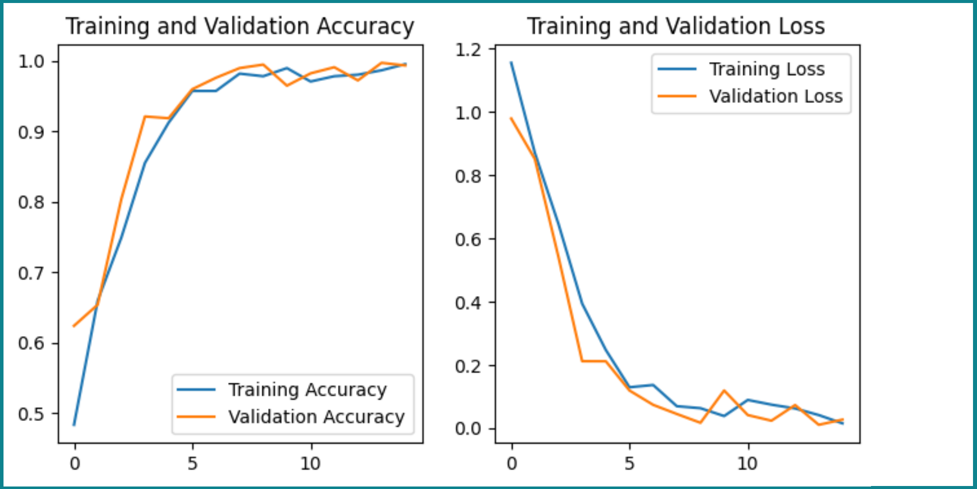
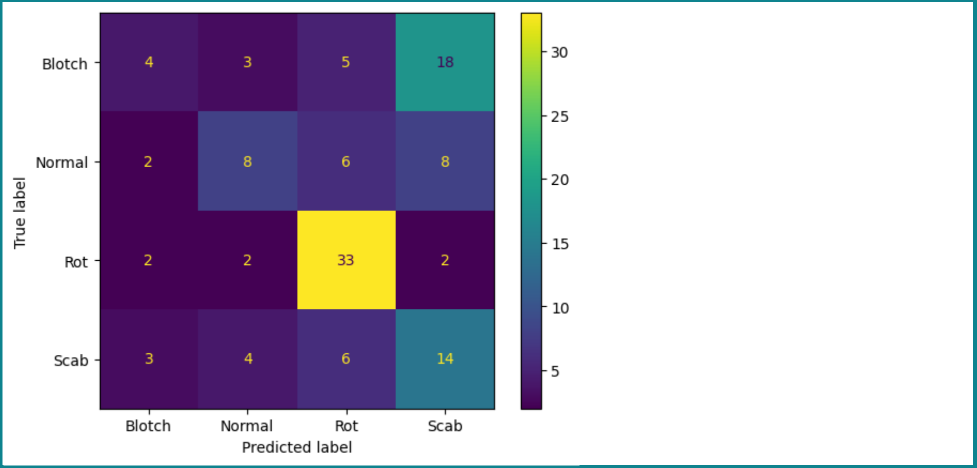
CNN (Apple Disease Classification) MODEL 1

```
model = models.Sequential([
             resize_and_rescale,
             layers.Conv2D(32, (3,3), activation='relu', input_shape=input_shape),
             layers.MaxPooling2D((2,2)),
             layers.Conv2D(64, (3,3), activation='relu'),
             layers.MaxPooling2D((2,2)),
             layers.Conv2D(64, (3,3), activation='relu'),
             layers.MaxPooling2D((2,2)),
             layers.Conv2D(128, (3,3), activation='relu'),
             layers.MaxPooling2D((2,2)),
             layers.Conv2D(128, (3,3), activation='relu'),
             layers.MaxPooling2D((2,2)),
             layers.Conv2D(256, (3,3), activation='relu'),
             layers.MaxPooling2D((2,2)),
             layers.Flatten(),
             layers.Dense(64, activation='relu'),
             layers.Dense(n_classes, activation='softmax')
             ])
```

Epochs = 15

image = 224x224
augmented_data (4x1000)
learning_rate = 0.001





	Blotch	Normal	Rot	Scab
Blotch	4	3	5	18
Normal	2	8	6	8
Rot	2	2	33	2
Scab	3	4	6	14

Confusion matrix 59/120 **0.4917**

Loss	0.0154	**Accuracy**	**0.9956**
Val_Loss	0.0271	**Val_Accuracy**	**0.9937**
Test_Loss	4.8243	**Test_Accuracy**	**0.4917**

COMMENTS

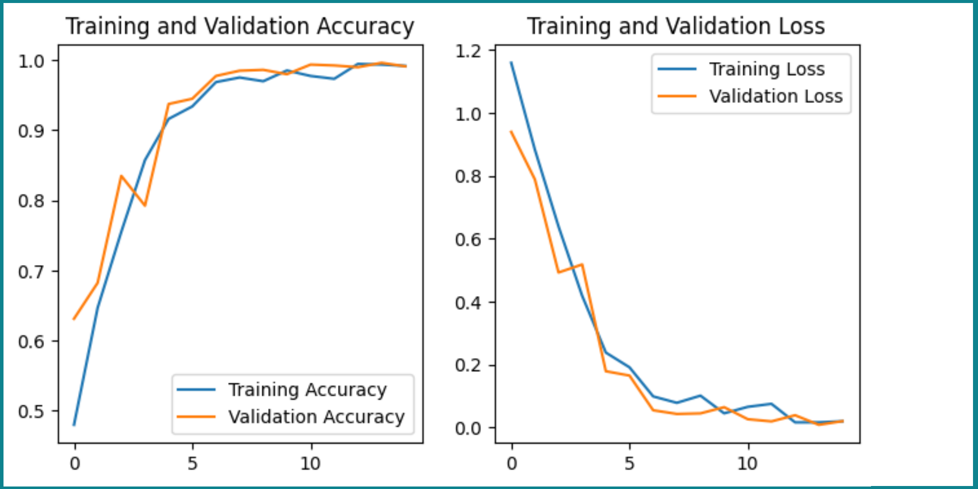
Ook met een dataset van 1000 afb. er categorie komt de basisopzet niet boven de 50% uit.
Minder dan de helft van de afbeeldingen wordt correct geclassificeerd.

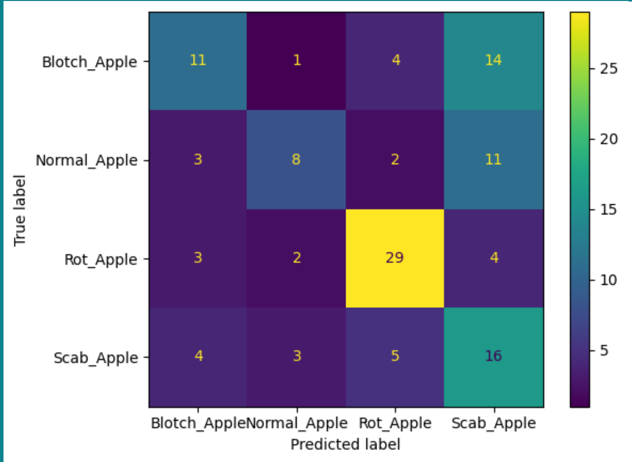
CNN (Apple Disease Classification) MODEL 2

```
model = models.Sequential([
            resize_and_rescale,
            layers.Conv2D(32, (3,3), activation='relu', input_shape=input_shape),
            layers.MaxPooling2D((2,2)),
            layers.Conv2D(64, (3,3), activation='relu'),
            layers.MaxPooling2D((2,2)),
            layers.Conv2D(64, (3,3), activation='relu'),
            layers.MaxPooling2D((2,2)),
            layers.Conv2D(128, (3,3), activation='relu'),
            layers.MaxPooling2D((2,2)),
            layers.Conv2D(128, (3,3), activation='relu'),
            layers.MaxPooling2D((2,2)),
            layers.Conv2D(256, (3,3), activation='relu'),
            layers.MaxPooling2D((2,2)),
            layers.Flatten(),
            layers.Dense(64, activation='relu'),
            layers.Dense(n_classes, activation='softmax')
            ])
```

Epochs = 15

image = 224x224
augmented_data (4x1000)
learning_rate = 0.001





	Blotch_Apple	Normal_Apple	Rot_Apple	Scab_Apple
Blotch_Apple	11	1	4	14
Normal_Apple	3	8	2	11
Rot_Apple	3	2	29	4
Scab_Apple	4	3	5	16

Confusion matrix 64/120 **0.5333**

COMMENTS

Zelfde condities als Model 1. Lichte verbetering van de Test_Accuracy.
Train en Validation Accuracies zijn iets minder als bij Model 1.

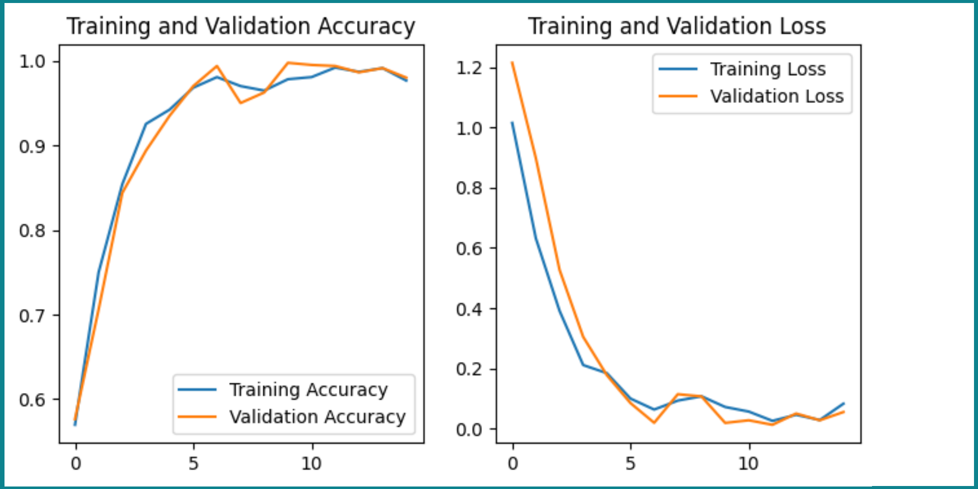
Loss	0.0202	**Accuracy**	**0.9919**
Val_Loss	0.0205	**Val_Accuracy**	**0.9912**
Test_Loss	---	**Test_Accuracy**	**0.5333**

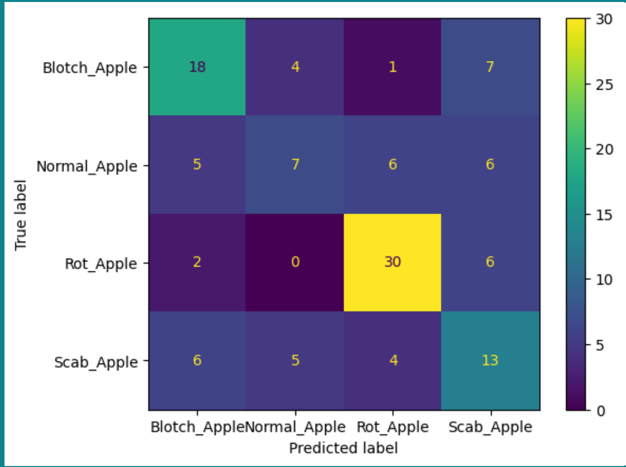
CNN (Apple Disease Classification) MODEL 3

```
model = models.Sequential([
             resize_and_rescale,
             layers.Conv2D(32, (3,3), activation='relu', input_shape=input_shape),
             tf.keras.layers.BatchNormalization(),
             layers.MaxPooling2D((2,2)),
             layers.Conv2D(64, (3,3), activation='relu'),
             layers.MaxPooling2D((2,2)),
             layers.Conv2D(64, (3,3), activation='relu'),
             layers.MaxPooling2D((2,2)),
             layers.Conv2D(128, (3,3), activation='relu'),
             layers.MaxPooling2D((2,2)),
             layers.Conv2D(128, (3,3), activation='relu'),
             layers.MaxPooling2D((2,2)),
             layers.Conv2D(256, (3,3), activation='relu'),
             layers.MaxPooling2D((2,2)),
             layers.Flatten(),
             layers.Dense(64, activation='relu'),
             layers.Dense(n_classes, activation='softmax')
             ])
```

Epochs = 15

image = 224x224
augmented_data (4x1000)
learning_rate = 0.001





	Blotch_Apple	Normal_Apple	Rot_Apple	Scab_Apple
Blotch_Apple	18	4	1	7
Normal_Apple	5	7	6	6
Rot_Apple	2	0	30	6
Scab_Apple	6	5	4	13

Confusion matrix 68/120 **0.5667**

COMMENTS

Met Batch Normalization in 1 laag zijn de Train en Validation Accuracy exact gelijk (bij deze run).
Test Accuracy gaat ook weer een stapje omhoog.

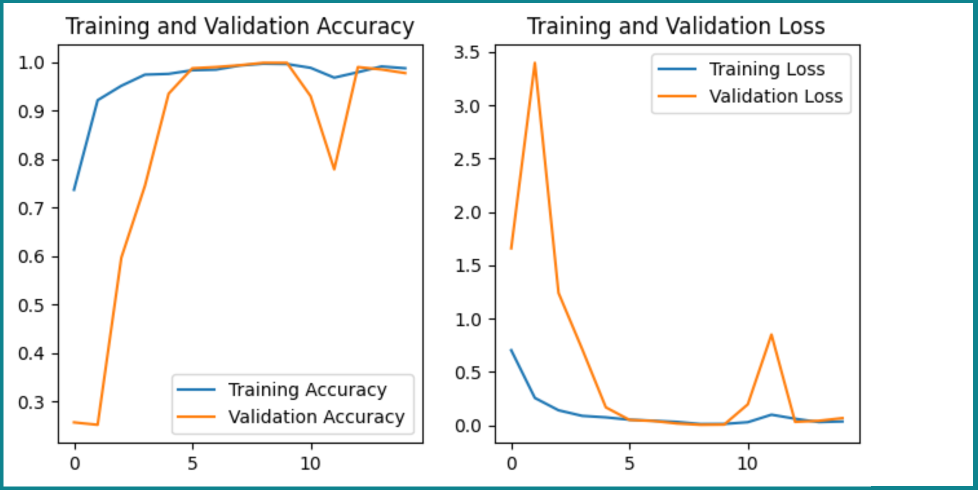
Loss	0.0284	**Accuracy**	**0.9912**
Val_Loss	0.0276	**Val_Accuracy**	**0.9912**
Test_Loss	2.0958	**Test_Accuracy**	**0.5667**

