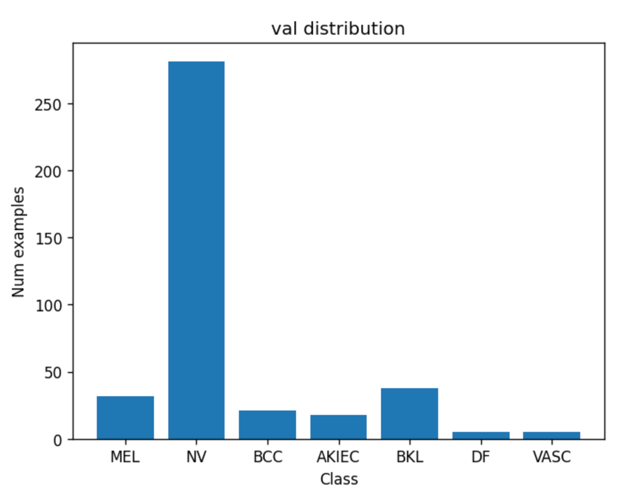
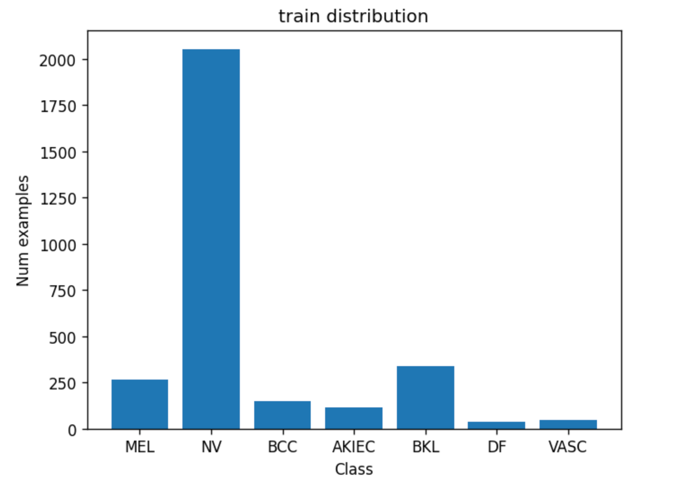
# Check for data issues

There is a class imbalance issue, as shown in the distribution plots below. In both the training and validation datasets, the ‘NV’ class is being overrepresented while the remaining classes are being underrepresented. This usually results in a model that performs poorly on the minority classes.

