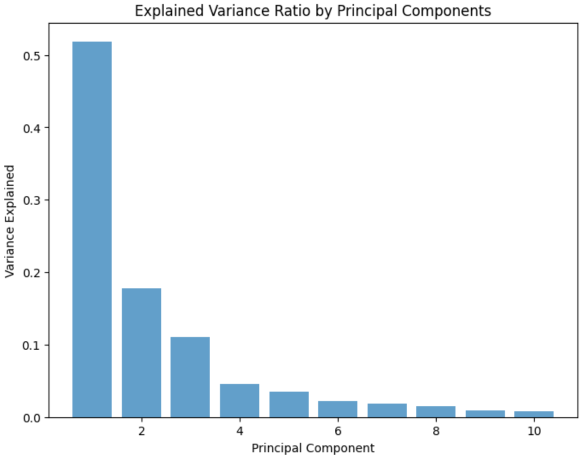


Figure 15. Features with the highest variance

idea of the benefits of combining features, feature 11 and 90 were combined since they had thresholds at similar values. Using two features instead of just one could make the classification more robust and less dependent on local deviations.

The histogram of the distributions for the combined feature is presented in figure 17 and a distinction between the two distributions can be seen. By looking at the histogram, threshold 1.5 was chosen for the combined feature. Using this threshold generated an accuracy of 97.6 %, which is much higher than for the separate features.

A confusion matrix for the classifications is presented in table VI, showing a balance in the number of ID and OOD images that were misclassified, indicating a good threshold. Some missclassifications are shown in figure 18. It can be seen that many similar images of OOD instances were misclassified. It might be that the line in the middle of the coffee bean is mistaken as a feature for bark which also often has a line with a different contrast in the middle.