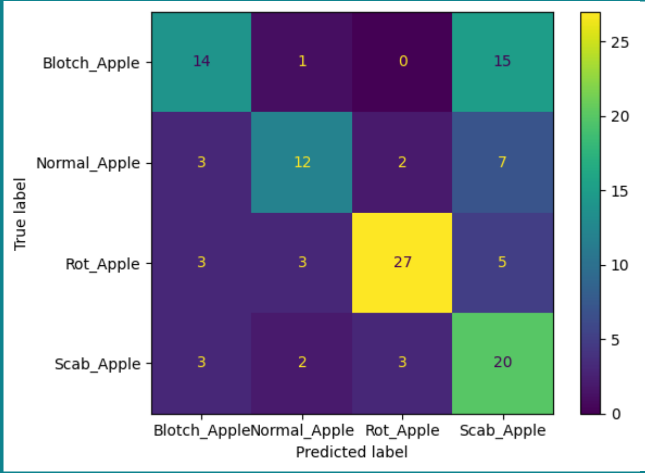
CNN (Apple Disease Classification) MODEL 4

```
model = models.Sequential([
          resize_and_rescale,
          layers.Conv2D(32, (3,3), activation='relu', input_shape=input_shape),
          tf.keras.layers.BatchNormalization(),                                    BLOK
          layers.MaxPooling2D((2,2)),
          layers.Conv2D(64, (3,3), activation='relu'),
          tf.keras.layers.BatchNormalization(),                                    BLOK
          layers.MaxPooling2D((2,2)),
          layers.Conv2D(64, (3,3), activation='relu'),
          tf.keras.layers.BatchNormalization(),                                    BLOK
          layers.MaxPooling2D((2,2)),
          layers.Conv2D(128, (3,3), activation='relu'),
          tf.keras.layers.BatchNormalization(),                                    BLOK
          layers.MaxPooling2D((2,2)),
          layers.Conv2D(128, (3,3), activation='relu'),
          tf.keras.layers.BatchNormalization(),                                    BLOK
          layers.MaxPooling2D((2,2)),
          layers.Conv2D(256, (3,3), activation='relu'),
          tf.keras.layers.BatchNormalization(),                                    BLOK
          layers.MaxPooling2D((2,2)),
          layers.Flatten(),
          layers.Dense(64, activation='relu'),
          layers.Dense(n_classes, activation='softmax')
          ])
```

Epochs = 15

image = 224x224
augmented_data (4x1000)
learning_rate = 0.001





	Blotch_Apple	Normal_Apple	Rot_Apple	Scab_Apple
Blotch_Apple	14	1	0	15
Normal_Apple	3	12	2	7
Rot_Apple	3	3	27	5
Scab_Apple	3	2	3	20

Confusion matrix 73/120 **0.6083**

Loss	0.0353	**Accuracy**	**0.9875**
Val_Loss	0.0669	**Val_Accuracy**	**0.9775**
Test_Loss	2.425	**Test_Accuracy**	**0.6083**

COMMENTS

Met Batch Normalization in elk '***BLOK***' komen we eindelijk net boven de 60% uit.
Rond epoch 12 een piek die daarna weer normaliseert.
De gewenste diagonaal begint zich langzaam te tonen in de Confusion Matrix.

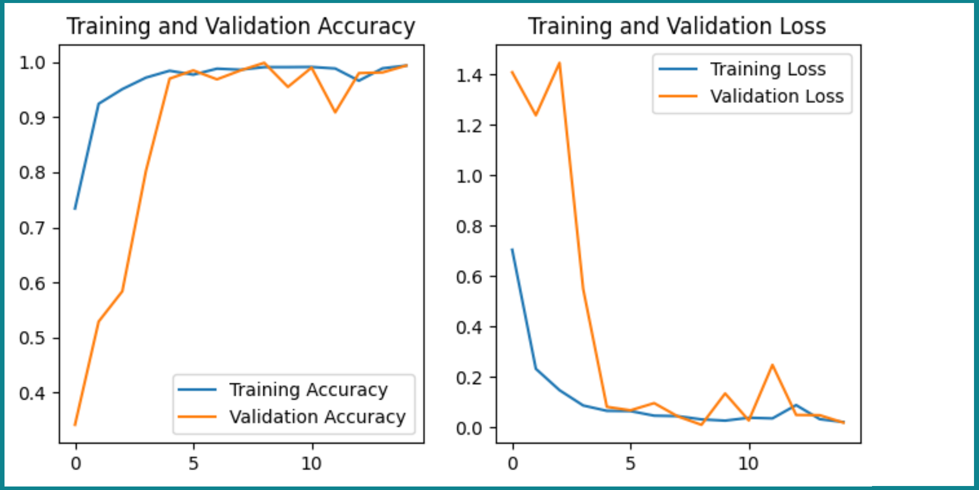
Het model heeft duidelijk moeite met het onderscheid tussen Blotch en Scab appels.

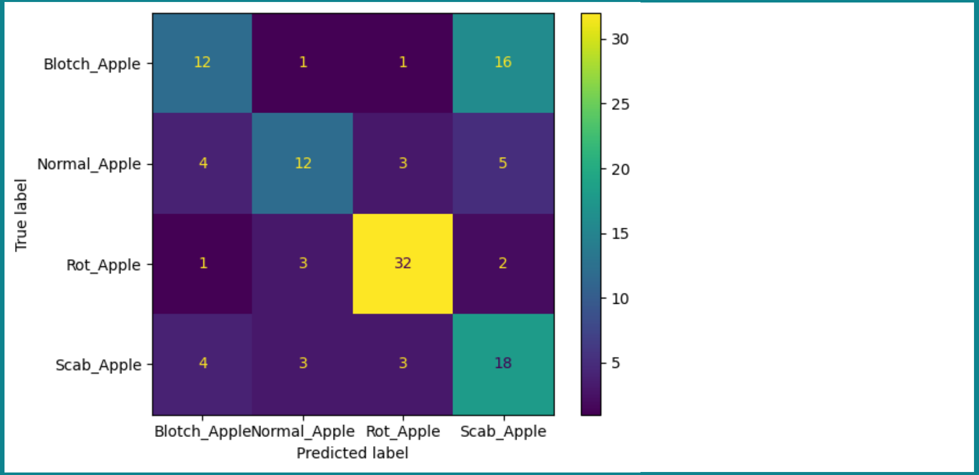
CNN (Apple Disease Classification)   MODEL   6

```
model = models.Sequential([
             resize_and_rescale,
             layers.Conv2D(32, (3,3), activation='relu', input_shape=input_shape),
             tf.keras.layers.BatchNormalization(),
             layers.MaxPooling2D((2,2)),
             layers.Conv2D(64, (3,3), activation='relu'),
             tf.keras.layers.BatchNormalization(),
             layers.MaxPooling2D((2,2)),
             layers.Conv2D(64, (3,3), activation='relu'),
             tf.keras.layers.BatchNormalization(),
             layers.MaxPooling2D((2,2)),
             layers.Conv2D(128, (3,3), activation='relu'),
             tf.keras.layers.BatchNormalization(),
             layers.MaxPooling2D((2,2)),
             layers.Conv2D(128, (3,3), activation='relu'),
             tf.keras.layers.BatchNormalization(),
             layers.MaxPooling2D((2,2)),
             layers.Conv2D(256, (3,3), activation='relu'),
             tf.keras.layers.BatchNormalization(),
             layers.MaxPooling2D((2,2)),
             layers.Flatten(),
             layers.Dense(64, activation='relu'),
             layers.Dense(n_classes, activation='softmax')
             ])
```

Epochs = 15

image = 224x224
augmented_data (4x1000)
learning_rate = 0.001





	Blotch_Apple	Normal_Apple	Rot_Apple	Scab_Apple
Blotch_Apple	12	1	1	16
Normal_Apple	4	12	3	5
Rot_Apple	1	3	32	2
Scab_Apple	4	3	3	18

Confusion matrix   74/120   **0.6167**

Loss	0.0197	**Accuracy**	**0.9937**
Val_Loss	0.0173	**Val_Accuracy**	**0.9937**
Test_Loss	2.0898	**Test_Accuracy**	**0.6167**

COMMENTS

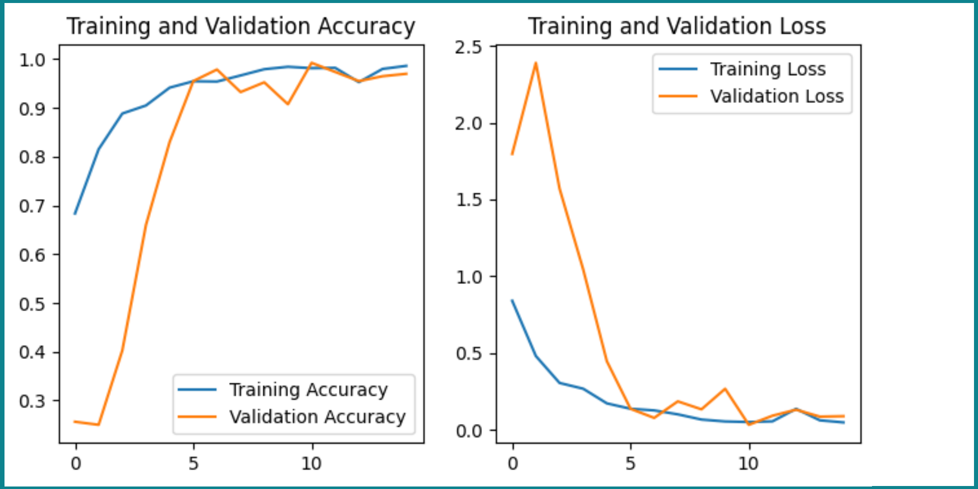
Check voor Model 4, ook hier zien we de opvallende piek rond epoch 12.
Weliswaar iets kleiner en voorafgegaan door twee kleinere piekjes.
Test Accuracy nog iets beter geworden.

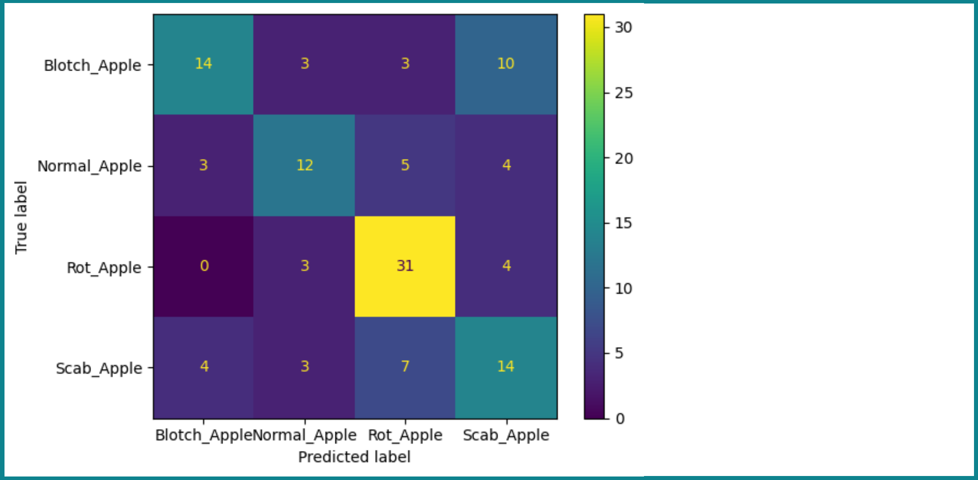
CNN (Apple Disease Classification) MODEL 7

```
model = models.Sequential([
            resize_and_rescale,
            data_augmentation,
                        RandomFlip("horizontal_and_vertical")
            layers.Conv2D(32, (3,3), activation='relu', input_shape=input_shape),
            tf.keras.layers.BatchNormalization(),
            layers.MaxPooling2D((2,2)),
            layers.Conv2D(64, (3,3), activation='relu'),
            tf.keras.layers.BatchNormalization(),
            layers.MaxPooling2D((2,2)),
            layers.Conv2D(64, (3,3), activation='relu'),
            tf.keras.layers.BatchNormalization(),
            layers.MaxPooling2D((2,2)),
            layers.Conv2D(128, (3,3), activation='relu'),
            tf.keras.layers.BatchNormalization(),
            layers.MaxPooling2D((2,2)),
            layers.Conv2D(128, (3,3), activation='relu'),
            tf.keras.layers.BatchNormalization(),
            layers.MaxPooling2D((2,2)),
            layers.Conv2D(256, (3,3), activation='relu'),
            tf.keras.layers.BatchNormalization(),
            layers.MaxPooling2D((2,2)),
            layers.Flatten(),
            layers.Dense(64, activation='relu'),
            layers.Dense(n_classes, activation='softmax')
            ])
```

Epochs = 15

image = 224x224
augmented_data (4x1000)
learning_rate = 0.001





Confusion matrix 71/120 **0.5917**

Loss	0.0464	**Accuracy**	**0.9862**
Val_Loss	0.0870	**Val_Accuracy**	**0.9700**
Test_Loss	2.8493	**Test_Accuracy**	**0.5917**

COMMENTS

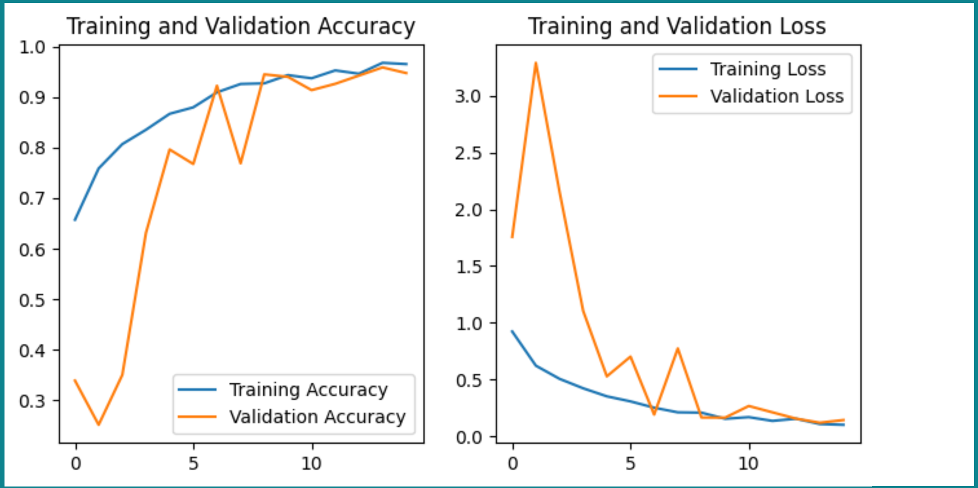
De eerst data augmentatie: RandomFlip, zorgt niet voor de verwachtte verbetering.
Het model wordt zelfs minder goed dan voorheen.

