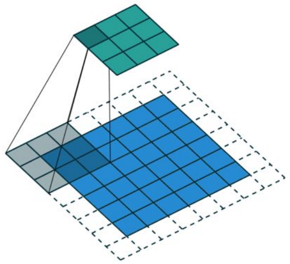
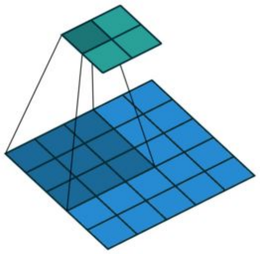




Leaf classification

CNN supervised machine learning

Dataset description

- 54303 images
- size: 256x256
- RGB space
- 13 species
- 38 categories
- healthy and unhealthy plant leaves
- unbalanced categories

- Apple
- Blueberry
- Cherry
- Corn
- Grape
- Orange
- Peach
- Pepper
- Potato
- Raspberry
- Squash
- Strawberry
- Tomato
- Empty background

Dataset: index of labels

1.Apple scab

2.Apple black rot

3.Apple cedar apple rust

4.Apple healthy

5.Background without leaves

6.Blueberry healthy

7.Cherry powdery mildew

8.Cherry healthy

9.Corn gray leaf spot

10.Corn common rust

11.Corn northern leaf blight

12.Corn healthy

13.Grape black rot

14.Grape black measles

15.Grape leaf blight

16.Grape healthy

17.Orange haunglongbing

18.Peach bacterial spot

19.Peach healthy

20.Pepper bacterial spot

21.Pepper healthy

22.Potato early blight

23.Potato healthy

24.Potato late blight

25.Raspberry healthy

26.Soybean healthy

27.Squash powdery mildew

28.Strawberry healthy

29.Strawberry leaf scorch

30.Tomato bacterial spot

31.Tomato early blight

32.Tomato healthy

33.Tomato late blight

34.Tomato leaf mold

35.Tomato septoria leaf spot

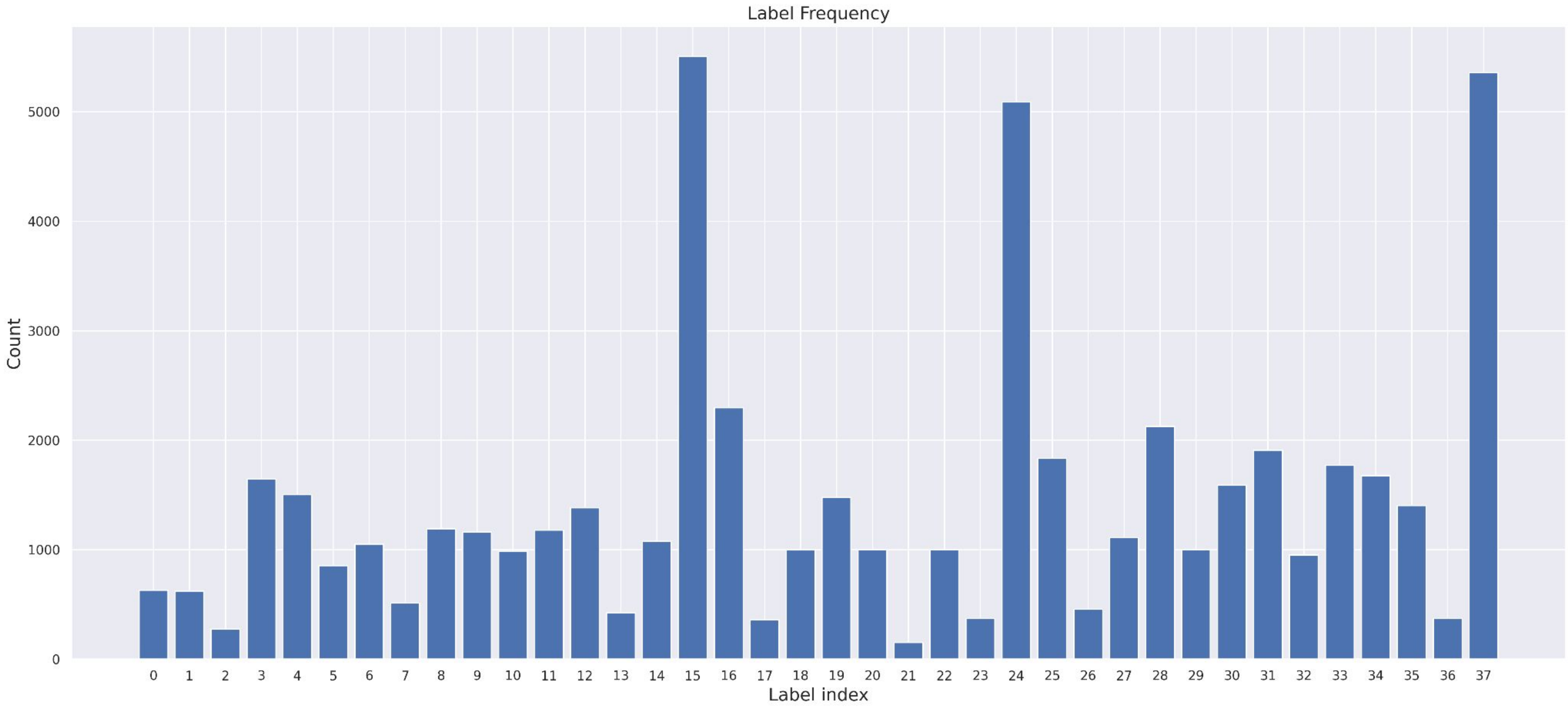
36.Tomato spider mites two-spotted spider mite