	Predicted	
Actual	ID	OOD
ID	975	25
OOD	23	977

Table VI
CONFUSION MATRIX

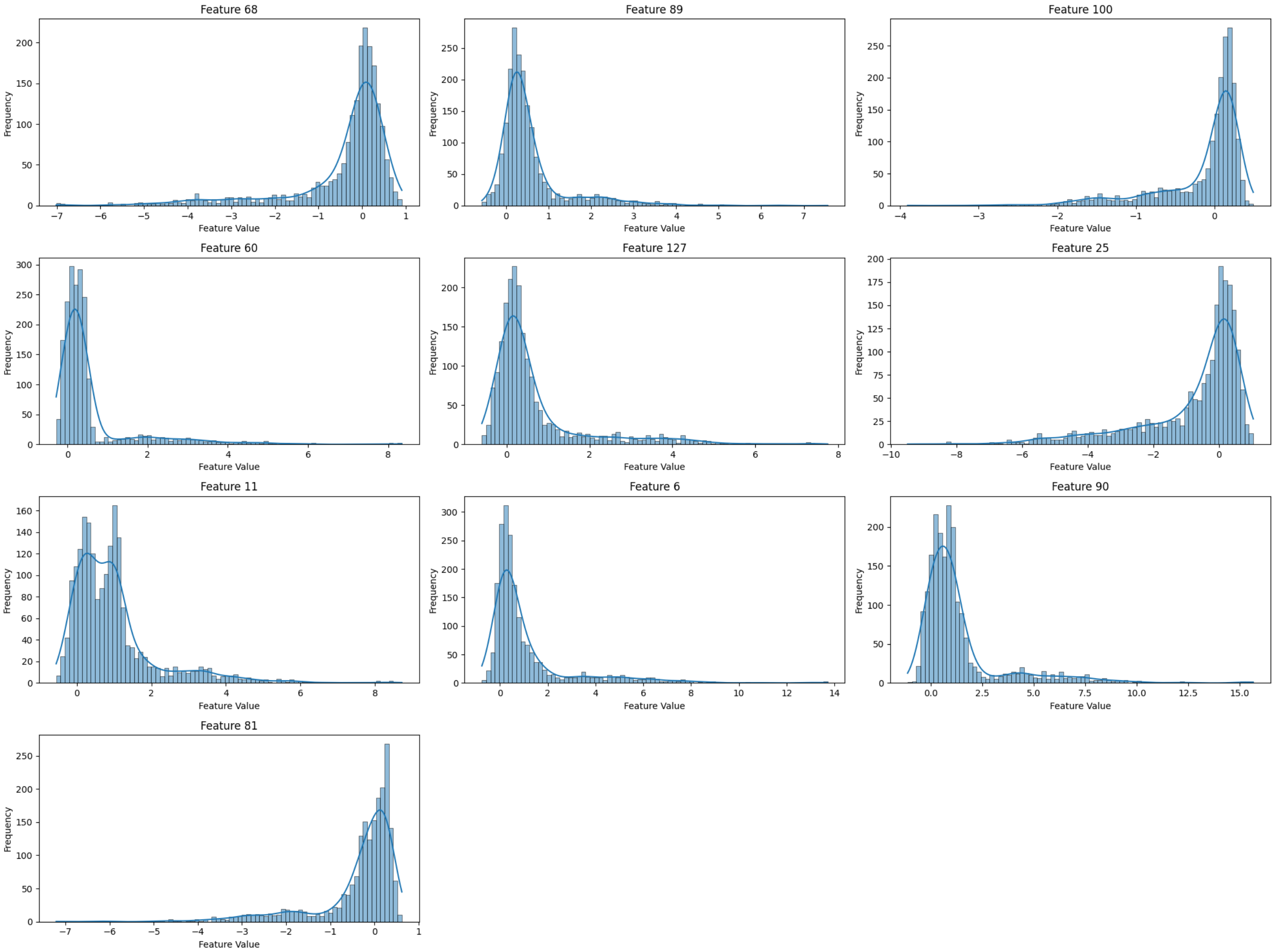


Figure 16. Histogram of feature value distributions for principal component features

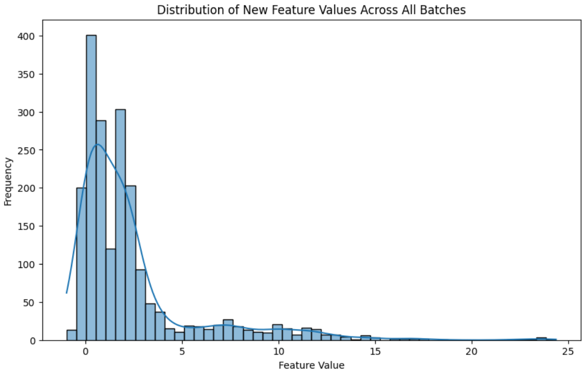


Figure 17. Histogram of feature value distributions for combination of feature 11 and 90

As can be seen in figure 16, many of the distributions for the important features had very long tails. Therefore, a new PCA was made after outliers had been removed. The new most important features were features 119, 48, 33, 19, 42, 90, 107, 91, 85 and 72. Histograms of their distributions are shown in figure 19. Most of these histograms have clear distinctions into two different distributions.