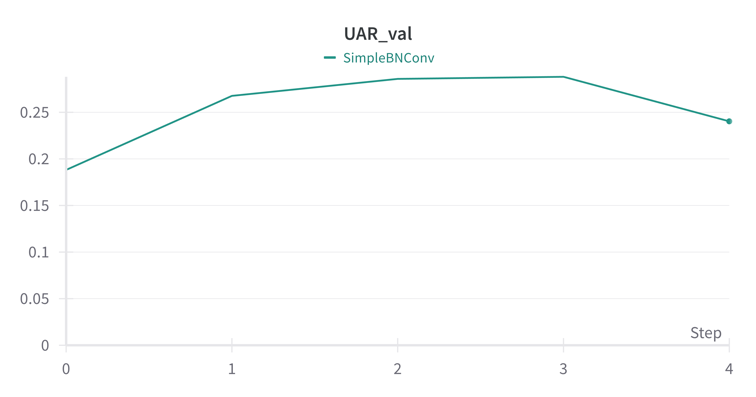


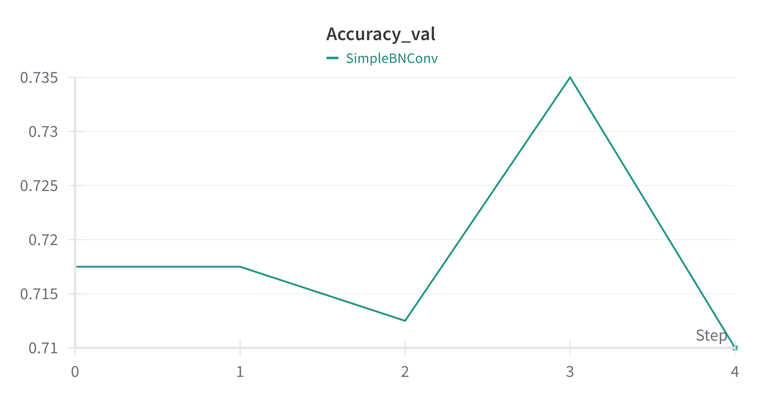


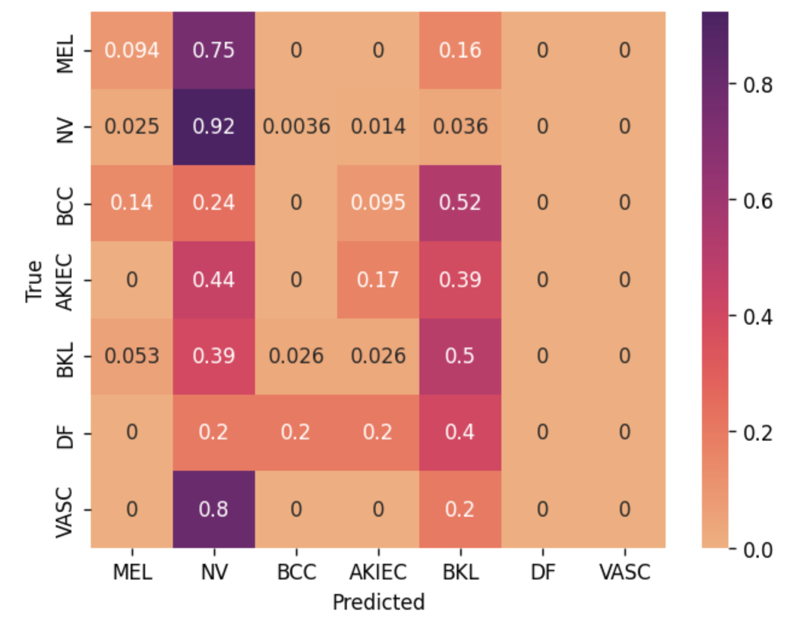
## We also checked for missing values and there are none in this dataset.

# Implement a baseline convolutional neural network

While the baseline CNN model achieves a 71% accuracy, its UAR is very low at around 24%. This could be attributed to the class imbalance issue. As ‘NV’ accounts for 70% of the examples in the validation dataset, simply classifying all instances as ‘NV’ could yield a misleadingly high accuracy of 70%. In fact, this imbalance leads the model to misclassify numerous instances into the majority class ‘NV’, as shown in the confusion matrix. For example, 75% of “MEL” records were incorrectly classified as “NV”, which is not good because MEL (melanoma) is a type of skin cancer while NV (melanocytic nevus) is just a benign, non-cancerous mole.





