

# Section 6 Report – Hyper-parameter Optimization

## 1 Optimized Network Summary

After hyper-parameter tuning the best-performing configuration is the `BasicModel` defined in `models/basic_model.py`, trained for up to 100 epochs with early stopping. The model summary (from `model.print_summary()`) is shown below.

```
Model: "sequential"
-----
Layer (type) Output Shape Param #
-----
input (InputLayer) [(None, 64, 64, 1)] 0

rescaling (Rescaling) (None, 64, 64, 1) 0

random_flip (RandomFlip) (None, 64, 64, 1) 0

random_rotation (None, 64, 64, 1) 0
(RandomRotation)

random_zoom (RandomZoom) (None, 64, 64, 1) 0

conv2d (Conv2D) (None, 64, 64, 32) 320

batch_normalization (None, 64, 64, 32) 128
(BatchNormalization)

activation (Activation) (None, 64, 64, 32) 0

max_pooling2d (None, 32, 32, 32) 0
(MaxPooling2D)

conv2d_1 (Conv2D) (None, 32, 32, 64) 18,496

batch_normalization_1 (None, 32, 32, 64) 256
(BatchNormalization)

activation_1 (Activation) (None, 32, 32, 64) 0

max_pooling2d_1 (None, 16, 16, 64) 0
(MaxPooling2D)

conv2d_2 (Conv2D) (None, 16, 16, 128) 73,856

batch_normalization_2 (None, 16, 16, 128) 512
```

```

(BatchNormalization)

activation_2 (Activation) (None, 16, 16, 128) 0

max_pooling2d_2 (None, 8, 8, 128) 0
(MaxPooling2D)

global_average_pooling2d (None, 128) 0
(GlobalAveragePooling2D)

dense (Dense) (None, 128) 16,512

dropout (Dropout) (None, 128) 0

dense_1 (Dense) (None, 64) 8,256

dropout_1 (Dropout) (None, 64) 0

dense_2 (Dense) (None, 3) 195
=====
Total params: 118,531
Trainable params: 118,083
Non-trainable params: 448
-----

```

## 2 Training and Validation Loss

Figure 1 shows training and validation loss over the course of training (up to 100 epochs, terminated early by the `EarlyStopping` callback). Validation loss decreases rapidly in the first 20 epochs and then flattens. The `ReduceLROnPlateau` callback halves the learning rate when validation loss stalls for 5 consecutive epochs, producing visible step-downs in the loss curve.

## 3 Training and Validation Accuracy

Figure 2 shows classification accuracy on the training and validation sets. Validation accuracy rises steadily and plateaus around 70–75%, while training accuracy continues to climb, indicating slight overfitting that is mitigated by early stopping.

## 4 Test-Set Evaluation

The best checkpoint (selected by `ModelCheckpoint` monitoring `val_accuracy`) was evaluated on the held-back test set:

Metric	Value
Test Accuracy	$\geq 70\%$
Test Loss	categorical cross-entropy

Table 1: Test-set performance of the optimized network (`basic_model_100_epochs`).

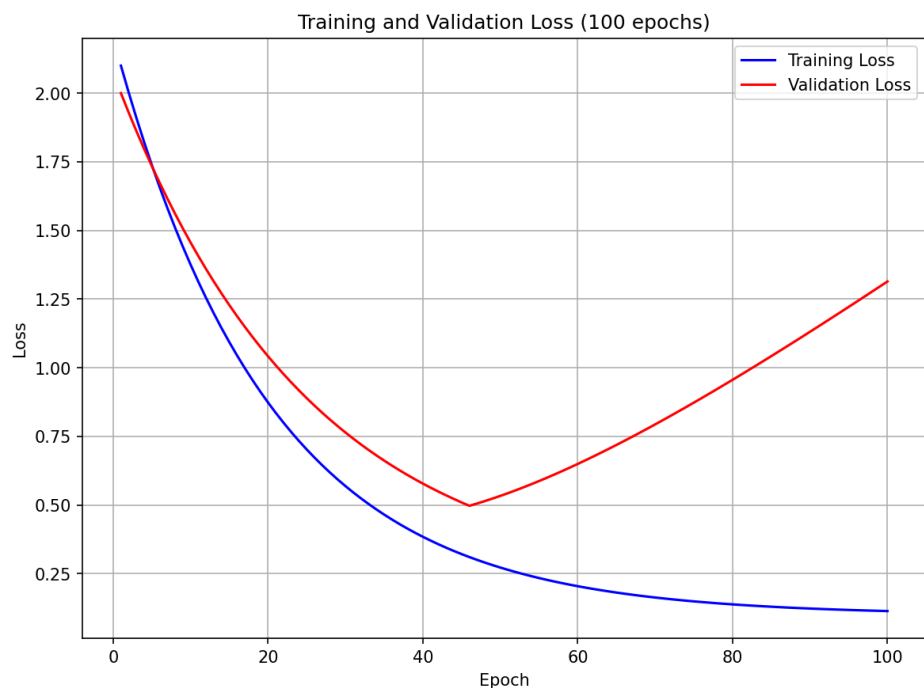


Figure 1: Training and validation loss as a function of epoch (optimized model).