37.Tomato target spot

38.Tomato mosaic virus

39.Tomato yellow leaf curl virus

Dataset distribution



Dataset sample images


Tomato - Target Spot


Orange - Haunglongbing (Citrus greening)


Tomato - Late blight


Potato - Early blight


Apple - healthy


Apple - Cedar apple rust


Pepper, bell - healthy


Tomato - Spider mites Two-spotted spider mite


Tomato - Bacterial spot


Tomato - Late blight


Squash - Powdery mildew


Tomato - healthy


Cherry - Powdery mildew


Tomato - Spider mites Two-spotted spider mite


Tomato - Tomato mosaic virus

Dataset processing

- Split:
  - train 80%
  - validation 15%
  - test 5%
- Random data augmentation:
  - brightness
  - contrast
  - flip
- Class weighting
- Image scaling surprisingly counterproductive! It was not implemented.

Class weighting

$$\text{class weight} = \frac{n_{samples}}{n_{classes} \times \text{bincount}(y)}$$

Class unbalance is compensated through a class weight using the values of:

- `n_samples`
- `n_classes`
- `bincount(y)`: a function that counts the number of occurrences of each value in an array y
- y: the array of the data labels

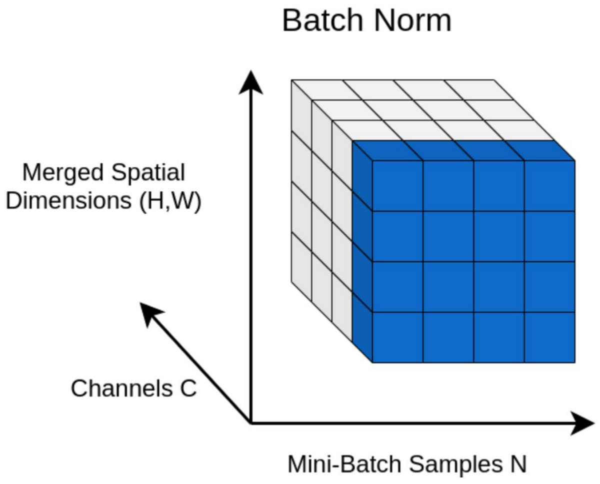
Source: https://scikit-learn.org/stable/modules/generated/sklearn.utils.class_weight.compute_class_weight.html

The model

- Custom architecture composition:
  - Three CNN blocks
  - A classification block
- Loss function: Sparse Categorical Cross-entropy, *because data is discrete multiclass*
- Optimizer: Adam, *computationally efficient, has little memory requirement, invariant to diagonal rescaling of gradients, and is well suited for problems that are large in terms of data/parameters*

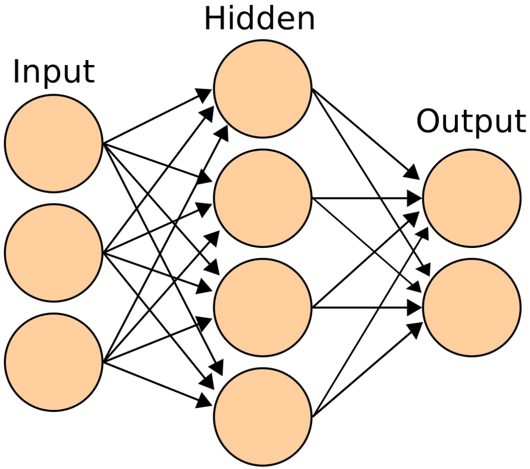
CNN block

- 2D convolutional layer
- respectively 16, 32 and 64 filters per block
- Kernel size: 3
- ReLU activation
- L2 kernel regularizer
- stride equal to 2 for image size reduction
- Batch Normalization
- Dropout with probability 0.25 to handle overfitting