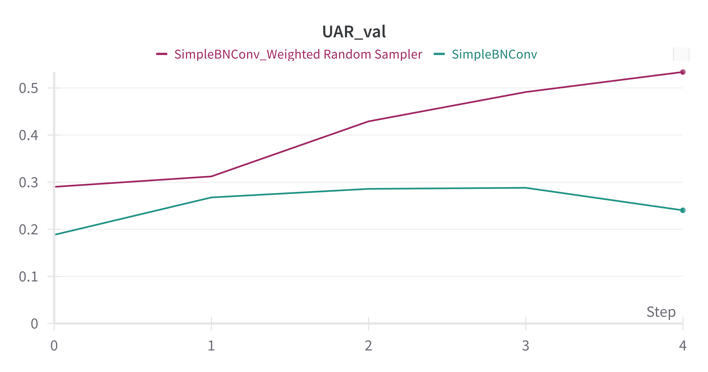


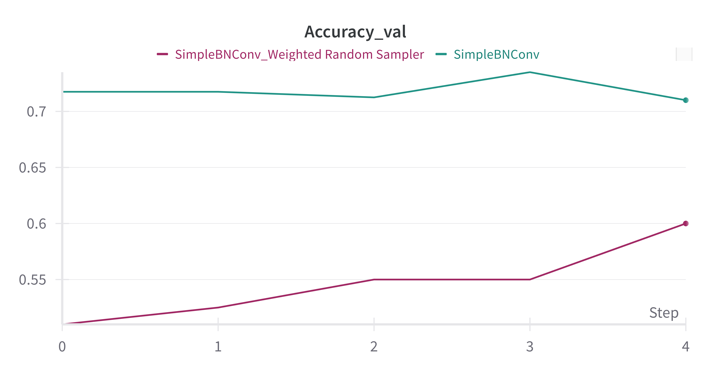
# Account for data issues

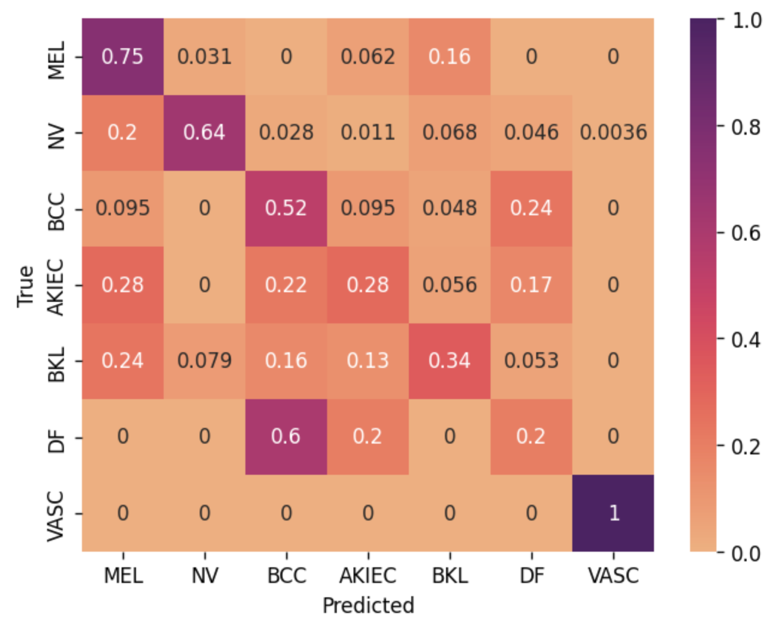
We tried two techniques to address the class imbalance issue: Weighted Random Sampler and Weighted Loss Function. Both methods appeared to solve this problem as they improved validation UAR and decreased the percentage of other classes being incorrectly classified as the majority class ‘NV’. However, they also resulted in lower validation accuracy.

**Weighted Random Sampler**

The Weighted Random Sampler technique significantly improved Validation UAR (from 24% to 53%). As shown in the new confusion matrix, the percentage of images being misclassified as the majority class (NV) greatly reduced. For example, only 3% of “MEL” images were misclassified as “NV”, compared to 75% before using the Weighted Random Sampler. However, validation accuracy decreased from 71% to 60%.

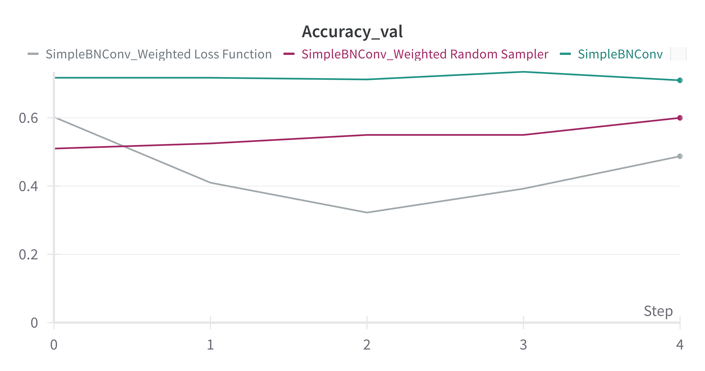
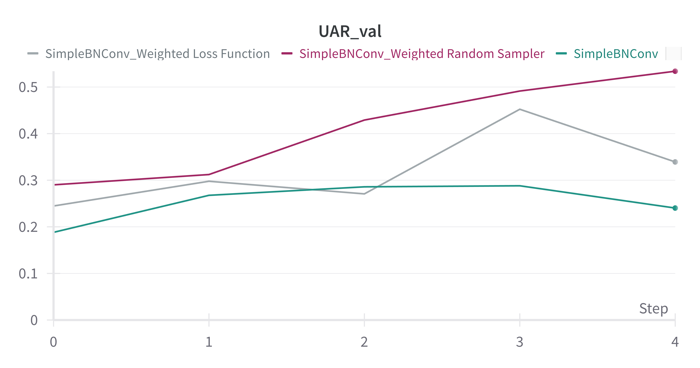


