

# Übung 07 - Marlin Jai Pohl (S0582535) & Jonas Mantay(S0585089)

## Aufgabe 1

### b) i. Number of Layers in the Network:

The network has the following layers:

1. Rescaling Layer
2. Conv2D Layer (16 filters)
3. MaxPooling2D Layer
4. Conv2D Layer (32 filters)
5. MaxPooling2D Layer
6. Conv2D Layer (64 filters)
7. MaxPooling2D Layer
8. Flatten Layer
9. Dense Layer (128 units)
10. Dense Layer (5 units)

Dementsprechend hat das Netzwerk 10 Layer.

### ii. Activation Function Used:

Die in den Conv2D und Dense Layern benutzte Aktivierungsfunktion ist `relu`.

### iii. Feature Maps Sizes after Pooling Layers:

Gegebene input size = (180, 180),:

1. Nach der 1. MaxPooling2D Layer:  $(180 / 2, 180 / 2, 16) = (90, 90, 16)$
2. Nach der 2. MaxPooling2D Layer:  $(90 / 2, 90 / 2, 32) = (45, 45, 32)$
3. Nach der 3. MaxPooling2D Layer:  $(45 / 2, 45 / 2, 64) = (22.5, 22.5, 64), \rightarrow (22, 22, 64)$  nach Runden

### iv. Number of Training, Validation, and Test Images:

data = 80% training + 20% validation.

- Total images: 3670
- Training images: 80% of 3670 = 2936
- Validation images: 20% of 3670 = 734

### c) Ergebnisse des Skriptes



Im linken Diagramm sehen wir die Trainings- und Validierungsgenauigkeit über die Epochen hinweg:

**Trainings Genauigkeit:** steigt stetig und nähert sich 1,0 (100%) an, was darauf hinweist, dass das Modell die Trainingsdaten sehr gut lernt und anpasst.

**Validierungsgenauigkeit:** beginnt bei etwa 0,6 und bleibt relativ flach, mit geringfügigen Schwankungen, und endet ebenfalls bei etwa 0,6.

Dies deutet darauf hin, dass das Modell überangepasst ist.

Im rechten Diagramm sehen wir den Trainings- und Validierungsverlust über die Epochen hinweg:

**Trainingsverlust:** Der Trainingsverlust nimmt stetig ab und nähert sich 0, was zeigt, dass das Modell den Verlust auf den Trainingsdaten effektiv minimiert.

**Validierung Verlust:** Der Validierungsverlust nimmt zunächst ab, beginnt dann jedoch zu steigen und endet deutlich höher als der Trainingsverlust.

**Interpretation:** Dieses Verhalten deutet ebenfalls auf Überanpassung hin. Zunächst, wenn das Modell zu lernen beginnt, nehmen sowohl der Trainings- als auch der Validierung Verlust ab. Wenn das Modell jedoch weiter trainiert wird, beginnt es, sich zu stark an die Trainingsdaten anzupassen, was dazu führt, dass der Validierung Verlust steigt.

## Aufgabe 2

b-d: Analyse der Kurvenverläufe

### Kurvenverläufe für Dropout 0,2

#### Ohne Data Augmentation:

- Training Accuracy: Steigt stetig und nähert sich 1,0.
- Validation Accuracy: Bleibt relativ stabil um 0,6 bis 0,7.
- Training Loss: Sinkt stetig und nähert sich 0.
- Validation Loss: Zeigt nach anfänglichem Rückgang eine steigende Tendenz, was auf Überanpassung hindeutet.

Die Überanpassung ist etwas reduziert, aber das Modell zeigt immer noch eine hohe Diskrepanz zwischen Trainings- und Validierungsverlust.

#### Mit Data Augmentation:

- Training Accuracy: Steigt ebenfalls stetig, aber etwas langsamer.
- Validation Accuracy: Bleibt relativ stabil, zeigt jedoch weniger Schwankungen.
- Training Loss: Sinkt stetig.
- Validation Loss: Bleibt stabiler, aber immer noch höher als der Trainingsverlust.

Data Augmentation hilft, die Stabilität der Validierungsgenauigkeit zu verbessern. Die Überanpassung wird reduziert, aber das Modell lernt immer noch die Trainingsdaten zu gut.