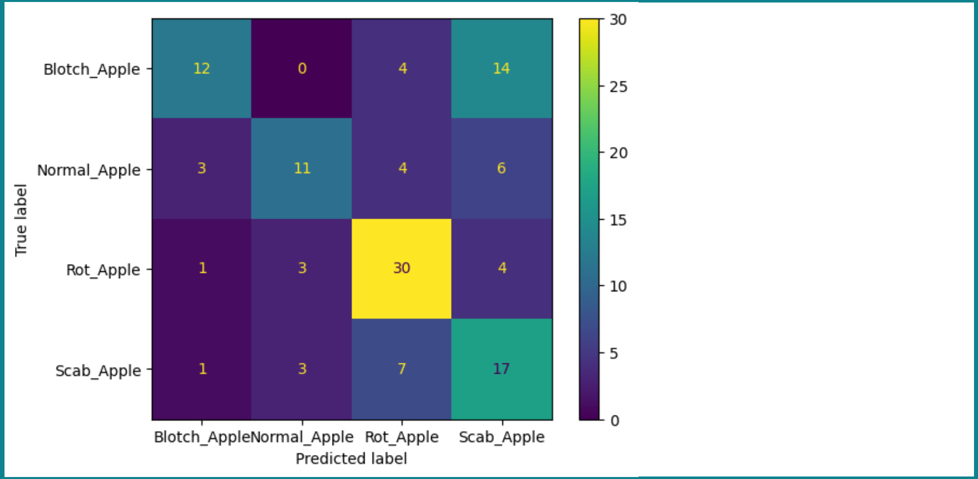
CNN (Apple Disease Classification) MODEL 8

```
model = models.Sequential([
             resize_and_rescale,
             data_augmentation,
                          RandomFlip("horizontal_and_vertical")
                          RandomRotation(0.2),
             layers.Conv2D(32, (3,3), activation='relu', input_shape=input_shape),
             tf.keras.layers.BatchNormalization(),
             layers.MaxPooling2D((2,2)),
             layers.Conv2D(64, (3,3), activation='relu'),
             tf.keras.layers.BatchNormalization(),
             layers.MaxPooling2D((2,2)),
             layers.Conv2D(64, (3,3), activation='relu'),
             tf.keras.layers.BatchNormalization(),
             layers.MaxPooling2D((2,2)),
             layers.Conv2D(128, (3,3), activation='relu'),
             tf.keras.layers.BatchNormalization(),
             layers.MaxPooling2D((2,2)),
             layers.Conv2D(128, (3,3), activation='relu'),
             tf.keras.layers.BatchNormalization(),
             layers.MaxPooling2D((2,2)),
             layers.Conv2D(256, (3,3), activation='relu'),
             tf.keras.layers.BatchNormalization(),
             layers.MaxPooling2D((2,2)),
             layers.Flatten(),
             layers.Dense(64, activation='relu'),
             layers.Dense(n_classes, activation='softmax')
             ])
```

Epochs = 15

image = 224x224
augmented_data (4x1000)
learning_rate = 0.001





Confusion matrix 70/120 **0.5833**

COMMENTS

Naast RandomFlip, ook RandomRotation toegevoegd aan de Data Augmentation.
Model heeft er wederom moeite mee. Score blijft afnemen.

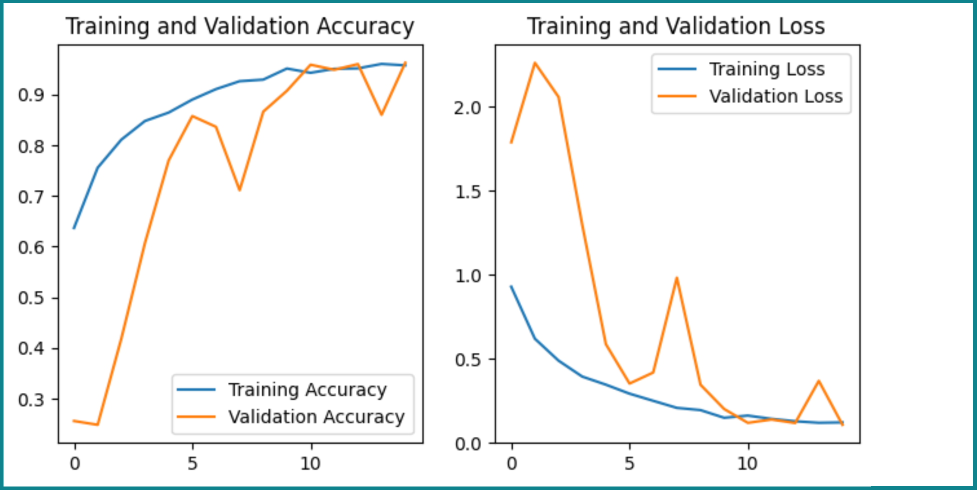
Loss	0.1008	**Accuracy**	**0.9653**
Val_Loss	0.1426	**Val_Accuracy**	**0.9475**
Test_Loss	2.1795	**Test_Accuracy**	**0.5833**

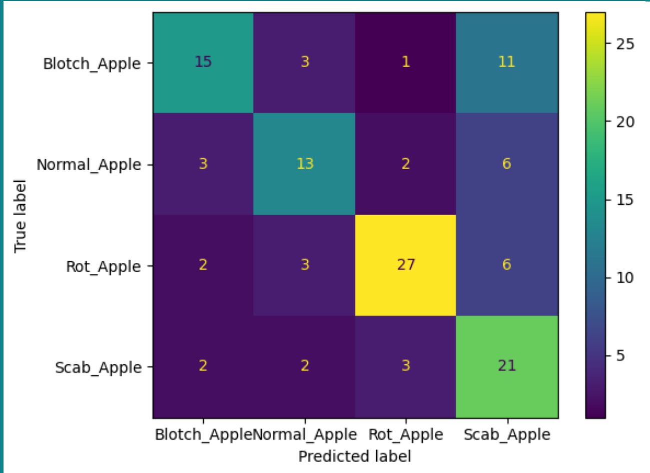
CNN (Apple Disease Classification) MODEL 9

```
model = models.Sequential([
             resize_and_rescale,
             data_augmentation,
                          RandomFlip("horizontal_and_vertical")
                          RandomRotation(0.2),
             layers.Conv2D(32, (3,3), activation='relu', input_shape=input_shape),
             tf.keras.layers.BatchNormalization(),
             layers.MaxPooling2D((2,2)),
             layers.Conv2D(64, (3,3), activation='relu'),
             tf.keras.layers.BatchNormalization(),
             layers.MaxPooling2D((2,2)),
             layers.Conv2D(64, (3,3), activation='relu'),
             tf.keras.layers.BatchNormalization(),
             layers.MaxPooling2D((2,2)),
             layers.Conv2D(128, (3,3), activation='relu'),
             tf.keras.layers.BatchNormalization(),
             layers.MaxPooling2D((2,2)),
             layers.Conv2D(128, (3,3), activation='relu'),
             tf.keras.layers.BatchNormalization(),
             layers.MaxPooling2D((2,2)),
             layers.Conv2D(256, (3,3), activation='relu'),
             tf.keras.layers.BatchNormalization(),
             layers.MaxPooling2D((2,2)),
             layers.Flatten(),
             layers.Dense(64, activation='relu'),
             layers.Dense(n_classes, activation='softmax')
             ])
```

Epochs = 15

image = 224x224
augmented_data (4x1000)
learning_rate = 0.001





Confusion matrix 76/120 **0.6333**

Loss	0.1218	**Accuracy**	**0.9978**
Val_Loss	0.1075	**Val_Accuracy**	**0.9625**
Test_Loss	1.6645	**Test_Accuracy**	**0.6333**

COMMENTS

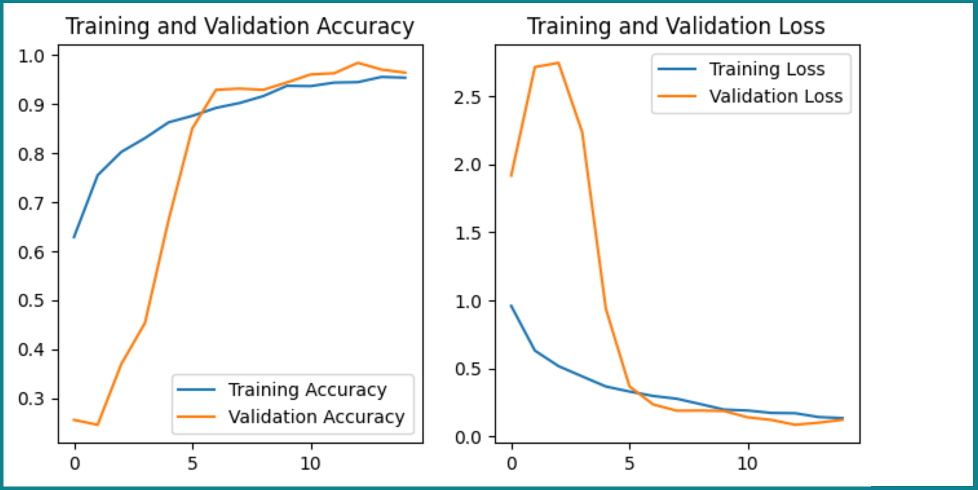
Validation is erg schokkerig in Model 8, vandaar met zelfde instellingen nogmaals gerund.
Ook nu een erg 'erratic' lijn bij de Validation, maar wel een beduidend betere score.

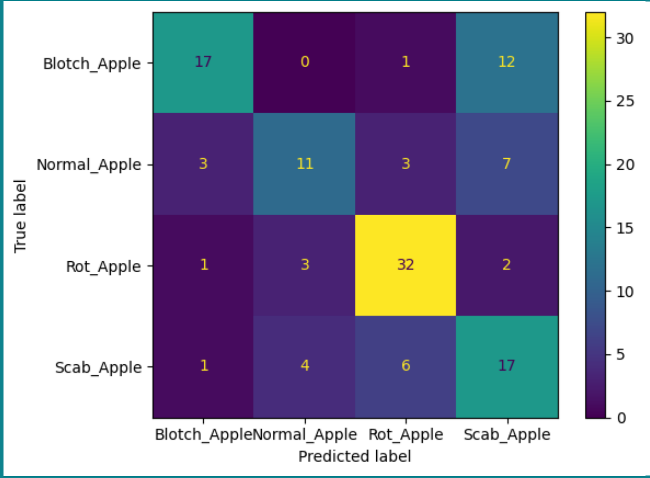
CNN (Apple Disease Classification) MODEL 23

```
model = models.Sequential([
             resize_and_rescale,
             data_augmentation,
                          RandomFlip("horizontal_and_vertical")
                          RandomRotation(0.2),
             layers.Conv2D(32, (3,3), activation='relu', input_shape=input_shape),
             batch_normalization,
             layers.MaxPooling2D((2,2)),
             layers.Conv2D(64, (3,3),activation='relu'),
             tf.keras.layers.BatchNormalization(),
             layers.MaxPooling2D((2,2)),
             layers.Conv2D(64, (3,3), activation='relu'),
             tf.keras.layers.BatchNormalization(),
             layers.MaxPooling2D((2,2)),
             layers.Conv2D(128, (3,3), activation='relu'),
             tf.keras.layers.BatchNormalization(),
             layers.MaxPooling2D((2,2)),
             layers.Conv2D(128, (3,3), activation='relu'),
             tf.keras.layers.BatchNormalization(),
             layers.MaxPooling2D((2,2)),
             layers.Conv2D(256, (3,3), activation='relu'),
             tf.keras.layers.BatchNormalization(),
             layers.MaxPooling2D((2,2)),
             layers.Flatten(),
             layers.Dense(64, activation='relu'),
             layers.Dense(n_classes, activation='softmax')
             ])
```

Epochs = 15

image = 224x224
augmented_data (4x1000)
learning_rate = 0.0001





True \ Predicted	Blotch_Apple	Normal_Apple	Rot_Apple	Scab_Apple
Blotch_Apple	17	0	1	12
Normal_Apple	3	11	3	7
Rot_Apple	1	3	32	2
Scab_Apple	1	4	6	17

Confusion matrix 77/120 **0.6417**

Loss	0.1369	**Accuracy**	**0.9534**
Val_Loss	0.1242	**Val_Accuracy**	**0.9638**
Test_Loss	1.5591	**Test_Accuracy**	**0.6417**

COMMENTS

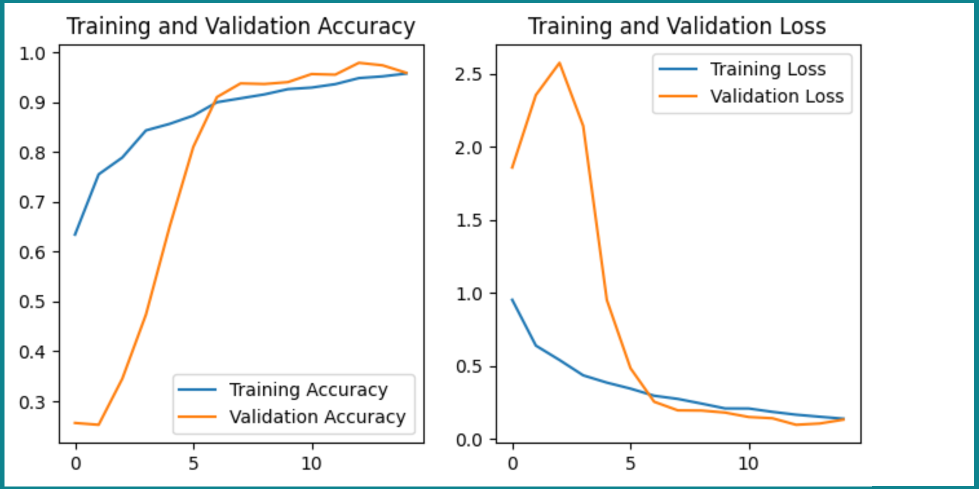
Nogmaals met instelligen van Model 8/9. Veel rustigere Validation-lijn.
Ook hier moet de Validation een paar epochs door voor de lijn naar de Train-lijn toe trekt.
Score wordt iets beter.