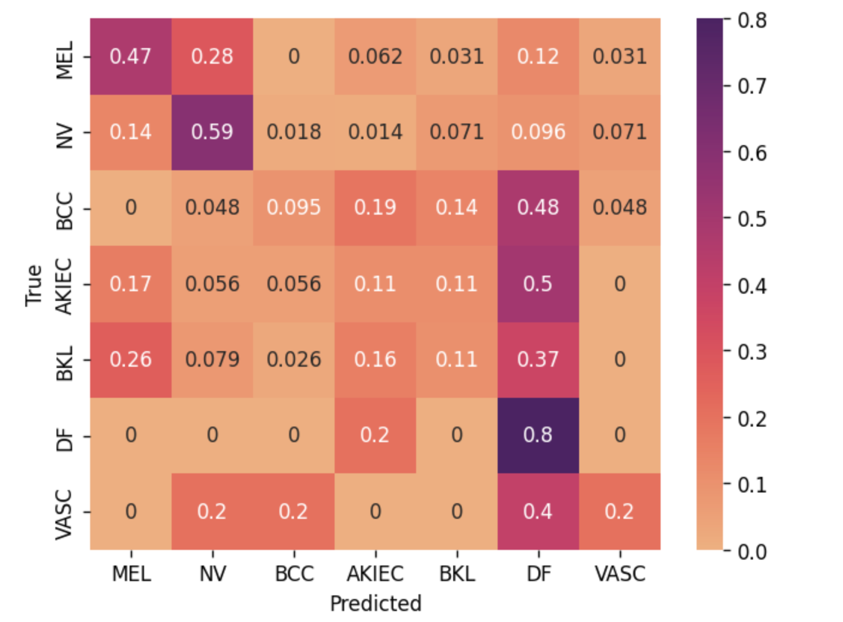




**Weighted Loss Function**

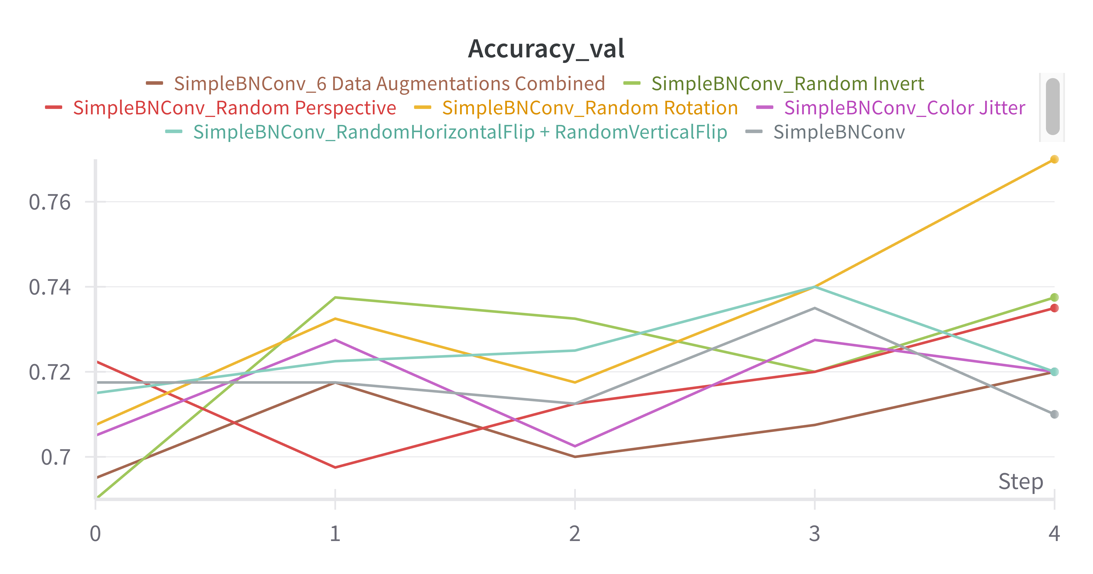
The Weighted Loss Function underperformed compared to the Weighted Random Sampler in both accuracy and UAR. However, it still improved the validation UAR by almost 10%, from 24% to approximately 34%.