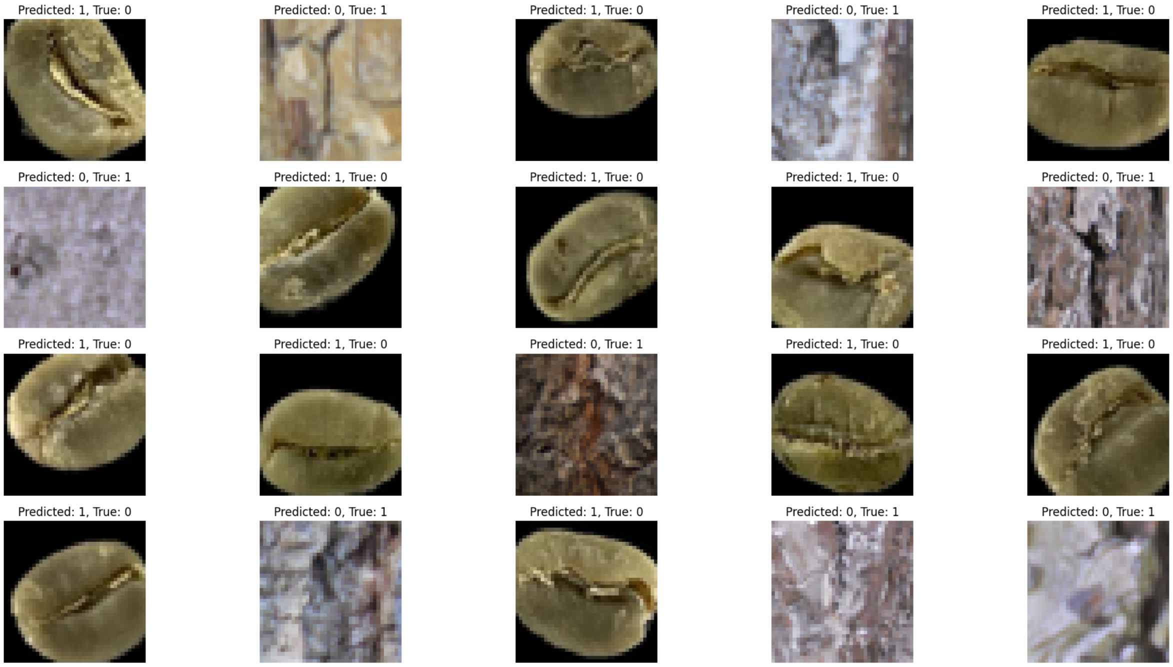


Figure 18. Missclassifications for feature 11 and 90 combined

The features that appeared to have distinct distributions were chosen and their thresholds were decided by studying the histograms. These features, their thresholds and accuracies are shown in table VII. Feature 42 should have had the sign swaped for deciding which label to give the image, but except for this, all features have accuracies above 90. The feature with the highest accuracy was feature 85.

Feature	Threshold	Accuracy
119	1.5	0.9095
33	1.0	0.9780
19	0.5	0.9795
42	-0.5	0.0315
107	1.0	0.9765
91	0.9	0.9640
85	1.0	0.9985
72	1.9	0.9755

Table VII
ACCURACY FOR FEATURES WITH CORRESPONDING THRESHOLDS AFTER REMOVAL OF OUTLIERS

Feature 85 only misclassified three instances and these are shown in figure 20. It is understandable that these images are hard to classify, the first one would be difficult for a human as well since it is very dark.