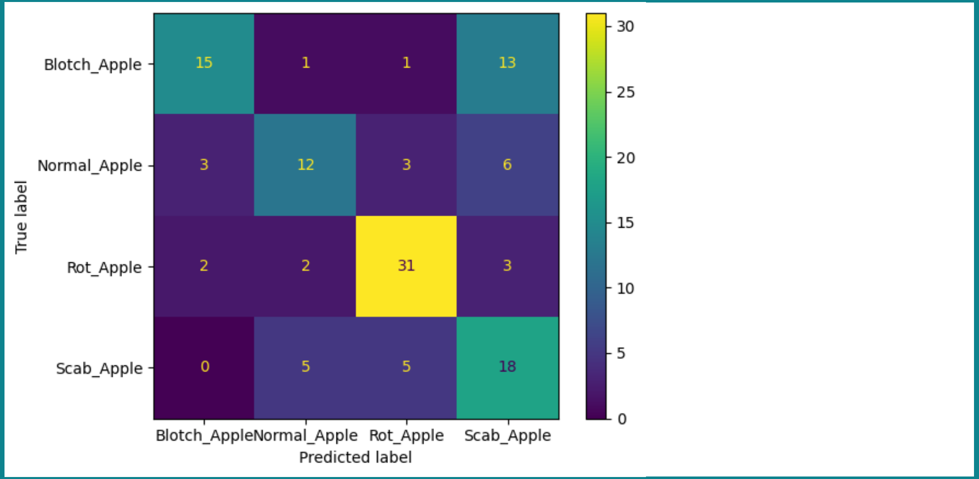
CNN (Apple Disease Classification) MODEL 24

```
model = models.Sequential([
             resize_and_rescale,
             data_augmentation,
                          RandomFlip("horizontal_and_vertical")
                          RandomRotation(0.2),
             layers.Conv2D(32, (3,3), activation='relu', input_shape=input_shape),
             batch_normalization,
             layers.MaxPooling2D((2,2)),
             layers.Conv2D(64, (3,3),activation='relu'),
             tf.keras.layers.BatchNormalization(),
             layers.MaxPooling2D((2,2)),
             layers.Conv2D(64, (3,3), activation='relu'),
             tf.keras.layers.BatchNormalization(),
             layers.MaxPooling2D((2,2)),
             layers.Conv2D(128, (3,3), activation='relu'),
             tf.keras.layers.BatchNormalization(),
             layers.MaxPooling2D((2,2)),
             layers.Conv2D(128, (3,3), activation='relu'),
             tf.keras.layers.BatchNormalization(),
             layers.MaxPooling2D((2,2)),
             layers.Conv2D(256, (3,3), activation='relu'),
             tf.keras.layers.BatchNormalization(),
             layers.MaxPooling2D((2,2)),
             layers.Flatten(),
             layers.Dense(64, activation='relu'),
             layers.Dense(n_classes, activation='softmax')
             ])
```

Epochs = 15

image = 224x224
augmented_data (4x1000)
learning_rate = 0.0001





True label \ Predicted label	Blotch_Apple	Normal_Apple	Rot_Apple	Scab_Apple
Blotch_Apple	15	1	1	13
Normal_Apple	3	12	3	6
Rot_Apple	2	2	31	3
Scab_Apple	0	5	5	18

Confusion matrix 76/120 **0.6333**

Loss	0.1382	**Accuracy**	**0.9572**
Val_Loss	0.1320	**Val_Accuracy**	**0.9588**
Test_Loss	1.6219	**Test_Accuracy**	**0.6333**

COMMENTS

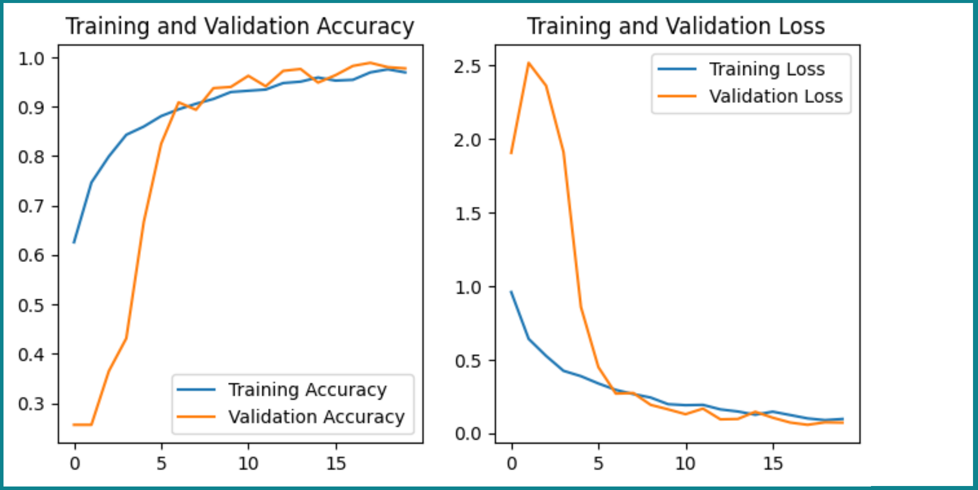
Ter controle
Met dezelfde instellingen als Model 23. Iets lagere score.

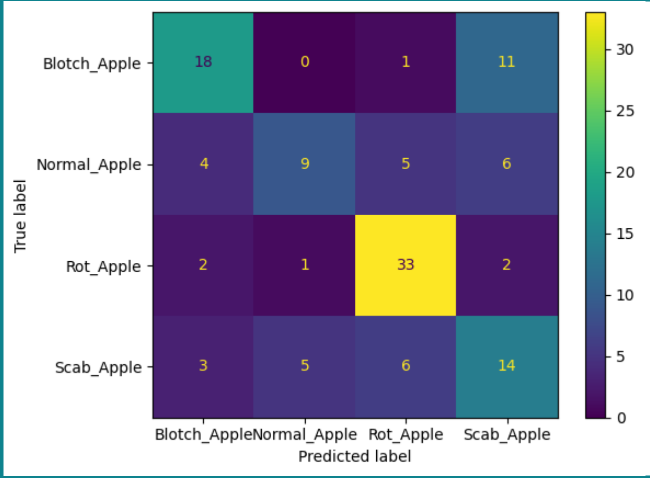
CNN (Apple Disease Classification) MODEL 25

```
model = models.Sequential([
            resize_and_rescale,
            data_augmentation,
                        RandomFlip("horizontal_and_vertical")
                        RandomRotation(0.2),
            layers.Conv2D(32, (3,3), activation='relu', input_shape=input_shape),
            batch_normalization,
            layers.MaxPooling2D((2,2)),
            layers.Conv2D(64, (3,3),activation='relu'),
            tf.keras.layers.BatchNormalization(),
            layers.MaxPooling2D((2,2)),
            layers.Conv2D(64, (3,3), activation='relu'),
            tf.keras.layers.BatchNormalization(),
            layers.MaxPooling2D((2,2)),
            layers.Conv2D(128, (3,3), activation='relu'),
            tf.keras.layers.BatchNormalization(),
            layers.MaxPooling2D((2,2)),
            layers.Conv2D(128, (3,3), activation='relu'),
            tf.keras.layers.BatchNormalization(),
            layers.MaxPooling2D((2,2)),
            layers.Conv2D(256, (3,3), activation='relu'),
            tf.keras.layers.BatchNormalization(),
            layers.MaxPooling2D((2,2)),
            layers.Flatten(),
            layers.Dense(64, activation='relu'),
            layers.Dense(n_classes, activation='softmax')
            ])
```

Epochs = 20

image = 224x224
augmented_data (4x1000)
learning_rate = 0.0001





	Blotch_Apple	Normal_Apple	Rot_Apple	Scab_Apple
Blotch_Apple	18	0	1	11
Normal_Apple	4	9	5	6
Rot_Apple	2	1	33	2
Scab_Apple	3	5	6	14

Confusion matrix 74/120 **0.6167**

COMMENTS

Met 20 epochs wordt de score niet beter ...

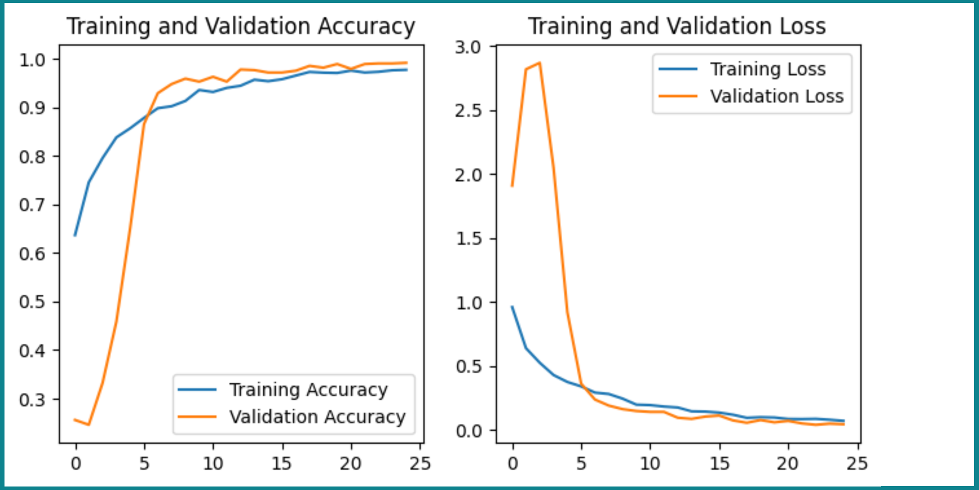
Loss	0.0973	**Accuracy**	**0.9694**
Val_Loss	0.0726	**Val_Accuracy**	**0.9775**
Test_Loss	1.5984	**Test_Accuracy**	**0.6167**

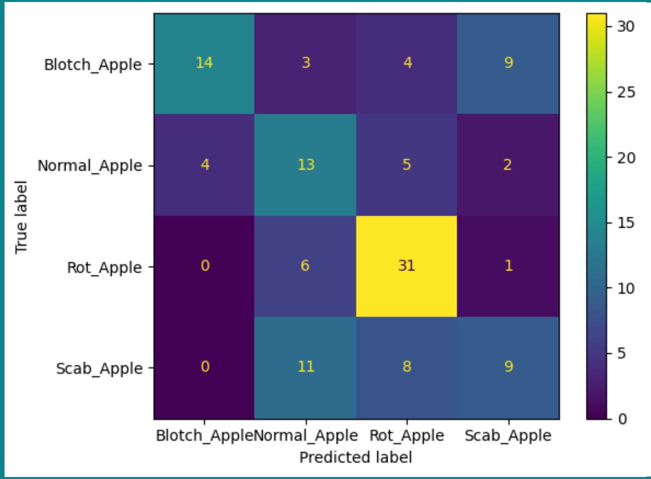
CNN (Apple Disease Classification) MODEL 26

```
model = models.Sequential([
            resize_and_rescale,
            data_augmentation,
                        RandomFlip("horizontal_and_vertical")
                        RandomRotation(0.2),
            layers.Conv2D(32, (3,3), activation='relu', input_shape=input_shape),
            batch_normalization,
            layers.MaxPooling2D((2,2)),
            layers.Conv2D(64, (3,3),activation='relu'),
            tf.keras.layers.BatchNormalization(),
            layers.MaxPooling2D((2,2)),
            layers.Conv2D(64, (3,3), activation='relu'),
            tf.keras.layers.BatchNormalization(),
            layers.MaxPooling2D((2,2)),
            layers.Conv2D(128, (3,3), activation='relu'),
            tf.keras.layers.BatchNormalization(),
            layers.MaxPooling2D((2,2)),
            layers.Conv2D(128, (3,3), activation='relu'),
            tf.keras.layers.BatchNormalization(),
            layers.MaxPooling2D((2,2)),
            layers.Conv2D(256, (3,3), activation='relu'),
            tf.keras.layers.BatchNormalization(),
            layers.MaxPooling2D((2,2)),
            layers.Flatten(),
            layers.Dense(64, activation='relu'),
            layers.Dense(n_classes, activation='softmax')
            ])
```

Epochs = 25

image = 224x224
augmented_data (4x1000)
learning_rate = 0.0001





True label \ Predicted label	Blotch_Apple	Normal_Apple	Rot_Apple	Scab_Apple
Blotch_Apple	14	3	4	9
Normal_Apple	4	13	5	2
Rot_Apple	0	6	31	1
Scab_Apple	0	11	8	9

Confusion matrix 67/120 **0.5583**

COMMENTS

Met 25 epochs wordt er geen verbetering behaald. Resultaat wordt zelfs (beduidend) minder.

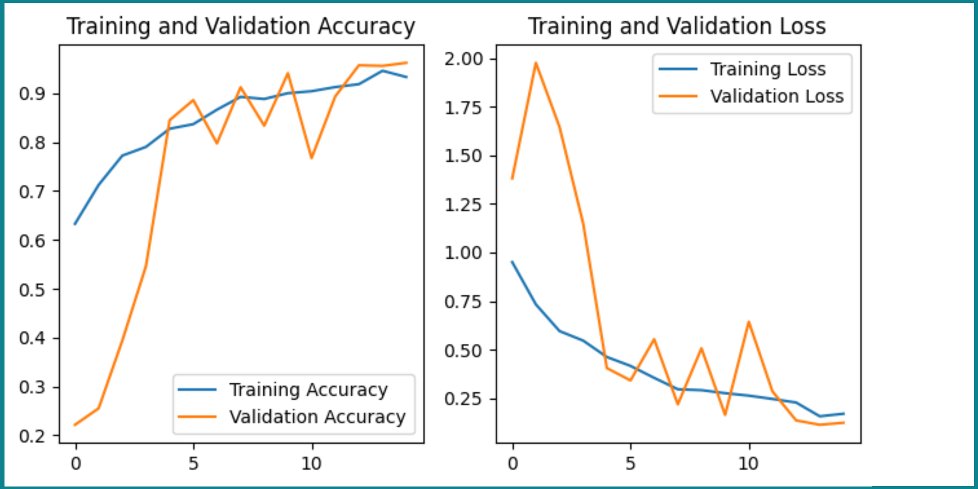
Loss	0.0693	**Accuracy**	**0.9769**
Val_Loss	0.0428	**Val_Accuracy**	**0.9912**
Test_Loss	1.9283	**Test_Accuracy**	**0.5583**

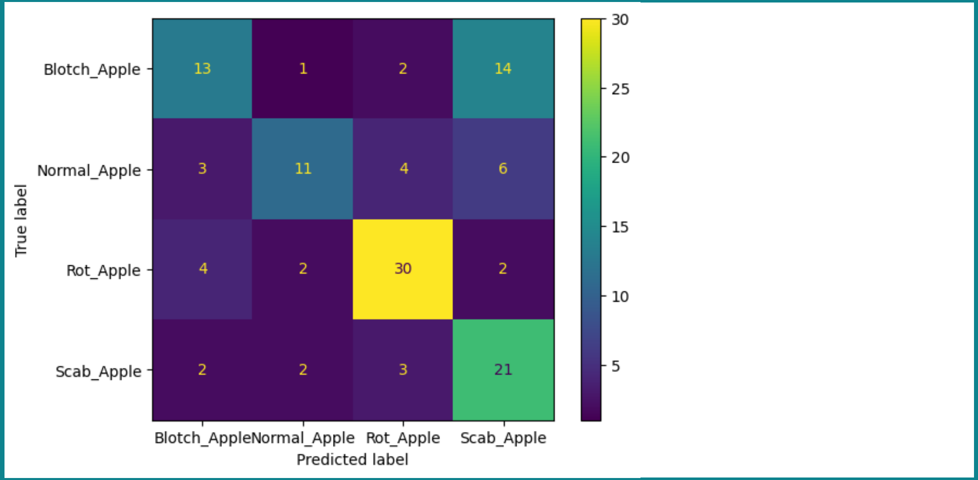
CNN (Apple Disease Classification) MODEL 10

```
model = models.Sequential([
            resize_and_rescale,
            data_augmentation,
                        RandomFlip("horizontal_and_vertical")
                        RandomRotation(0.3),
                        RandomZoom(0.3)
            layers.Conv2D(32, (3,3), activation='relu', input_shape=input_shape),
            tf.keras.layers.BatchNormalization(),
            layers.MaxPooling2D((2,2)),
            layers.Conv2D(64, (3,3), activation='relu'),
            tf.keras.layers.BatchNormalization(),
            layers.MaxPooling2D((2,2)),
            layers.Conv2D(64, (3,3), activation='relu'),
            tf.keras.layers.BatchNormalization(),
            layers.MaxPooling2D((2,2)),
            layers.Conv2D(128, (3,3), activation='relu'),
            tf.keras.layers.BatchNormalization(),
            layers.MaxPooling2D((2,2)),
            layers.Conv2D(128, (3,3), activation='relu'),
            tf.keras.layers.BatchNormalization(),
            layers.MaxPooling2D((2,2)),
            layers.Conv2D(256, (3,3), activation='relu'),
            tf.keras.layers.BatchNormalization(),
            layers.MaxPooling2D((2,2)),
            layers.Flatten(),
            layers.Dense(64, activation='relu'),
            ])
            ])
```

Epochs = 15

image = 224x224
augmented_data (4x1000)
learning_rate = 0.001





	Blotch_Apple	Normal_Apple	Rot_Apple	Scab_Apple
Blotch_Apple	13	1	2	14
Normal_Apple	3	11	4	6
Rot_Apple	4	2	30	2
Scab_Apple	2	2	3	21