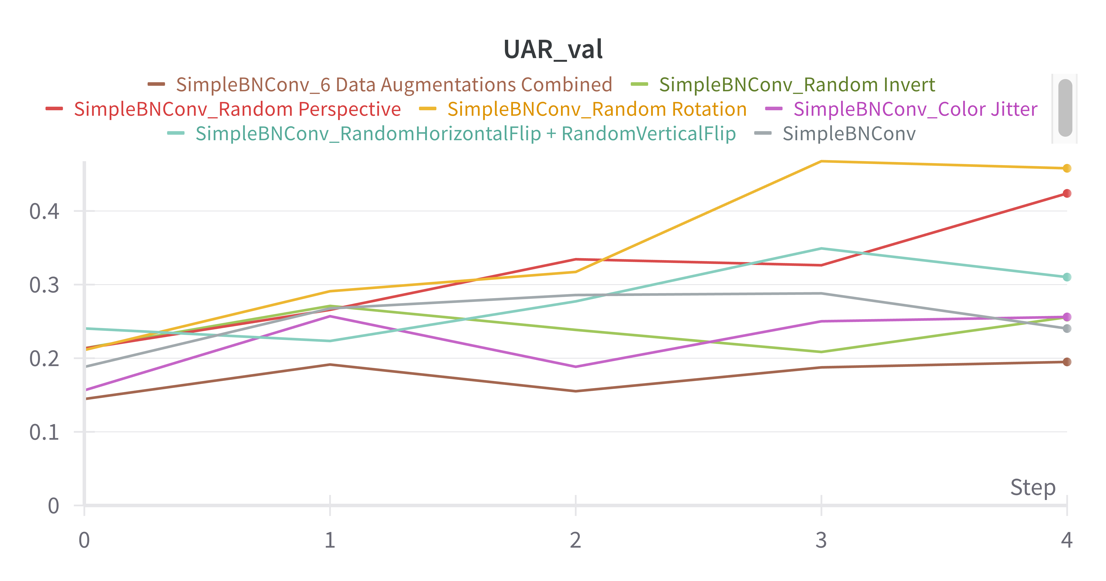




# Data Augmentation

Data augmentations generally improved validation accuracy, with most techniques also boosting validation UAR. However, combining six data augmentations actually resulted in lower validation UAR. Random Rotation proved particularly effective for this dataset, nearly doubling the validation UAR and increasing validation accuracy by 6%, whereas other techniques resulted in only a 1-2% improvement in accuracy.





# Chase Improved Performance

Firstly, we tried to improve the performance of the baseline model by increasing its depth (by adding more layers) and width (by increasing the number of output feature maps per convolutional layer). This new model is called the EnhancedBNConv model. If we use the same hyperparameters, EnhancedBNConv would actually lead to lower accuracy and UAR compared to the baseline model (Figure 1f.1)