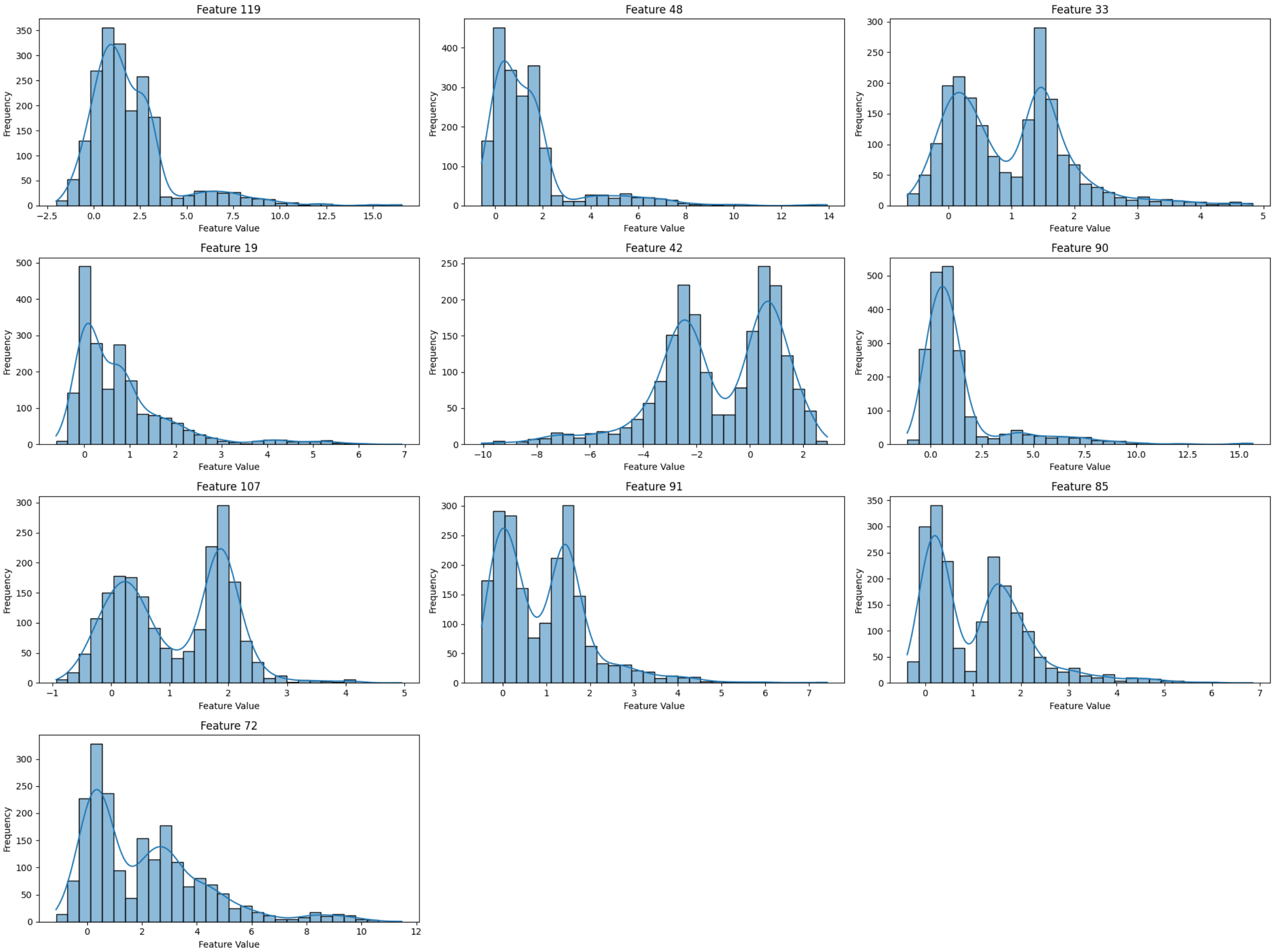


Figure 19. Distributions for most important features when outliers were removed

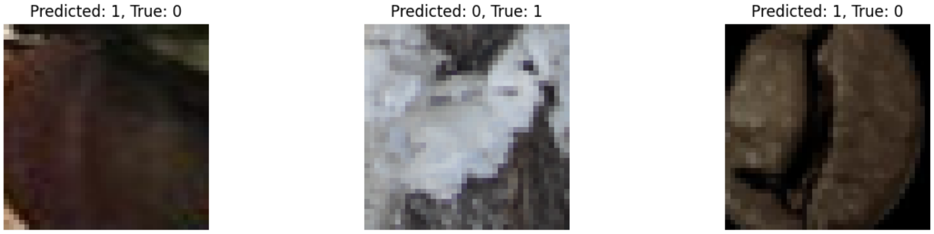


Figure 20. Miss classifications for feature 85

Once again, features 33, 85 and 107 were added together since these had similar threshold values. The new histogram for the combined feature is shown in figure 21. Two separated distributions can clearly be seen in this histogram. Using threshold 5 for this feature gave an accuracy of 100%, showing that this combined feature was perfect at separating the ID and OOD samples in this data.

References

[Bro19] Jason Brownlee. A gentle introduction to early stopping to avoid overtraining neural networks. August 6 2019.

[Mad22] Sarah Madeleine. Normalization, zero centering and standardization of ct images, January 21 2022. Updated on September 26, 2022.

[Man24] Manav Mandal. Introduction to convolutional neural networks (cnn). 2024. Accessed: 2024-02-23.

[Ola23] Juan C. Olamendy. Understanding relu, leakyrelu, and prelu: A comprehensive guide. December 4 2023. 6 min read.

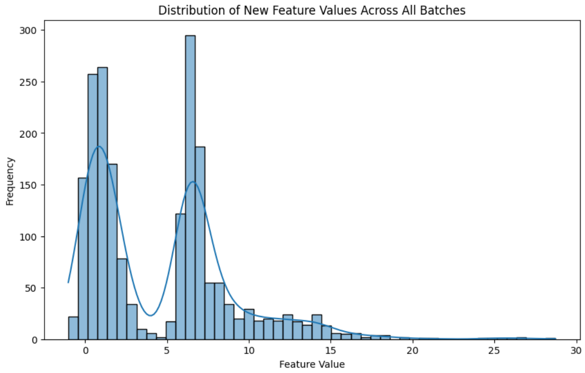


Figure 21. Histogram of distributions for feature 33, 85 and 107 added

[Sax24] Shipra Saxena. Introduction to batch normalization. February 20 2024.

[T23] Sivaram T. What are skip connections in deep learning?, August 14 2023. 7 min read.