Confusion matrix 75/120 **0.6250**

COMMENTS

RandomZoom(0.3) toegevoegd. Resultaat weer wat beter

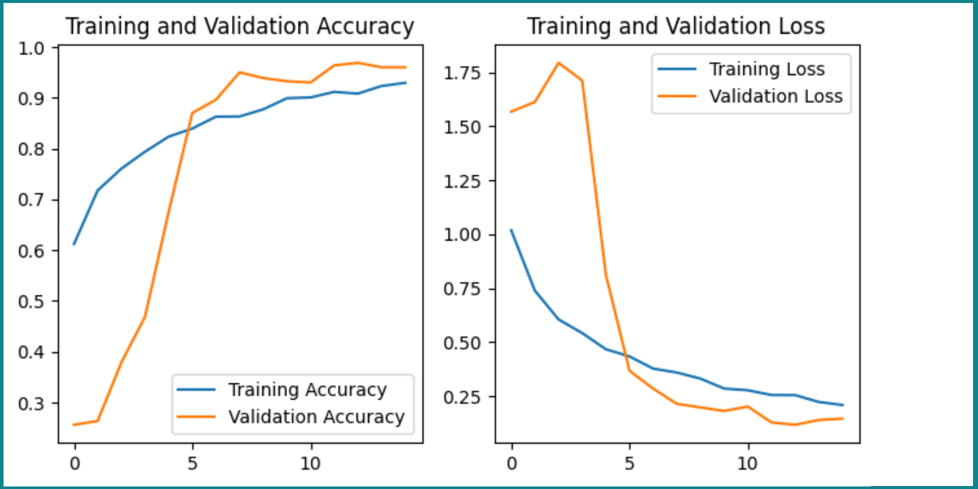
Loss	0.1695	**Accuracy**	**0.9334**
Val_Loss	0.1232	**Val_Accuracy**	**0.9625**
Test_Loss	1.779	**Test_Accuracy**	**0.6250**

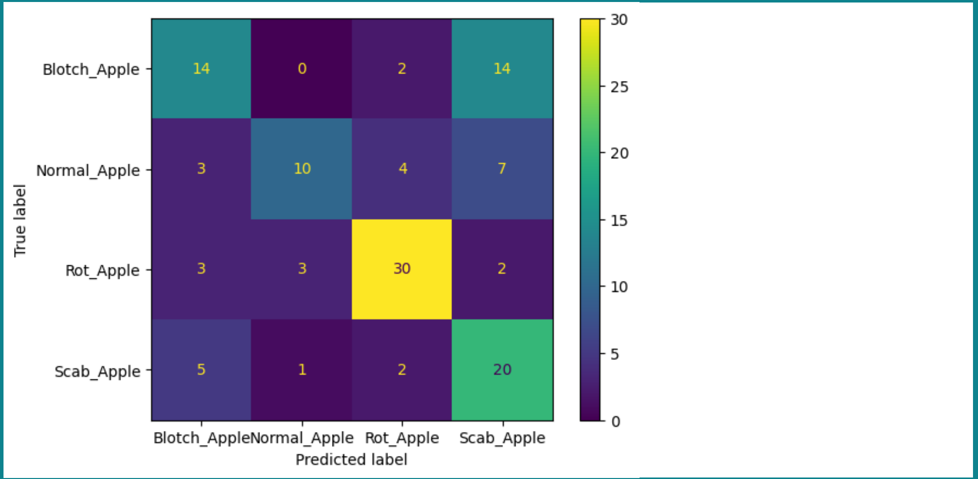
CNN (Apple Disease Classification) MODEL 27

```
model = models.Sequential([
            resize_and_rescale,
            data_augmentation,
                        RandomFlip("horizontal_and_vertical")
                        RandomRotation(0.3),
                        RandomZoom(0.3)
            layers.Conv2D(32, (3,3), activation='relu', input_shape=input_shape),
            tf.keras.layers.BatchNormalization(),
            layers.MaxPooling2D((2,2)),
            layers.Conv2D(64, (3,3), activation='relu'),
            tf.keras.layers.BatchNormalization(),
            layers.MaxPooling2D((2,2)),
            layers.Conv2D(64, (3,3), activation='relu'),
            tf.keras.layers.BatchNormalization(),
            layers.MaxPooling2D((2,2)),
            layers.Conv2D(128, (3,3), activation='relu'),
            tf.keras.layers.BatchNormalization(),
            layers.MaxPooling2D((2,2)),
            layers.Conv2D(128, (3,3), activation='relu'),
            tf.keras.layers.BatchNormalization(),
            layers.MaxPooling2D((2,2)),
            layers.Conv2D(256, (3,3), activation='relu'),
            tf.keras.layers.BatchNormalization(),
            layers.MaxPooling2D((2,2)),
            layers.Flatten(),
            layers.Dense(64, activation='relu'),
            ])
            ])
```

Epochs = 15

image = 224x224
augmented_data (4x1000)
learning_rate = 0.0001





	Blotch_Apple	Normal_Apple	Rot_Apple	Scab_Apple
Blotch_Apple	14	0	2	14
Normal_Apple	3	10	4	7
Rot_Apple	3	3	30	2
Scab_Apple	5	1	2	20

Confusion matrix 74/120 **0.6167**

COMMENTS

Ter controle werderom met RandomZoom(0.3) gerund. Abusievelijk met lr=0.0001.
De Learning Rate lijkt niet heel veel invloed te hebben op het resultaat.

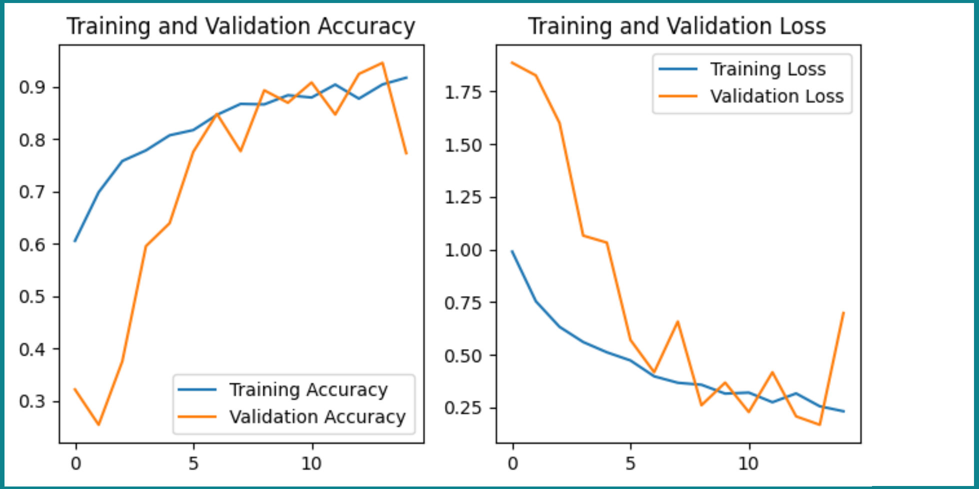
Loss	0.2085	**Accuracy**	**0.9294**
Val_Loss	0.1452	**Val_Accuracy**	**0.9600**
Test_Loss	1.4171	**Test_Accuracy**	**0.6167**

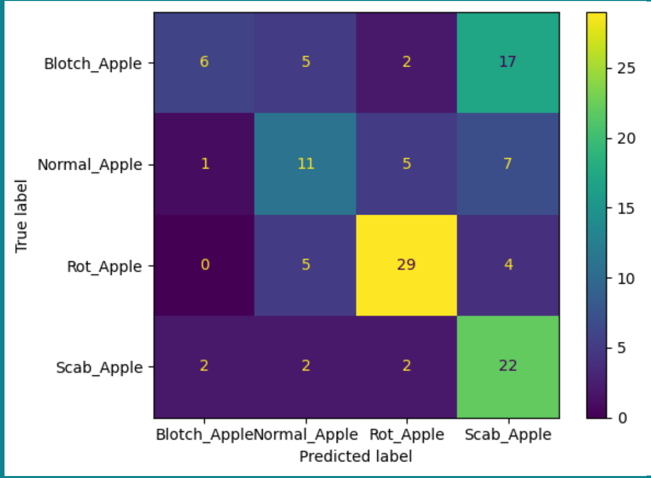
CNN (Apple Disease Classification) MODEL 11

```
model = models.Sequential([
            resize_and_rescale,
            data_augmentation,
                        RandomFlip("horizontal_and_vertical")
                        RandomRotation(0.3),
                        RandomZoom(0.4)
            layers.Conv2D(32, (3,3), activation='relu', input_shape=input_shape),
            tf.keras.layers.BatchNormalization(),
            layers.MaxPooling2D((2,2)),
            layers.Conv2D(64, (3,3), activation='relu'),
            tf.keras.layers.BatchNormalization(),
            layers.MaxPooling2D((2,2)),
            layers.Conv2D(64, (3,3), activation='relu'),
            tf.keras.layers.BatchNormalization(),
            layers.MaxPooling2D((2,2)),
            layers.Conv2D(128, (3,3), activation='relu'),
            tf.keras.layers.BatchNormalization(),
            layers.MaxPooling2D((2,2)),
            layers.Conv2D(128, (3,3), activation='relu'),
            tf.keras.layers.BatchNormalization(),
            layers.MaxPooling2D((2,2)),
            layers.Conv2D(256, (3,3), activation='relu'),
            tf.keras.layers.BatchNormalization(),
            layers.MaxPooling2D((2,2)),
            layers.Flatten(),
            layers.Dense(64, activation='relu'),
            ])
```

Epochs = 15

image = 224x224
augmented_data (4x1000)
learning_rate = 0.001





	Blotch_Apple	Normal_Apple	Rot_Apple	Scab_Apple
Blotch_Apple	6	5	2	17
Normal_Apple	1	11	5	7
Rot_Apple	0	5	29	4
Scab_Apple	2	2	2	22

Confusion matrix 68/120 **0.5667**

COMMENTS

Met een kleine aanpassing van +0.1 bij RandomZoom wordt het model slechter.

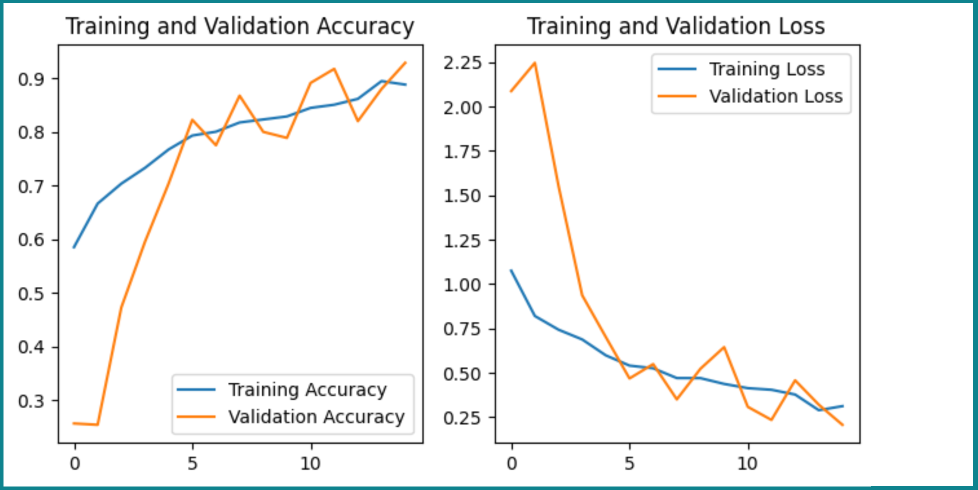
Loss	0.2313	**Accuracy**	**0.6976**
Val_Loss	0.6976	**Val_Accuracy**	**0.7725**
Test_Loss	2.0272	**Test_Accuracy**	**0.5667**

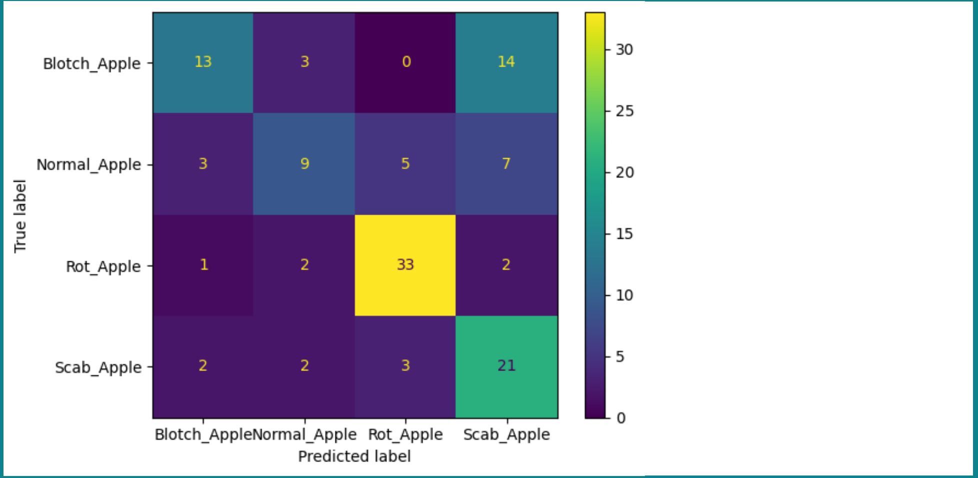
CNN (Apple Disease Classification) MODEL 30

```
model = models.Sequential([
            resize_and_rescale,
            data_augmentation,
                        RandomFlip("horizontal_and_vertical")
                        RandomRotation(0.3),
                        RandomZoom(0.4)
            layers.Conv2D(32, (3,3), activation='relu', input_shape=input_shape),
            tf.keras.layers.BatchNormalization(),
            layers.MaxPooling2D((2,2)),
            layers.Conv2D(64, (3,3), activation='relu'),
            tf.keras.layers.BatchNormalization(),
            layers.MaxPooling2D((2,2)),
            layers.Conv2D(64, (3,3), activation='relu'),
            tf.keras.layers.BatchNormalization(),
            layers.MaxPooling2D((2,2)),
            layers.Conv2D(128, (3,3), activation='relu'),
            tf.keras.layers.BatchNormalization(),
            layers.MaxPooling2D((2,2)),
            layers.Conv2D(128, (3,3), activation='relu'),
            tf.keras.layers.BatchNormalization(),
            layers.MaxPooling2D((2,2)),
            layers.Conv2D(256, (3,3), activation='relu'),
            tf.keras.layers.BatchNormalization(),
            layers.MaxPooling2D((2,2)),
            layers.Flatten(),
            layers.Dense(64, activation='relu'),
            ])
```

Epochs = 15

image = 224x224
augmented_data (4x1000)
learning_rate = 0.001





True label \ Predicted label	Blotch_Apple	Normal_Apple	Rot_Apple	Scab_Apple
Blotch_Apple	13	3	0	14
Normal_Apple	3	9	5	7
Rot_Apple	1	2	33	2
Scab_Apple	2	2	3	21

Confusion matrix 76/120 **0.6333**

Loss	0.3122	**Accuracy**	**0.8881**
Val_Loss	0.2071	**Val_Accuracy**	**0.9287**
Test_Loss 1.4600		**Test_Accuracy**	**0.6333**

COMMENTS

Model 11 rerun met cel:
```
train_ds = train_ds.map(
lambda x, y: (data_augmentation(x, training=True), y)
).prefetch(buffer_size=tf.data.AUTOTUNE)
```
geactiveerd.