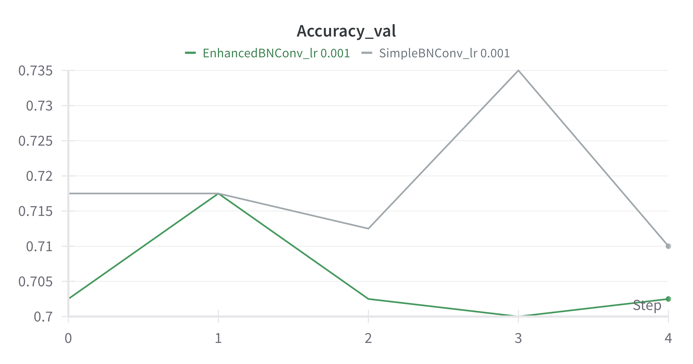
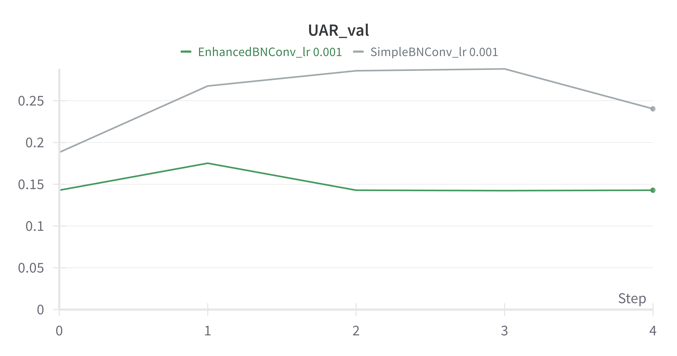
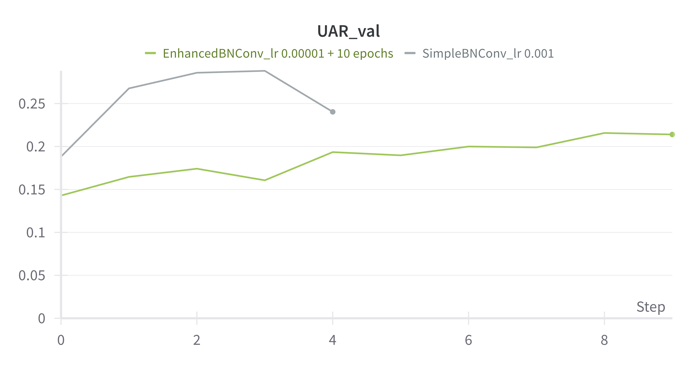


Figure 1f.1. EnhancedBNConv compared to baseline model

We found that the optimal learning rate for EnhancedBNConv is 0.00001, and we should train the model for 10 epochs. This helped increase the validation accuracy slightly compared to the baseline model (from 71% to approximately 73%). However, validation UAR is still lower than that of the baseline model (Figure 1f.2).



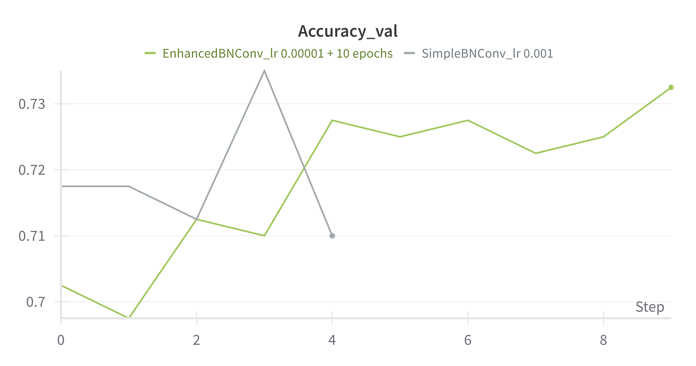


Figure 1f.2. Tuning hyperparameters for EnhancedBNConv

We also tried different activation functions with the baseline model, namely LeakyReLU, ELU, and GELU. However, none of them appeared to be particularly effective in improving the model performance (Figure 1f.3).

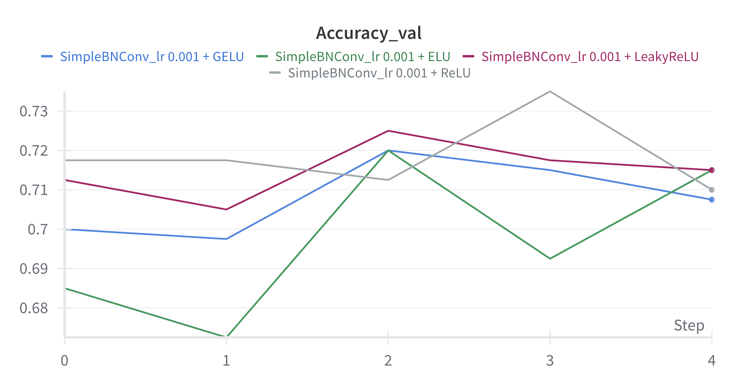
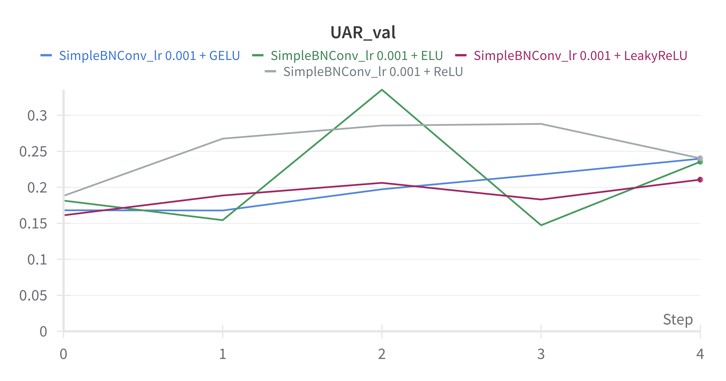