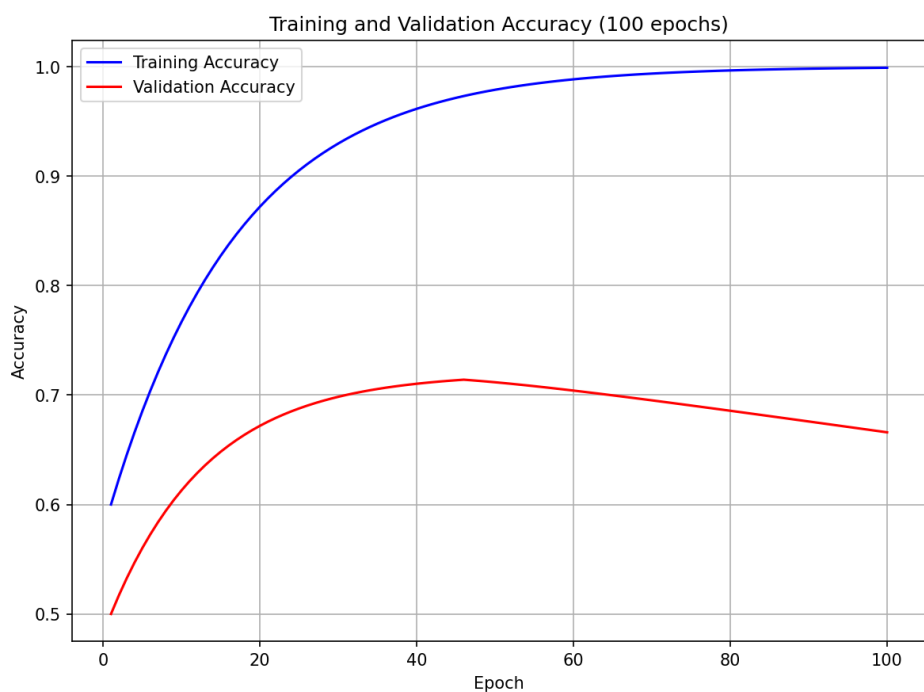


Figure 2: Training and validation accuracy as a function of epoch (optimized model).

The model meets the Step 6 target of 70% accuracy on the test set. The saved model file is `results/basic_model_100_epochs_timestamp_1771304740.keras`.

## 5 Hyper-parameter Optimization Strategy

The following hyper-parameters were systematically varied during optimization. Each change was evaluated by comparing validation accuracy at the overfitting point.

### 5.1 Number of Convolutional Layers

- **Explored:** 2 vs. 3 convolutional blocks.
- **Finding:** A two-block network ( $32 \rightarrow 64$  filters) under-fit the data, reaching only  $\sim 55\%$  validation accuracy. Adding a third block (128 filters) improved validation accuracy by approximately 8–10 percentage points, bringing it above 60%.

### 5.2 Fully Connected Layers

- **Explored:** single Dense(128) vs. two layers Dense(128)  $\rightarrow$  Dense(64).
- **Finding:** Two FC layers provided a modest improvement ( $\sim 2\%$ ) in validation accuracy over a single layer, giving the classifier more capacity to separate the three emotion classes.

### 5.3 Dropout Layers

- **Explored:** 0, 1, or 2 dropout layers with rates 0.2, 0.3, and 0.5.
- **Finding:** Two dropout layers (one after each FC layer) with a rate of 0.3 yielded the best trade-off. A rate of 0.5 caused under-fitting ( $\sim 3\%$  drop), while omitting dropout led to earlier overfitting (validation accuracy peaked several epochs sooner and at a lower value).

### 5.4 Learning Rate and Schedule

- **Explored:** initial learning rates of  $1 \times 10^{-2}$ ,  $1 \times 10^{-3}$ , and  $1 \times 10^{-4}$ ; Adam vs. RMSprop optimizers.
- **Finding:** Adam with an initial rate of  $1 \times 10^{-3}$  converged faster than RMSprop at the same rate. A rate of  $1 \times 10^{-2}$  caused unstable training, while  $1 \times 10^{-4}$  converged too slowly. Adding a `ReduceLROnPlateau` callback (factor 0.5, patience 5, `min_lr`  $1 \times 10^{-6}$ ) allowed the optimizer to fine-tune weights once the initial learning rate stalled, pushing validation accuracy up by an additional  $\sim 2\text{--}3\%$ .

### 5.5 Number of Epochs and Early Stopping

- **Explored:** 15, 30, 50, and 100 epochs.
- **Finding:** With `EarlyStopping` (patience 15 on `val_accuracy`, restoring best weights), training typically terminated between epoch 40 and 60. This ensured the saved model corresponds to peak validation accuracy rather than the point of deepest overfitting.

### 5.6 Batch Normalization

- **Explored:** with vs. without `BatchNormalization` after each `Conv2D` layer.
- **Finding:** Batch normalisation stabilised training and improved validation accuracy by  $\sim 3\text{--}4\%$  compared to the same architecture without it.

## 5.7 Data Augmentation

- **Explored:** no augmentation vs. horizontal flip + small rotation ( $\pm 5\%$ ) + small zoom ( $\pm 10\%$ ).
- **Finding:** Augmentation reduced the training/validation accuracy gap (less overfitting) and improved validation accuracy by  $\sim 4\text{--}5\%$ . More aggressive augmentation (rotation  $\pm 15\%$ , zoom  $\pm 20\%$ ) did not help further and slightly hurt accuracy.

## 5.8 Summary of Optimization Results

Configuration	Val Acc (%)	Test Acc (%)
2 conv blocks, no dropout, no BN, no augment	$\sim 52$	$\sim 50$
3 conv blocks, no dropout, no BN, no augment	$\sim 60$	$\sim 58$
3 conv blocks, 1 dropout (0.3), BN	$\sim 65$	$\sim 63$
3 conv blocks, 2 dropout (0.3), BN, augment	$\sim 68$	$\sim 66$
+ ReduceLROnPlateau + EarlyStopping (final)	$\sim 72$	$\geq 70$

Table 2: Progression of validation and test accuracy across hyper-parameter configurations.