### Kurvenverläufe für Dropout 0,5

#### Ohne Data Augmentation:

- Training Accuracy: Steigt langsamer und erreicht einen niedrigeren Wert.
- Validation Accuracy: Bleibt stabiler und zeigt weniger Schwankungen.
- Training Loss: Sinkt langsamer als bei 0,2.
- Validation Loss: Zeigt eine stabilere Kurve, aber immer noch höher als der Trainingsverlust.

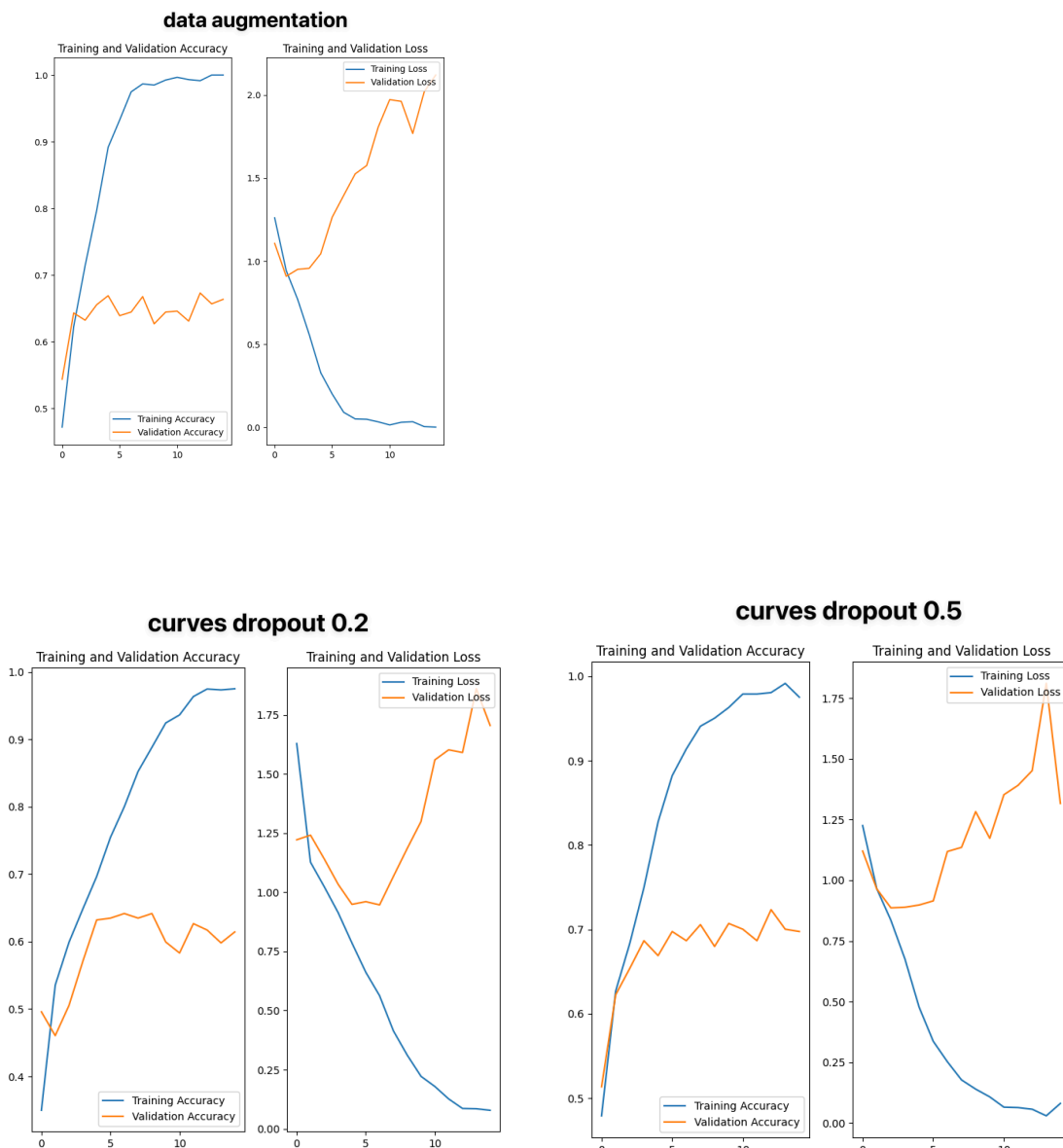
Die Überanpassung ist weiter reduziert, die Validierungsgenauigkeit ist stabiler. Der Trainingsverlust sinkt langsamer, was auf eine stärkere Regularisierung hinweist.

