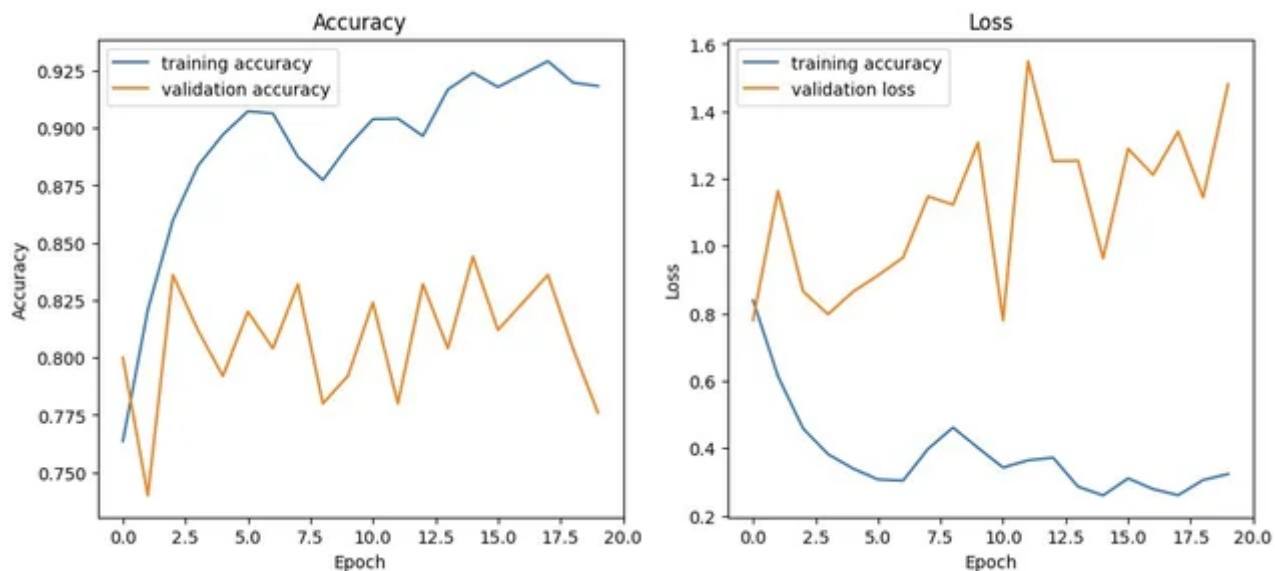


Attempt 1: Barebones

Trained a model on the dataset as is, without any changes.



The learning curve is staggering and not consistent. This could be caused by:

1. The learning rate is incorrect
2. The data split between learning and validation is not split properly

This in turn tells us the model is memorizing the data and not generalising. To fix this:

1. Tune learning rate, starting from a lower value
2. Reconsider the data split
3. Simplify problem and start there

Current learning rate: 0.001

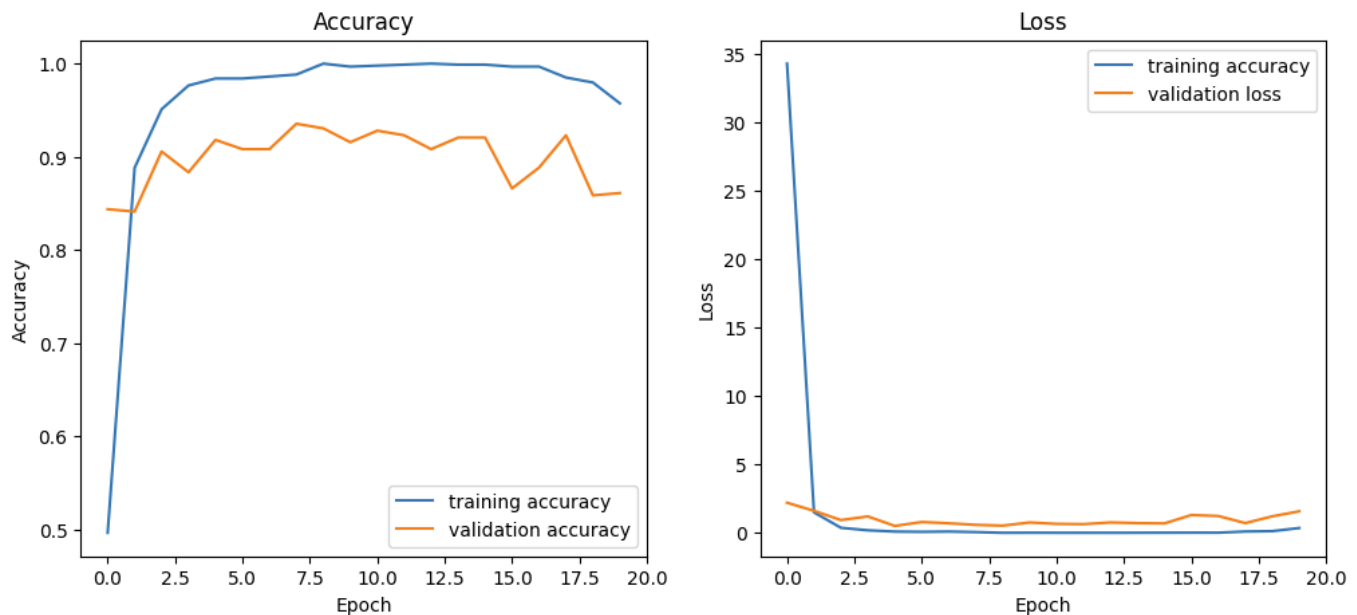
Data split: 97:3 train/val

Classes: 50

For the next attempt, we reconsider the data split to be 70:30 for train/test and then a further 70:30 for train/val. We also reduce the number of classes and check if it helps.

Attempt 2: Data Split & Class Reduction

For this attempt, we've reduced the classes from 50 to 10 and reconsidered the data split into: 49% / 21% / 30%



Now the model's learning curve has stabilized but around epoch 15, it starts going down meaning the model has started to memorize from that point onwards. The area between the learning and validation curves has also reduced meaning the model has started to generalize better.

The spikes mean that at that epoch, the data could be similar to the training data, which is why the model performed better just for that specific scenario.

Final Training Accuracy: 95.74%

Final Validation Accuracy: 86.10%

Best Epoch: 10

From here, we know learning rate was not the issue as with the same learning rate, the model started to generalize. This model currently is overfitting and to prevent that, we can optimize the dropout layer further.

Attempt 3: Dropout Layer Optimization & Class Increase

In this attempt, we increase the classes back to the original 50 and change the data split to not include validation anymore. We know the model is overfitting and we need to fix that. We can do that by optimizing the dropout layer.

The current dropout value is 0.2 but using a HP tuning, we check the values:

[0.1, 0.15, 0.2, 0.25, 0.3]

We're also reducing the samples from 8.7K down to 2.5K and splitting that 70/30 for train/test. Looking at the previous attempt, anything more than 13 epochs is where the model starts memorizing. For the remaining phases, we've reduced it down to 10 epochs.

To identify which classes are being mislabelled, we generate confusion matrix & classification report to see where the model is mixing classes up.