Figure 1f.3. Experiment with different activation functions

Transfer learning using the Resnet152 model (fine tune all the weights) seems to be the most effective technique. We were able to achieve a validation accuracy of 79% (8% higher than that of the baseline model) and validation UAR of approximately 45% (almost double that of the baseline model) (Figure 1f.4).

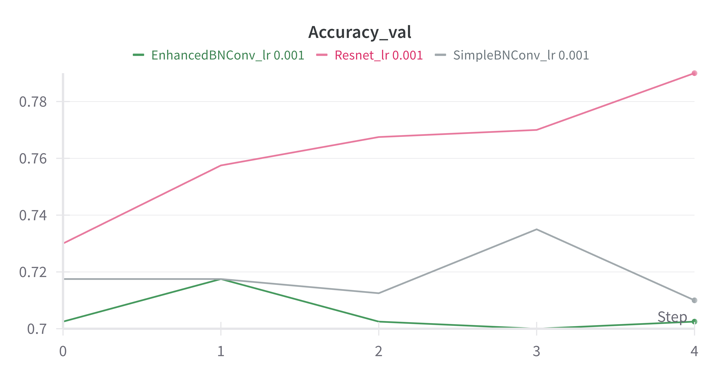
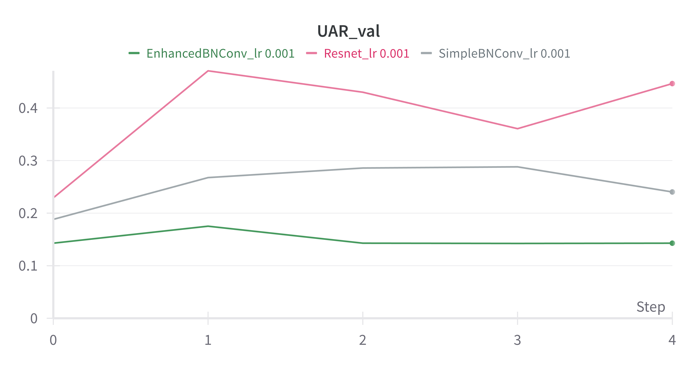


Figure 1f.4. Transfer learning using Resnet152

However, the validation loss curve of the Resnet model showed significant fluctuations (Figure 1f.5), which indicates that the learning rate of 0.001 might have been too high. In addition, it looks like the model has not converged yet as both training and validation loss were still decreasing (Figure 1f.5). Therefore, we tried decreasing the learning rate and increasing the number of training epochs.

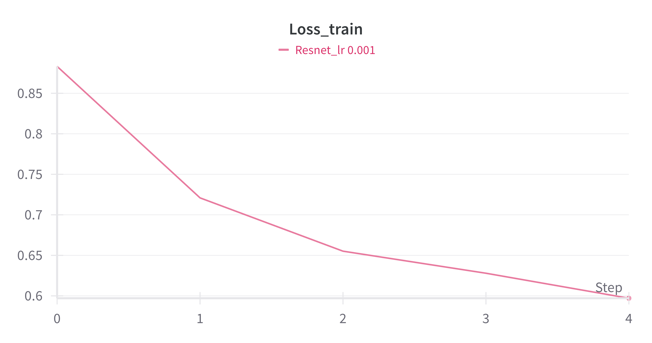


Figure 1f.5. Loss curves of Resnet152 model

We found that the optimal learning rate and number of training epochs for the Resnet model is 0.0001 and 10, respectively. With these parameters, we can achieve a validation accuracy of approximately 86% and a validation UAR of approximately 70%. We tried decreasing the batch size from 64 to 32 and trained for 15 and 20 epochs to address the overfitting problem, but it was not effective.