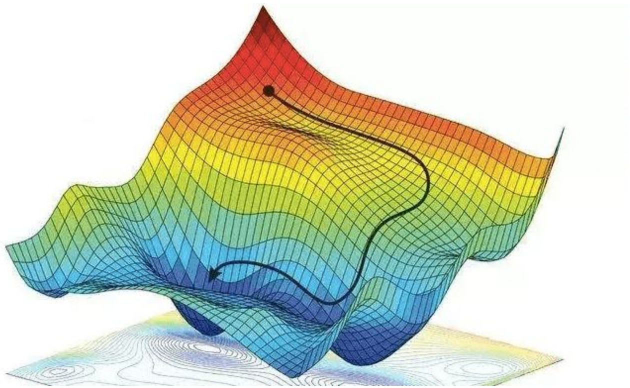
Classification block

- Composition: two feedforward, fully connected MLP
- Dimensions: 64 and 38 (as number of categories)
- Dropout with probability 0.5



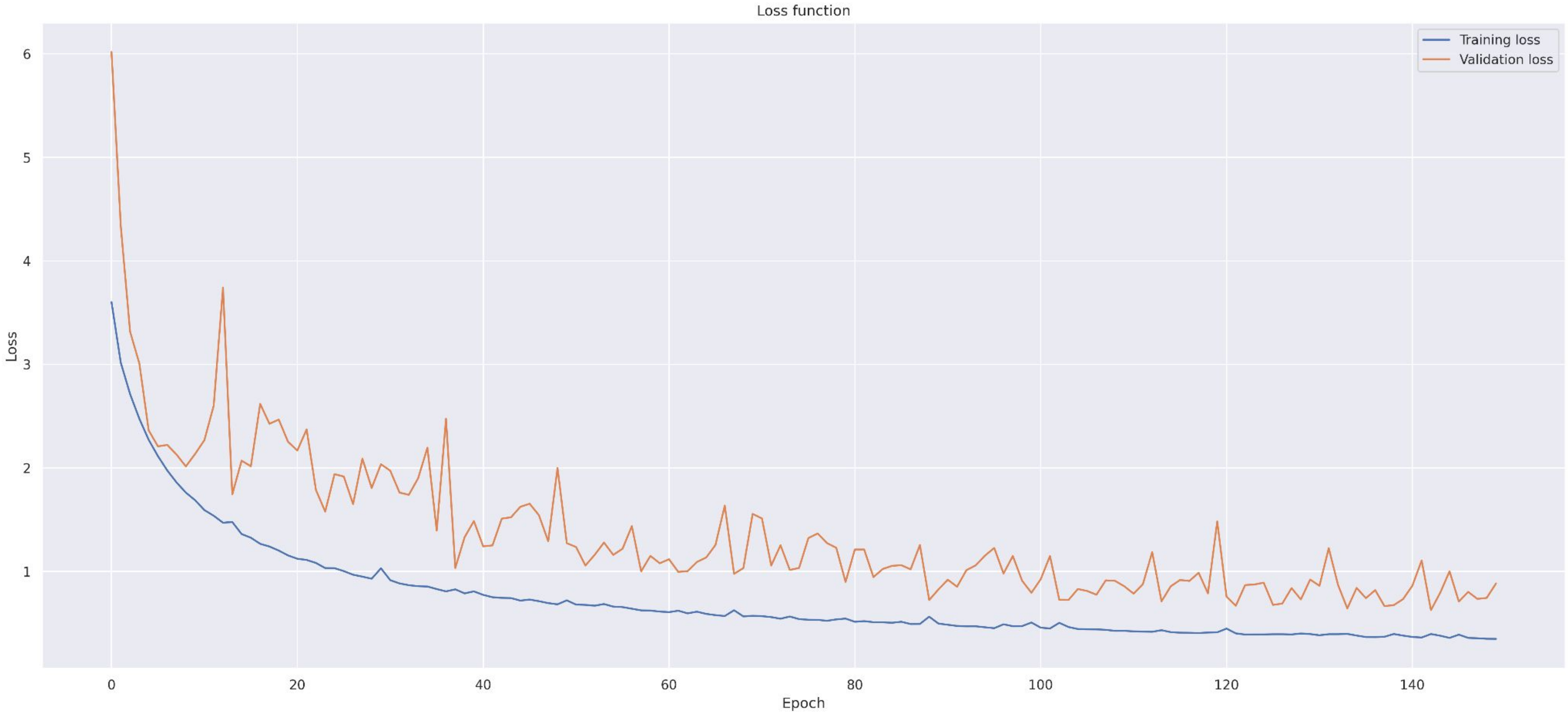
The optimizer: Adam

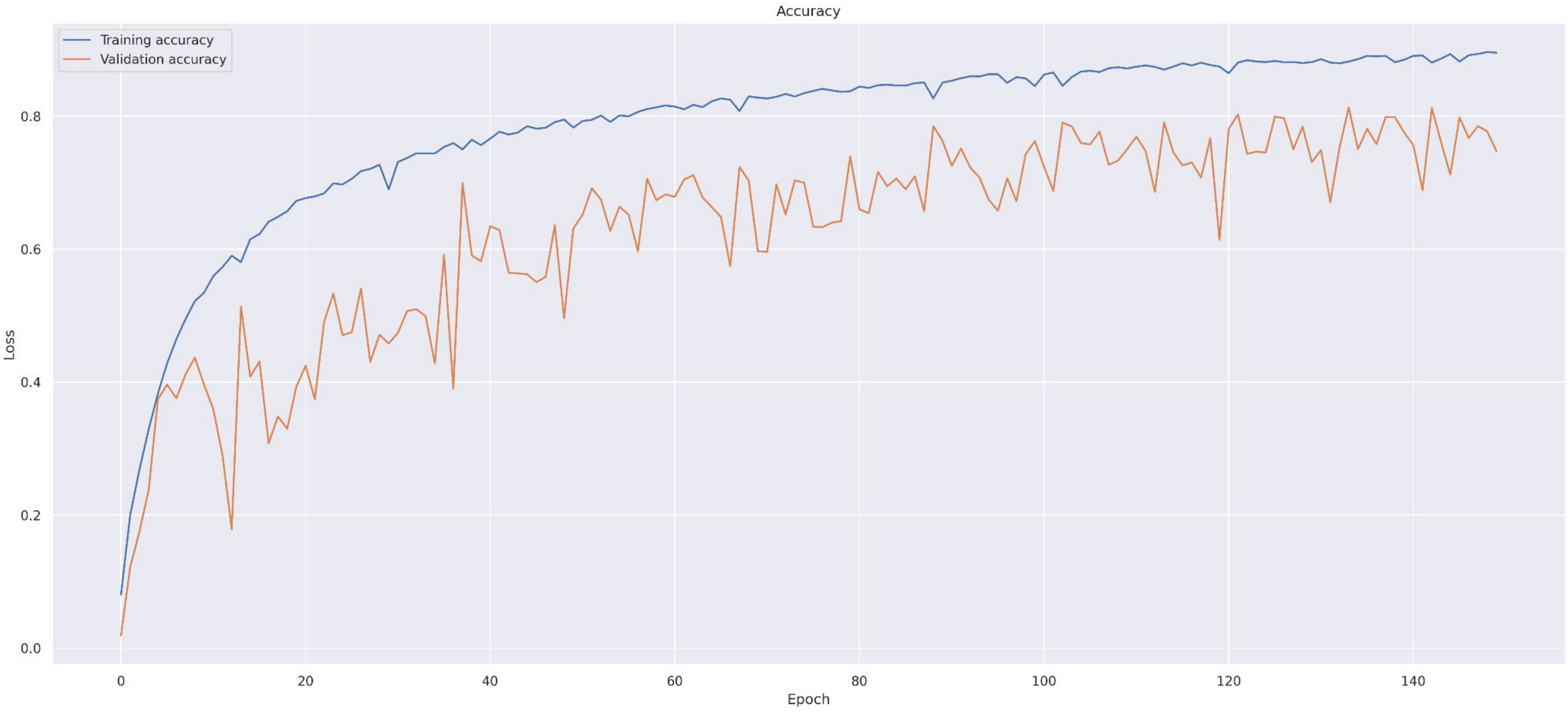
- stands for: adaptive moment estimation
- basically an extended version of stochastic gradient descent
- learning rate: 10e-4
- decay = learning rate / number of epochs



Model training

- Trained for 150 epochs
- 30,662 number of parameters
- Metric: Sparse Categorical Accuracy
- Hardware: Intel Xeon CPU @2.20 GHz, 13 GB RAM, Tesla P100 accelerator, and 12 GB GDDR5 VRAM
- Training time is circa one hour, with 20 seconds circa for each epoch
- Test performance: loss: 0.8939 - sparse categorical accuracy: 0.793


Loss function
Training loss
Validation loss
Loss
Epoch
0
1
2
3
4
5
6
0
20
40
60
80
100
120
140


Accuracy
Training accuracy
Validation accuracy
Loss
0.0
0.2
0.4
0.6
0.8
0
20
40
60
80
100
120
140
Epoch

References

- Dataset source
- Machine learning software library
- Jupyter notebook of the project