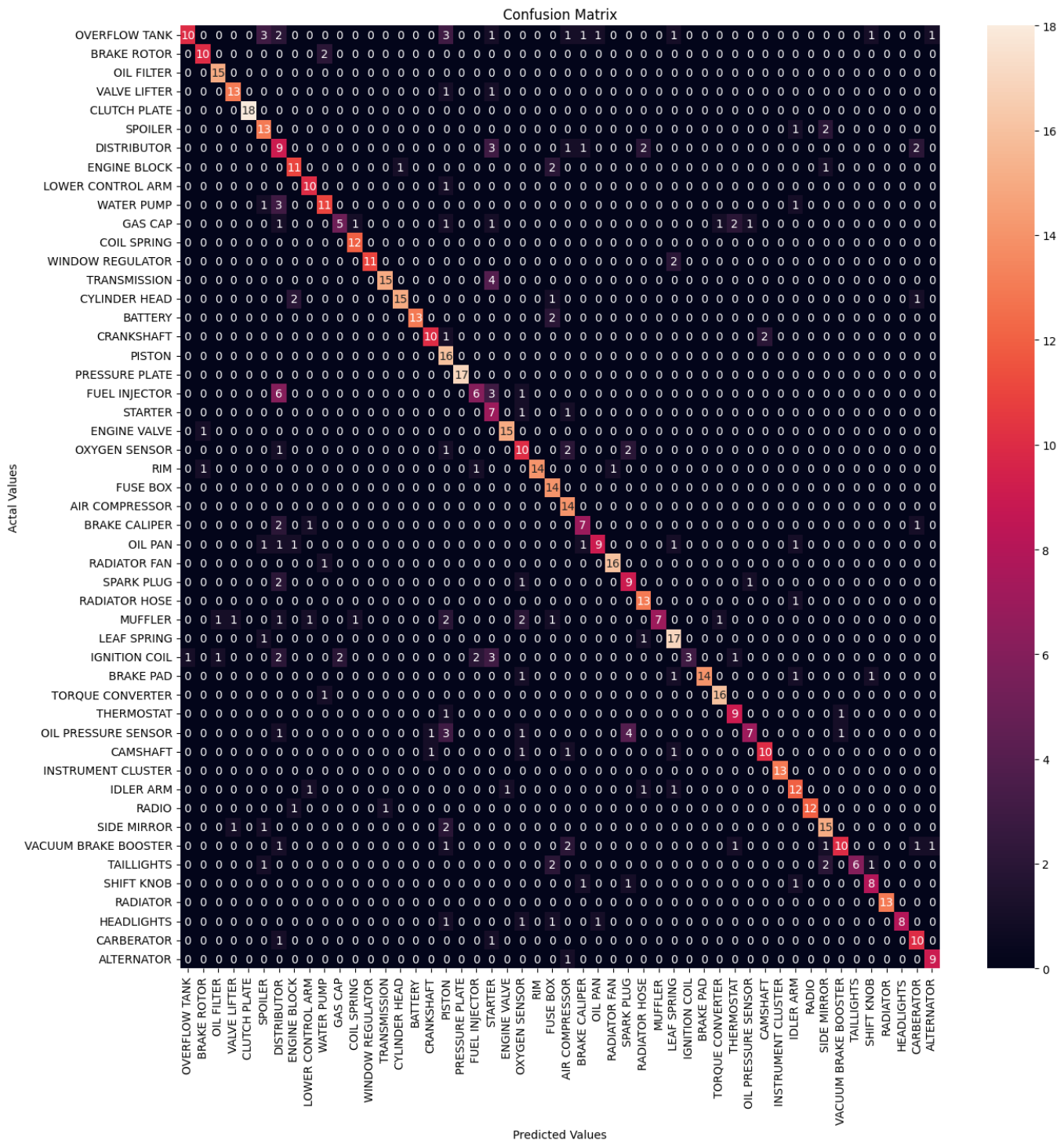
From just a quick scan, we can see that Water Pump, Vacuum Brake Booster and Gas Cap are all being mislabeled. This makes sense as they all, at least to the model, would look similar.



From here, we can remove the 5 worst performing classes and check the accuracy again.

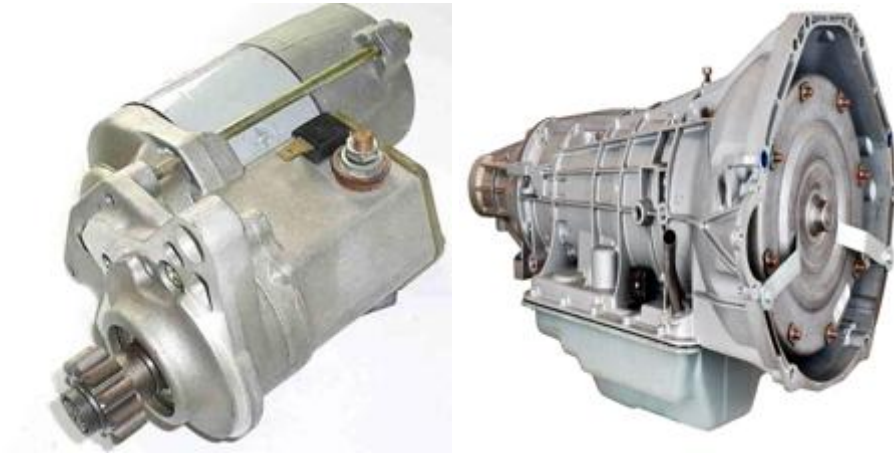
Changelog:

- Reduced sample count from 8.7k to 2.5K
- Reduced Epoch count to 10



So looking from this (limiting to only 5 class analysis because I am not doing 50):

- Transmission and Starter



- Fuel Injector and Distributor



- Oil Pressure Sensor and Spark Plug