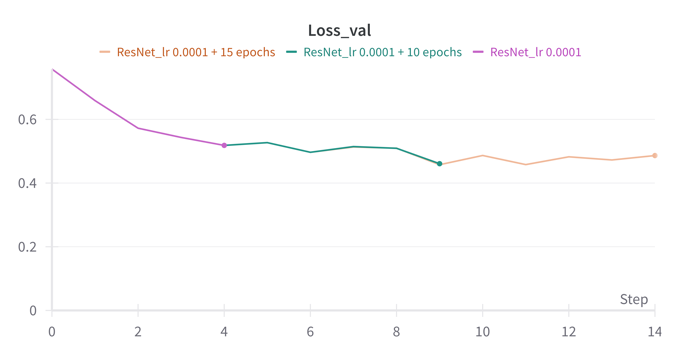
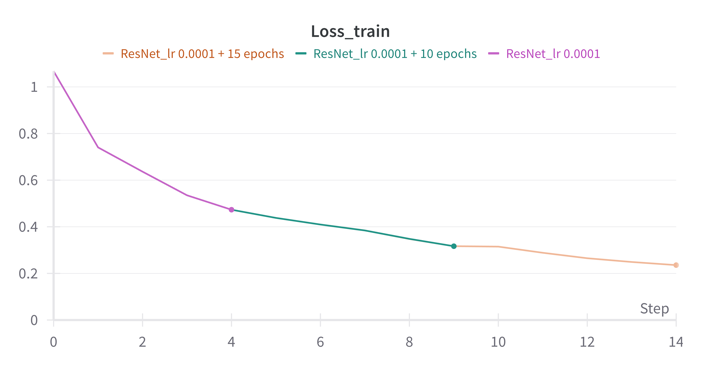
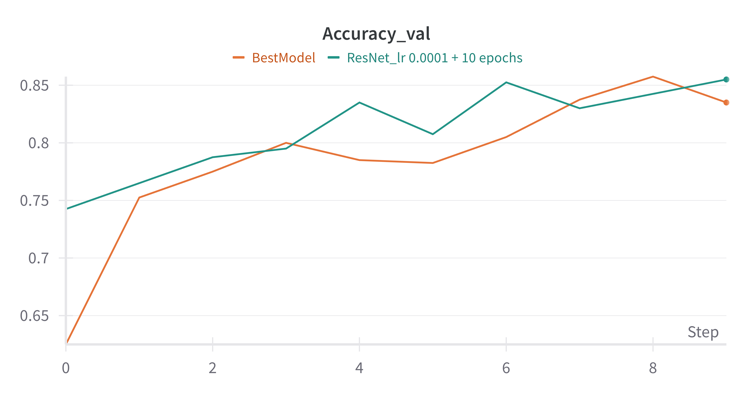
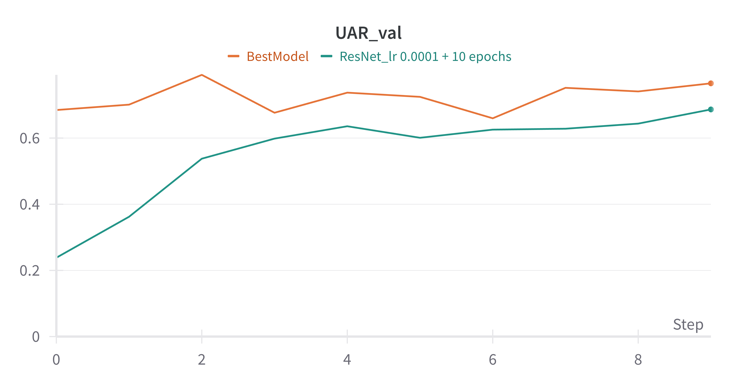


Figure 1f.6. Resnet152 started to overfit after 10 epochs

Finally, we combined the Resnet model with all the best performing techniques and hyperparameters that we had tried earlier. This new model slightly decreased validation accuracy (from 85.5% to 83.5%), however, it was able to increase validation UAR by approximately 8% (from 68.66% to 76.55%).



In summary, we were able to achieve the highest validation accuracy of 85.5% with the fine-tuned Resnet152 model, and highest validation UAR of 76.55% with the final model which combined fine-tuned Resnet152 with the best performing techniques and hyperparameters that we tried, including Random Rotation and Weighted Random Sampler.