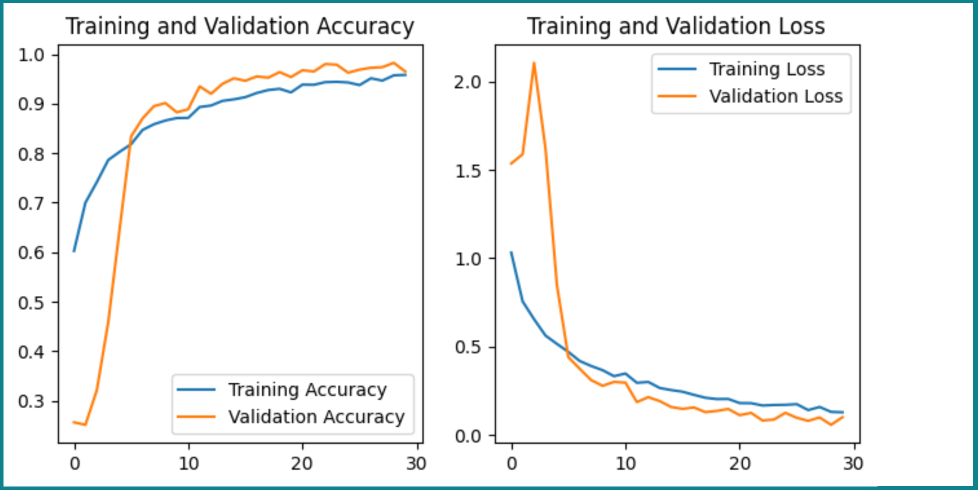
Model 31: 25 Epochs == 0.6333

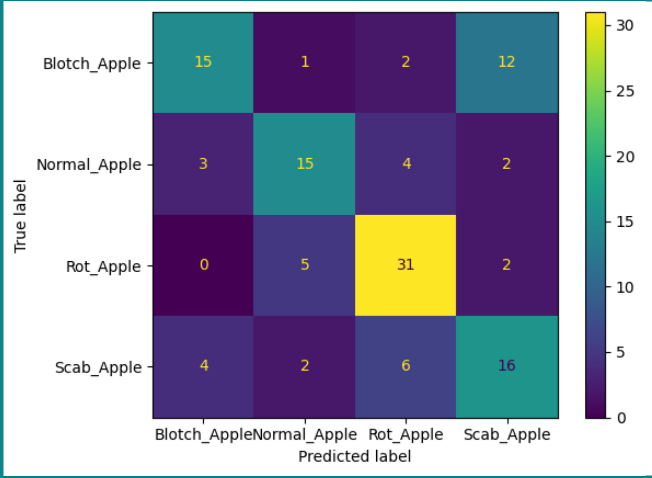
CNN (Apple Disease Classification) MODEL 12

```
model = models.Sequential([
            resize_and_rescale,
            data_augmentation,
                        RandomFlip("horizontal_and_vertical")
                        RandomRotation(0.3),
                        RandomZoom(0.4)
            layers.Conv2D(32, (3,3), activation='relu', input_shape=input_shape),
            tf.keras.layers.BatchNormalization(),
            layers.MaxPooling2D((2,2)),
            layers.Conv2D(64, (3,3), activation='relu'),
            tf.keras.layers.BatchNormalization(),
            layers.MaxPooling2D((2,2)),
            layers.Conv2D(64, (3,3), activation='relu'),
            tf.keras.layers.BatchNormalization(),
            layers.MaxPooling2D((2,2)),
            layers.Conv2D(128, (3,3), activation='relu'),
            tf.keras.layers.BatchNormalization(),
            layers.MaxPooling2D((2,2)),
            layers.Conv2D(128, (3,3), activation='relu'),
            tf.keras.layers.BatchNormalization(),
            layers.MaxPooling2D((2,2)),
            layers.Conv2D(256, (3,3), activation='relu'),
            tf.keras.layers.BatchNormalization(),
            layers.MaxPooling2D((2,2)),
            layers.Flatten(),
            layers.Dense(64, activation='relu'),
            ])
```

Epochs = 30

image = 224x224
augmented_data (4x1000)
learning_rate = 0.0001





	Blotch_Apple	Normal_Apple	Rot_Apple	Scab_Apple
Blotch_Apple	15	1	2	12
Normal_Apple	3	15	4	2
Rot_Apple	0	5	31	2
Scab_Apple	4	2	6	16

Confusion matrix 77/125 **0.6417**

Loss	0.1278	**Accuracy**	**0.9581**
Val_Loss	0.0996	**Val_Accuracy**	**0.965**
Test_Loss	1.7314	**Test_Accuracy**	**0.6417**

COMMENTS

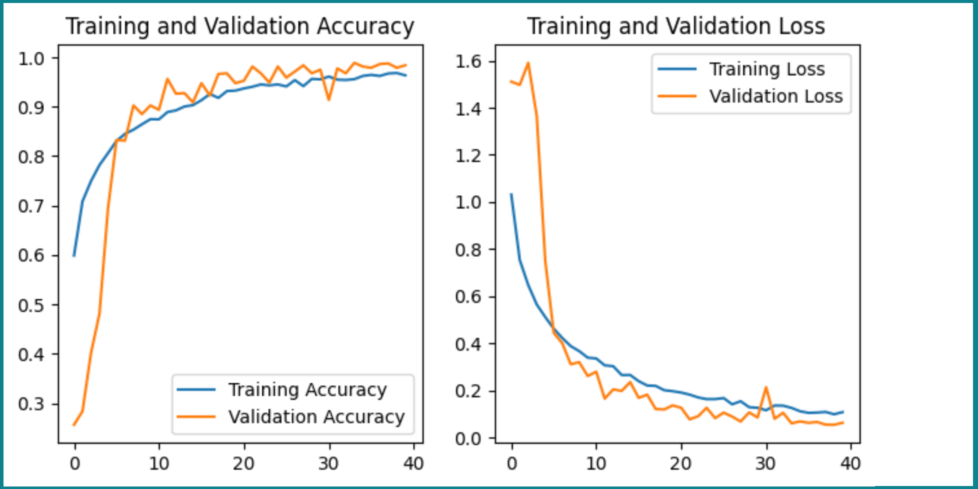
Meer Epochs (30) en een lagere Learning Rate (0.0001) geven een beter resultaat.

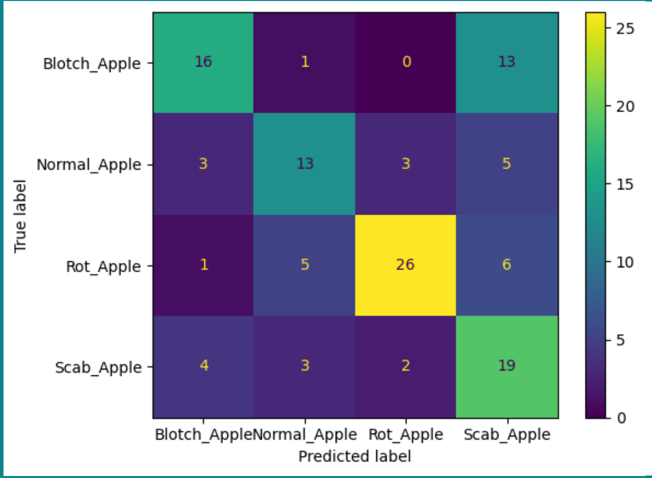
CNN (Apple Disease Classification) MODEL 13

```
model = models.Sequential([
            resize_and_rescale,
            data_augmentation,
                        RandomFlip("horizontal_and_vertical")
                        RandomRotation(0.3),
                        RandomZoom(0.4)
            layers.Conv2D(32, (3,3), activation='relu', input_shape=input_shape),
            tf.keras.layers.BatchNormalization(),
            layers.MaxPooling2D((2,2)),
            layers.Conv2D(64, (3,3), activation='relu'),
            tf.keras.layers.BatchNormalization(),
            layers.MaxPooling2D((2,2)),
            layers.Conv2D(64, (3,3), activation='relu'),
            tf.keras.layers.BatchNormalization(),
            layers.MaxPooling2D((2,2)),
            layers.Conv2D(128, (3,3), activation='relu'),
            tf.keras.layers.BatchNormalization(),
            layers.MaxPooling2D((2,2)),
            layers.Conv2D(128, (3,3), activation='relu'),
            tf.keras.layers.BatchNormalization(),
            layers.MaxPooling2D((2,2)),
            layers.Conv2D(256, (3,3), activation='relu'),
            tf.keras.layers.BatchNormalization(),
            layers.MaxPooling2D((2,2)),
            layers.Flatten(),
            layers.Dense(64, activation='relu'),
            ])
```

Epochs = 40

image = 224x224
augmented_data (4x1000)
learning_rate = 0.0001





	Blotch_Apple	Normal_Apple	Rot_Apple	Scab_Apple
Blotch_Apple	16	1	0	13
Normal_Apple	3	13	3	5
Rot_Apple	1	5	26	6
Scab_Apple	4	3	2	19

Confusion matrix 74/125 **0.6167**

COMMENTS

10 Extra Epochs maken het model niet beter.

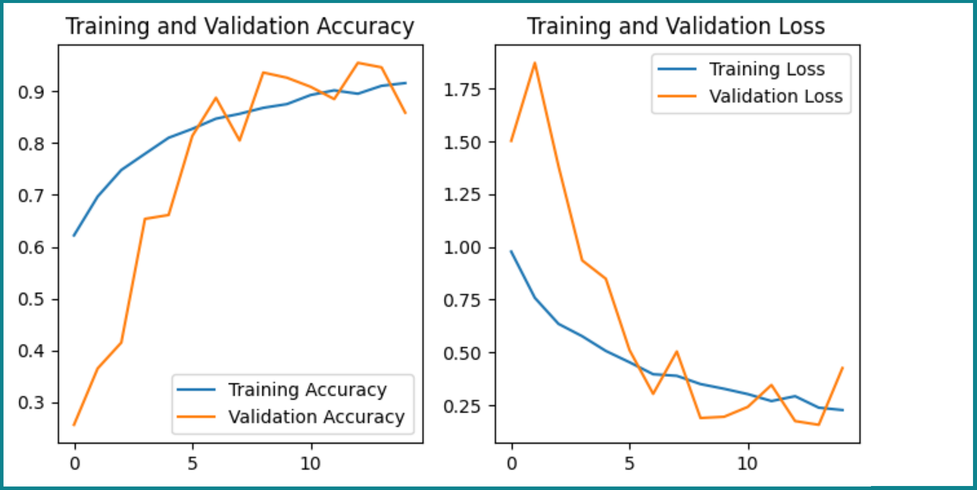
Loss	0.1075	**Accuracy**	**0.9634**
Val_Loss	0.0625	**Val_Accuracy**	**0.9837**
Test_Loss	1.4849	**Test_Accuracy**	**0.6167**

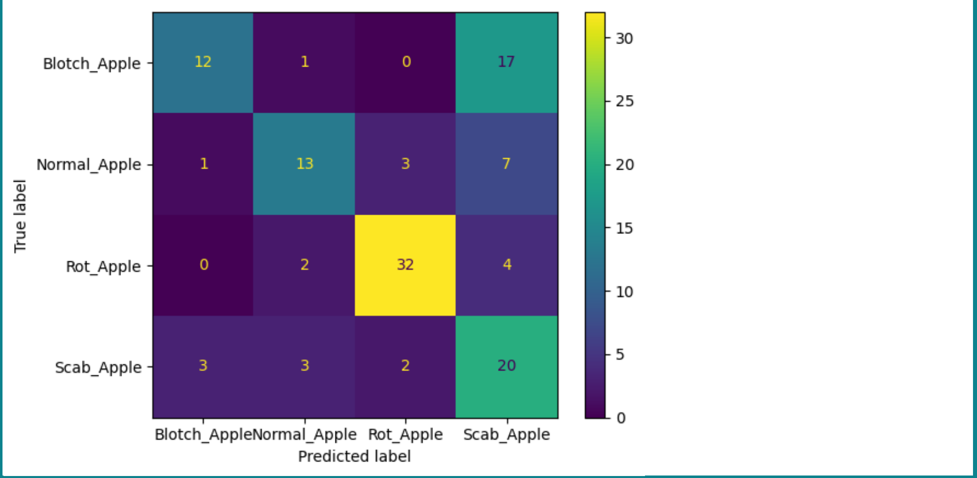
CNN (Apple Disease Classification) MODEL 29

```
model = models.Sequential([
            resize_and_rescale,
            data_augmentation,
                        RandomFlip("horizontal_and_vertical")
                        RandomRotation(0.3),
                        RandomZoom(0.4)
            layers.Conv2D(32, (3,3), activation='relu', input_shape=input_shape),
            tf.keras.layers.BatchNormalization(),
            layers.MaxPooling2D((2,2)),
            layers.Conv2D(64, (3,3), activation='relu'),
            tf.keras.layers.BatchNormalization(),
            layers.MaxPooling2D((2,2)),
            layers.Conv2D(64, (3,3), activation='relu'),
            tf.keras.layers.BatchNormalization(),
            layers.MaxPooling2D((2,2)),
            layers.Conv2D(128, (3,3), activation='relu'),
            tf.keras.layers.BatchNormalization(),
            layers.MaxPooling2D((2,2)),
            layers.Conv2D(128, (3,3), activation='relu'),
            tf.keras.layers.BatchNormalization(),
            layers.MaxPooling2D((2,2)),
            layers.Conv2D(256, (3,3), activation='relu'),
            tf.keras.layers.BatchNormalization(),
            layers.MaxPooling2D((2,2)),
            layers.Flatten(),
            layers.Dense(64, activation='relu'),
            ])
```

Epochs = 15

image = 224x224
augmented_data (4x1000)
learning_rate = 0.001





	Blotch_Apple	Normal_Apple	Rot_Apple	Scab_Apple
Blotch_Apple	12	1	0	17
Normal_Apple	1	13	3	7
Rot_Apple	0	2	32	4
Scab_Apple	3	3	2	20

Confusion matrix 77/120 **0.6417**

Loss	0.2261	**Accuracy**	**0.9159**
Val_Loss	0.4252	**Val_Accuracy**	**0.8587**
Test_Loss	1.8896	**Test_Accuracy**	**0.6417**

COMMENTS

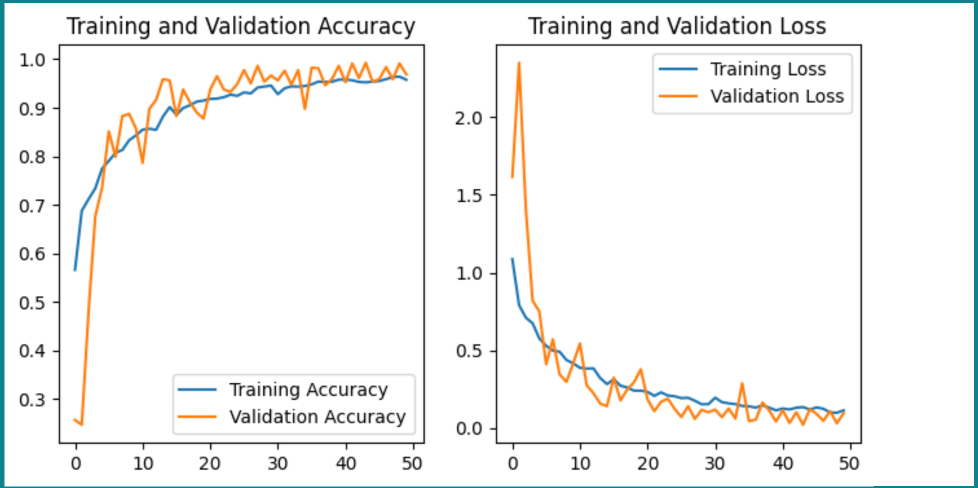
Zelfde set up als Model 11, veel betere score.
Nu ook matige Validation Accuracy en onstabiele Validation lijn, maar wel goede eindscore.