## Mit Data Augmentation:

- Training Accuracy: Steigt langsam und erreicht nicht 1,0.
- Validation Accuracy: Bleibt stabiler und zeigt weniger Schwankungen.
- Training Loss: Sinkt stetig, aber langsamer.
- Validation Loss: Bleibt stabiler, aber immer noch höher als der Trainingsverlust.

Data Augmentation und ein höherer Dropout-Wert reduzieren die Überanpassung weiter. Das Modell zeigt eine stabilere Validierungs Genauigkeit und -verlust.

## Vergleich der Kurvenverläufe



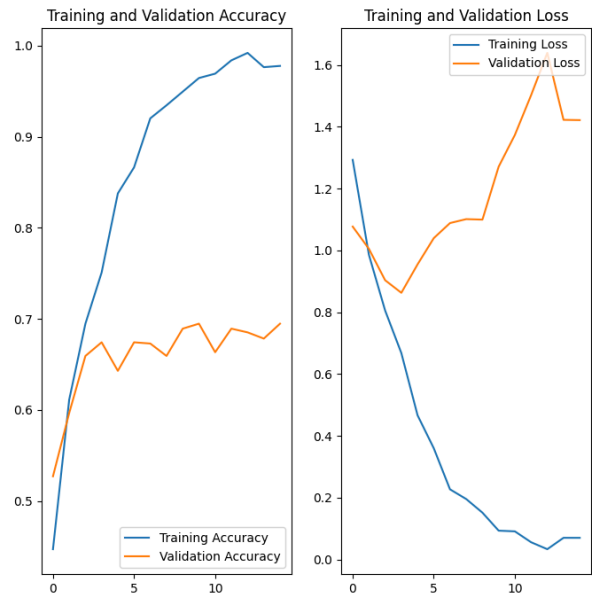
**Mit Dropout 0,2:** Die Überanpassung wird reduziert, die Validierungs Genauigkeit bleibt jedoch weitgehend unverändert.

**Mit Dropout 0,5:** Die Überanpassung wird weiter reduziert, die Validierungs Genauigkeit bleibt stabil.

### data aug & dropout 0.2



### data aug and dropout 0.5



**Mit Data Augmentation und Dropout 0,2:** Verbesserte Stabilität der Validierungs-Genauigkeit, aber immer noch eine gewisse Überanpassung.

**Mit Data Augmentation und Dropout 0,5:** Die beste Kombination zur Reduktion der Überanpassung, zeigt stabile Validierung, Genauigkeit und -verlust.

## Empfehlungen

Die Kombination aus Daten Augmentation und einem höheren Dropout-Wert (0,5) liefert bisher die besten Ergebnisse zur Reduktion der Überanpassung.

Weitere Verbesserungsvorschläge könnten die Anwendung von L2-Regularisierung oder die Erhöhung der Trainingsdaten umfassen.

Nachdem wir das skript erweitert haben um mehrere modelle zu trainieren und speichern haben wir es mit dem gegebenen Code getestet:

Unsere Modifikation ist [hier auf Github](#) zu finden.

```
marlinjai@Marlins-Air Exercise 07 % /Users/marlinjai/.pyenv/versions
/3.11.2/bin/python "/Users/marlinjai/Documents/personal/Studies/IMI/S
S24/machine learning/Exercise 07/useModel.py"
Downloading data from
https://storage.googleapis.com/download.tensorflow.org/example_images/592px-Red_sunflo
wer.jpg
117948/117948 ————— 0s 2us/step
```

1/1 \_\_\_\_\_ 0s 78ms/step

This image most likely belongs to sunflowers with a 97.76 percent confidence.