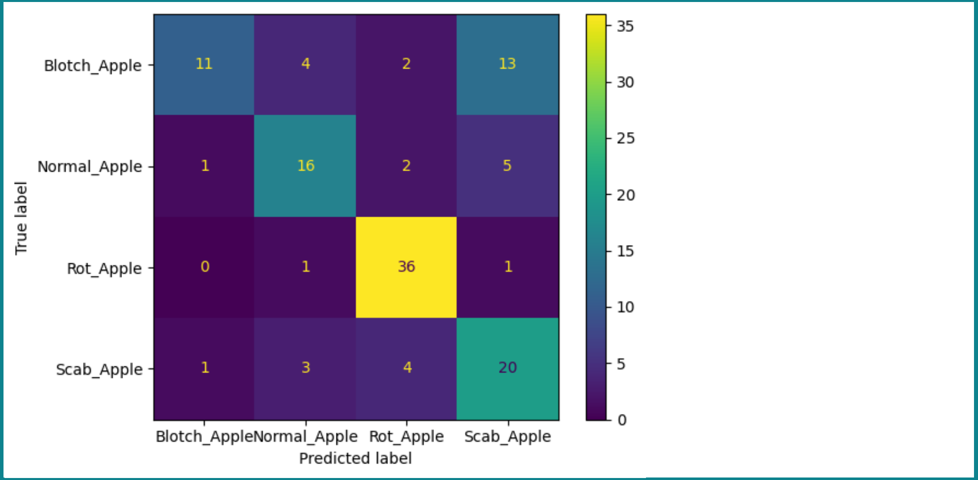
CNN (Apple Disease Classification) MODEL 32

```
model = models.Sequential([
            resize_and_rescale,
            data_augmentation,
                        RandomFlip("horizontal_and_vertical")
                        RandomRotation(0.3),
                        RandomZoom(0.4)
            layers.Conv2D(32, (3,3), activation='relu', input_shape=input_shape),
            tf.keras.layers.BatchNormalization(),
            layers.MaxPooling2D((2,2)),
            layers.Conv2D(64, (3,3), activation='relu'),
            tf.keras.layers.BatchNormalization(),
            layers.MaxPooling2D((2,2)),
            layers.Conv2D(64, (3,3), activation='relu'),
            tf.keras.layers.BatchNormalization(),
            layers.MaxPooling2D((2,2)),
            layers.Conv2D(128, (3,3), activation='relu'),
            tf.keras.layers.BatchNormalization(),
            layers.MaxPooling2D((2,2)),
            layers.Conv2D(128, (3,3), activation='relu'),
            tf.keras.layers.BatchNormalization(),
            layers.MaxPooling2D((2,2)),
            layers.Conv2D(256, (3,3), activation='relu'),
            tf.keras.layers.BatchNormalization(),
            layers.MaxPooling2D((2,2)),
            layers.Flatten(),
            layers.Dense(64, activation='relu'),
            ])
```

Epochs = 50

image = 224x224
augmented_data (4x1000)
learning_rate = 0.001





True label \ Predicted label	Blotch_Apple	Normal_Apple	Rot_Apple	Scab_Apple
Blotch_Apple	11	4	2	13
Normal_Apple	1	16	2	5
Rot_Apple	0	1	36	1
Scab_Apple	1	3	4	20

Confusion matrix 83/120 **0.6917**

Loss	0.1140	**Accuracy**	**0.9575**
Val_Loss	0.0960	**Val_Accuracy**	**0.9688**
Test_Loss	2.1467	**Test_Accuracy**	**0.6917**

COMMENTS

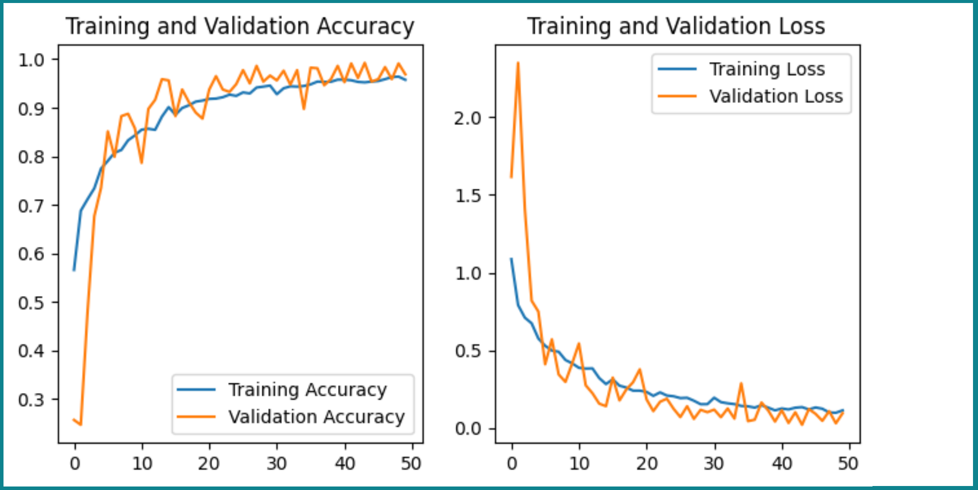
Zelfde set up als Model 11, veel betere score.
Nu ook matige Validation Accuracy en onstabiele Validation lijn.

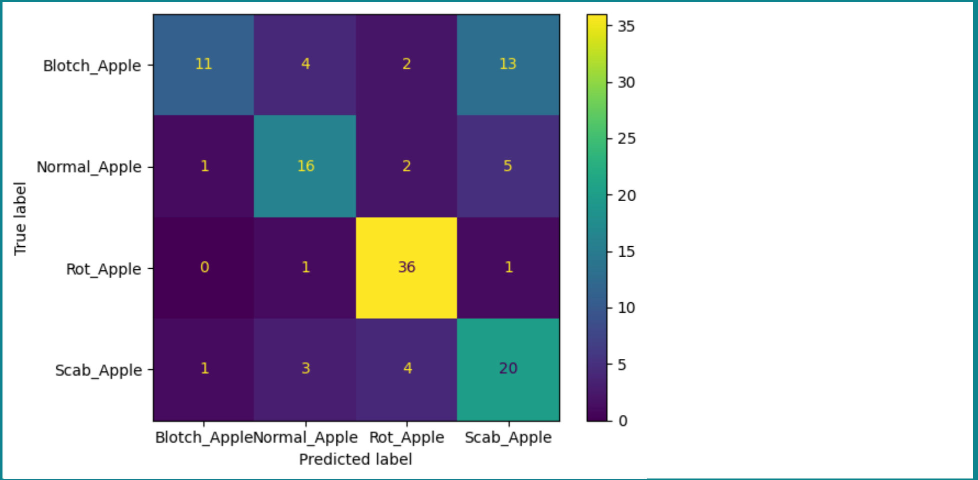
CNN (Apple Disease Classification) MODEL 33

```
model = models.Sequential([
            resize_and_rescale,
            data_augmentation,
                        RandomFlip("horizontal_and_vertical")
                        RandomRotation(0.3),
                        RandomZoom(0.4)
                        RandomContrast(factor=0.5, seed=None, name=None)
            layers.Conv2D(32, (3,3), activation='relu', input_shape=input_shape),
            tf.keras.layers.BatchNormalization(),
            layers.MaxPooling2D((2,2)),
            layers.Conv2D(64, (3,3), activation='relu'),
            tf.keras.layers.BatchNormalization(),
            layers.MaxPooling2D((2,2)),
            layers.Conv2D(64, (3,3), activation='relu'),
            tf.keras.layers.BatchNormalization(),
            layers.MaxPooling2D((2,2)),
            layers.Conv2D(128, (3,3), activation='relu'),
            tf.keras.layers.BatchNormalization(),
            layers.MaxPooling2D((2,2)),
            layers.Conv2D(128, (3,3), activation='relu'),
            tf.keras.layers.BatchNormalization(),
            layers.MaxPooling2D((2,2)),
            layers.Conv2D(256, (3,3), activation='relu'),
            tf.keras.layers.BatchNormalization(),
            layers.MaxPooling2D((2,2)),
            layers.Flatten(),
            layers.Dense(64, activation='relu'),
            ])
```

Epochs = 50

image = 224x224
augmented_data (4x1000)
learning_rate = 0.001





True label \ Predicted label	Blotch_Apple	Normal_Apple	Rot_Apple	Scab_Apple
Blotch_Apple	11	4	2	13
Normal_Apple	1	16	2	5
Rot_Apple	0	1	36	1
Scab_Apple	1	3	4	20

Confusion matrix 83/120 **0.6833**

COMMENTS

Zelfde set up als Model 11, met extra Epochs (50 totaal) veel betere score.
Onrustige Validation lijn.

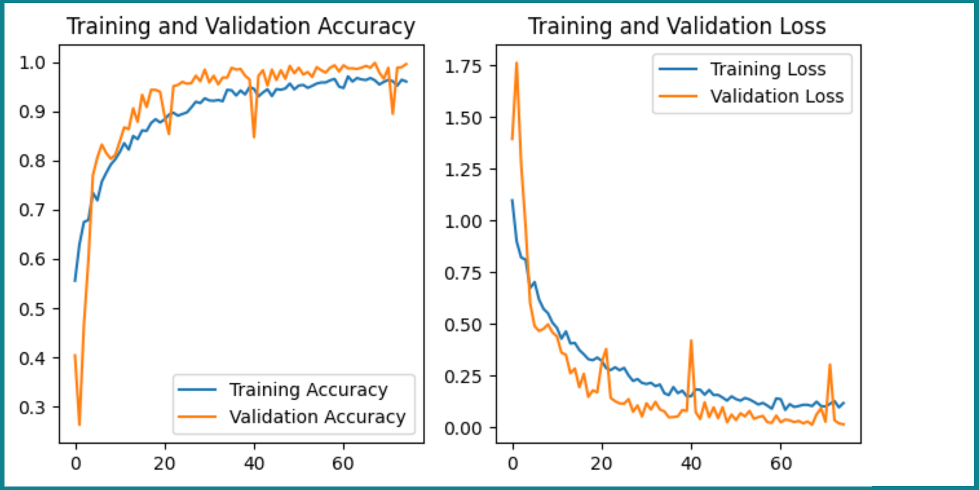
Loss	0.1689	**Accuracy**	**0.9347**
Val_Loss	0.0513	**Val_Accuracy**	**0.9850**
Test_Loss	1.4455	**Test_Accuracy**	**0.6833**

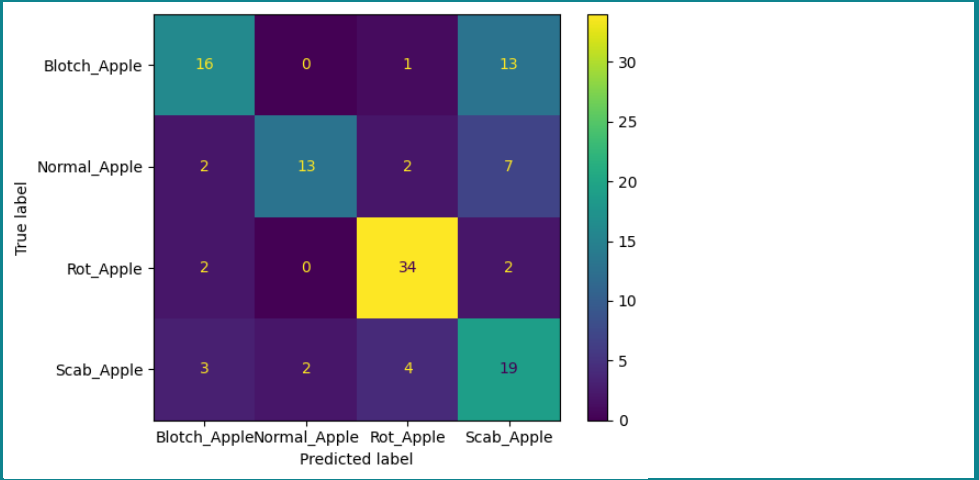
CNN (Apple Disease Classification) MODEL 34

```
model = models.Sequential([
             resize_and_rescale,
             data_augmentation,
                         RandomFlip("horizontal_and_vertical")
                         RandomRotation(0.3),
                         RandomZoom(0.4)
                         RandomContrast(factor=0.5, seed=None, name=None)
             layers.Conv2D(32, (3,3), activation='relu', input_shape=input_shape),
             tf.keras.layers.BatchNormalization(),
             layers.MaxPooling2D((2,2)),
             layers.Conv2D(64, (3,3), activation='relu'),
             tf.keras.layers.BatchNormalization(),
             layers.MaxPooling2D((2,2)),
             layers.Conv2D(64, (3,3), activation='relu'),
             tf.keras.layers.BatchNormalization(),
             layers.MaxPooling2D((2,2)),
             layers.Conv2D(128, (3,3), activation='relu'),
             tf.keras.layers.BatchNormalization(),
             layers.MaxPooling2D((2,2)),
             layers.Conv2D(128, (3,3), activation='relu'),
             tf.keras.layers.BatchNormalization(),
             layers.MaxPooling2D((2,2)),
             layers.Conv2D(256, (3,3), activation='relu'),
             tf.keras.layers.BatchNormalization(),
             layers.MaxPooling2D((2,2)),
             layers.Flatten(),
             layers.Dense(64, activation='relu'),
             ])
```

Epochs = 75

image = 200x200
augmented_data (4x1000)
learning_rate = 0.001





True label \ Predicted label	Blotch_Apple	Normal_Apple	Rot_Apple	Scab_Apple
Blotch_Apple	16	0	1	13
Normal_Apple	2	13	2	7
Rot_Apple	2	0	34	2
Scab_Apple	3	2	4	19

Confusion matrix 82/120 **0.6833**

COMMENTS

RandomContrast toegvoegd in de hoop dat dat het verschil tussen Blotch en Scab beter "zichtbaar" zou maken. Geen direct effect te bekenen. Score blijft redelijk.
Kleinere afbeeldingen (200x200) kan niet, model loopt vast.

De tijd die inmiddels nodig is om het model te trainen loopt op naar 1 uur.

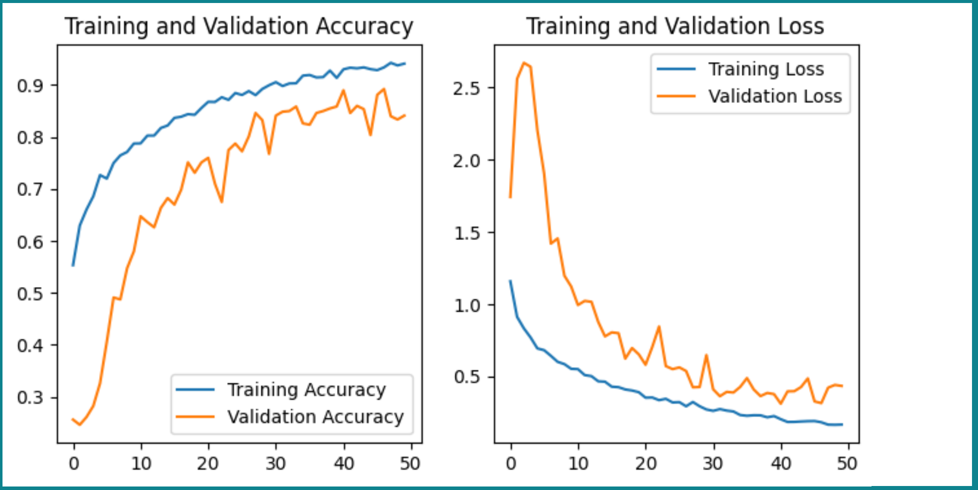
Loss	0.1188	**Accuracy**	**0.9606**
Val_Loss	0.0159	**Val_Accuracy**	**0.9962**
Test_Loss	1.7696	**Test_Accuracy**	**0.6833**

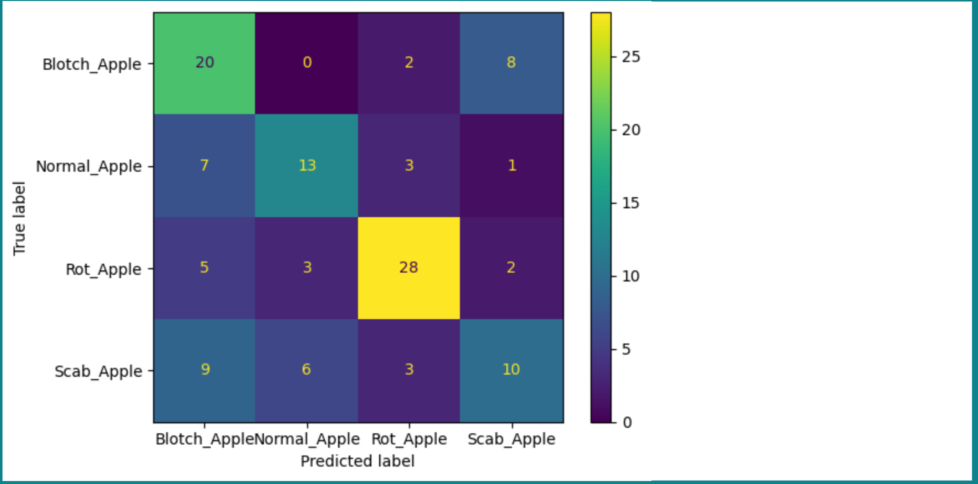
CNN (Apple Disease Classification) MODEL 15

```
model = models.Sequential([
          resize_and_rescale,
          data_augmentation,
                    RandomFlip("horizontal_and_vertical")
                    RandomRotation(0.3),
                    RandomZoom(0.4)
          layers.Conv2D(32, (3,3), activation='relu', input_shape=input_shape),
          tf.keras.layers.BatchNormalization(),
          layers.MaxPooling2D((2,2)),
          layers.Dropout(0.2),
          layers.Conv2D(64, (3,3), activation='relu'),
          tf.keras.layers.BatchNormalization(),
          layers.MaxPooling2D((2,2)),
          layers.Conv2D(64, (3,3), activation='relu'),
          tf.keras.layers.BatchNormalization(),
          layers.MaxPooling2D((2,2)),
          layers.Dropout(0.2),
          layers.Conv2D(128, (3,3), activation='relu'),
          tf.keras.layers.BatchNormalization(),
          layers.MaxPooling2D((2,2)),
          layers.Conv2D(128, (3,3), activation='relu'),
          tf.keras.layers.BatchNormalization(),
          layers.MaxPooling2D((2,2)),
          layers.Dropout(0.2),
          layers.Conv2D(256, (3,3), activation='relu'),
          tf.keras.layers.BatchNormalization(),
          layers.MaxPooling2D((2,2)),
          layers.Flatten(),
          layers.Dense(64, activation='relu'),
          layers.Dense(n_classes, activation='softmax')
          ])
```

Epochs = 50

image = 224x224
augmented_data (4x1000)
learning_rate = 0.0001





True label \ Predicted label	Blotch_Apple	Normal_Apple	Rot_Apple	Scab_Apple
Blotch_Apple	20	0	2	8
Normal_Apple	7	13	3	1
Rot_Apple	5	3	28	2
Scab_Apple	9	6	3	10

Confusion matrix 71/120 **0.5917**

Loss	0.167	**Accuracy**	**0.9413**
Val_Loss	0.4345	**Val_Accuracy**	**0.8413**
Test_Loss	1.3676	**Test_Accuracy**	**0.5917**

COMMENTS

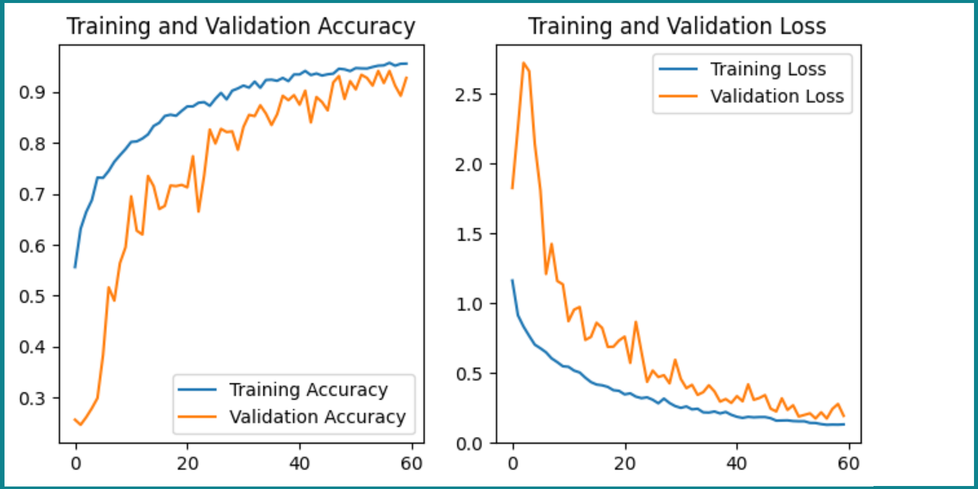
Meer Epochs (50) en enkele Dropout lagen tussen de blokken maken het model niet beter.
Training en Validation lopen gelijk op maar liggen verder van elkaar dan eerder.
Omdat er niet direct sprake was van overfittig voegt de Dropout mogelijk weinig toe.

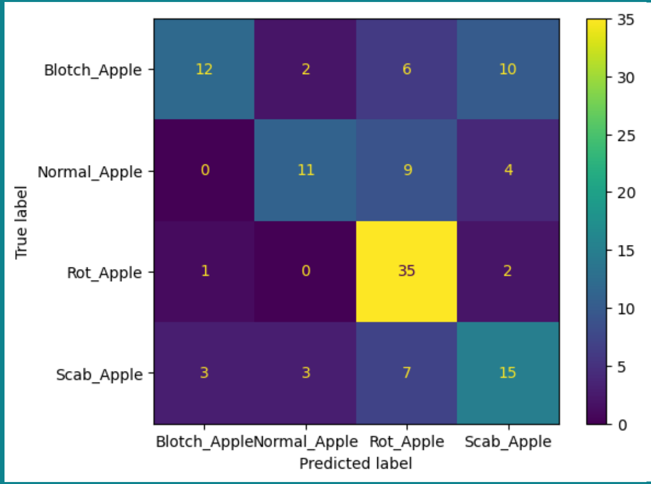
CNN (Apple Disease Classification) MODEL 16

```
model = models.Sequential([
          resize_and_rescale,
          data_augmentation,
                    RandomFlip("horizontal_and_vertical")
                    RandomRotation(0.3),
                    RandomZoom(0.4)
          layers.Conv2D(32, (3,3), activation='relu', input_shape=input_shape),
          tf.keras.layers.BatchNormalization(),
          layers.MaxPooling2D((2,2)),
          layers.Dropout(0.2),
          layers.Conv2D(64, (3,3), activation='relu'),
          tf.keras.layers.BatchNormalization(),
          layers.MaxPooling2D((2,2)),
          layers.Conv2D(64, (3,3), activation='relu'),
          tf.keras.layers.BatchNormalization(),
          layers.MaxPooling2D((2,2)),
          layers.Dropout(0.2),
          layers.Conv2D(128, (3,3), activation='relu'),
          tf.keras.layers.BatchNormalization(),
          layers.MaxPooling2D((2,2)),
          layers.Conv2D(128, (3,3), activation='relu'),
          tf.keras.layers.BatchNormalization(),
          layers.MaxPooling2D((2,2)),
          layers.Dropout(0.2),
          layers.Conv2D(256, (3,3), activation='relu'),
          tf.keras.layers.BatchNormalization(),
          layers.MaxPooling2D((2,2)),
          layers.Flatten(),
          layers.Dense(64, activation='relu'),
          layers.Dense(n_classes, activation='softmax')
          ])
```

Epochs = 60

image = 224x224
augmented_data (4x1000)
learning_rate = 0.0001





	Blotch_Apple	Normal_Apple	Rot_Apple	Scab_Apple
Blotch_Apple	12	2	6	10
Normal_Apple	0	11	9	4
Rot_Apple	1	0	35	2
Scab_Apple	3	3	7	15

Confusion matrix 73/120 **0.6083**

Loss	0.1301	**Accuracy**	**0.9556**
Val_Loss	0.1915	**Val_Accuracy**	**0.9275**
Test_Loss	1.5059	**Test_Accuracy**	**0.6083**

COMMENTS

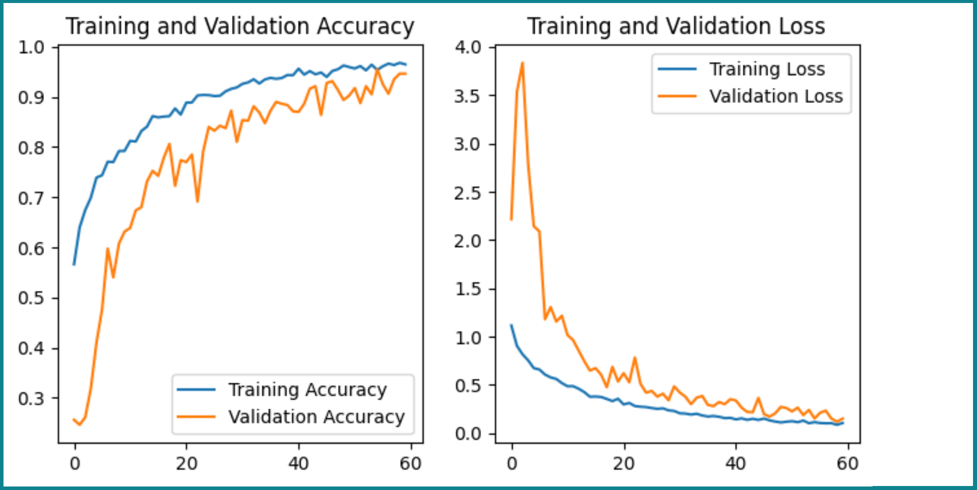
Meer Epochs (60) zorgen voor een iets verbeterde score, maar niet de beste.

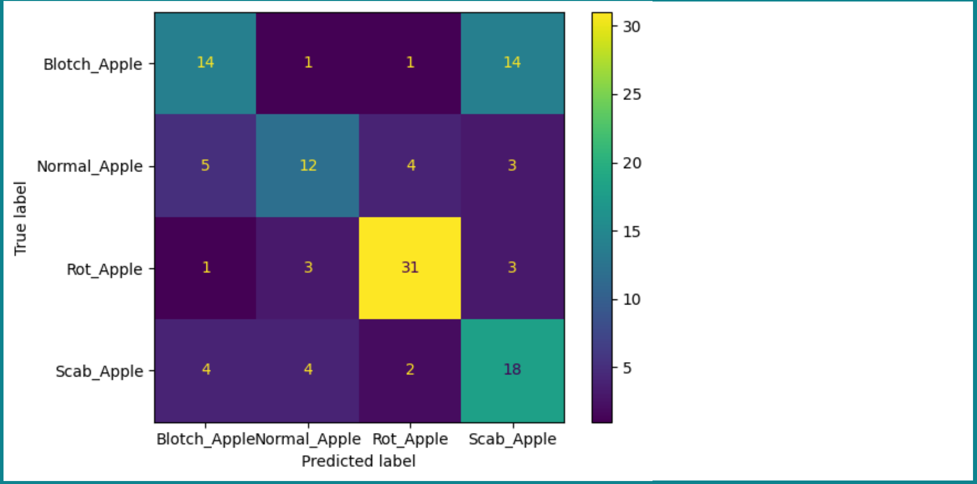
CNN (Apple Disease Classification) MODEL 17

```
model = models.Sequential([
            resize_and_rescale,
            data_augmentation,
                        RandomFlip("horizontal_and_vertical")
                        RandomRotation(0.3),
                        RandomZoom(0.4)
            layers.Conv2D(32, (3,3), activation='relu', input_shape=input_shape),
            tf.keras.layers.BatchNormalization(),
            layers.MaxPooling2D((2,2)),
            layers.Dropout(0.2),
            layers.Conv2D(64, (3,3), padding='same', activation='relu'),
            tf.keras.layers.BatchNormalization(),
            layers.MaxPooling2D((2,2)),
            layers.Conv2D(64, (3,3), padding='same', activation='relu'),
            tf.keras.layers.BatchNormalization(),
            layers.MaxPooling2D((2,2)),
            layers.Dropout(0.2),
            layers.Conv2D(128, (3,3), padding='same', activation='relu'),
            tf.keras.layers.BatchNormalization(),
            layers.MaxPooling2D((2,2)),
            layers.Conv2D(128, (3,3), padding='same', activation='relu'),
            tf.keras.layers.BatchNormalization(),
            layers.MaxPooling2D((2,2)),
            layers.Dropout(0.2),
            layers.Conv2D(256, (3,3), padding='same', activatio
            tf.keras.layers.BatchNormalization(),
            layers.MaxPooling2D((2,2)),
            layers.Flatten(),
            layers.Dense(64, activation='relu'),
            layers.Dense(n_classes, activation='softmax')
            ])
```

Epochs = 60

image = 224x224
augmented_data (4x1000)
learning_rate = 0.0001





Confusion matrix 75/120 **0.6250**

COMMENTS

Meer Padding wordt de score iets beter, ook al staan de meeste appels in de afbeeldingen in het midden.

Loss	0.1046	**Accuracy**	**0.9647**
Val_Loss	0.1498	**Val_Accuracy**	**0.9463**
Test_Loss	1.5716	**Test_Accuracy**	**0.6250**