Exercise 3 - Deep Learning

Machine Learning

2020/2021 | Technische Universität Wien

Grupo 06
Diogo Braga (E12007525)
Enrico Coluccia (E12005483)
Matthias Kiss (01218325)

Datasets

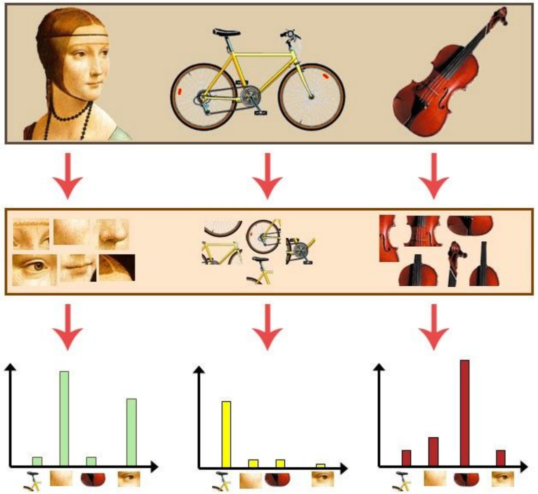
- **CIFAR-10** is a data set that consists of 60,000 32x32 **colour images** divided in 10 classes (e.g. dog, frog, truck, airplane), with 6,000 images per class. There are 50,000 training images and 10,000 test images.
- **Fashion MNIST** is a data set of Zalando's article images consisting of a training set of 60,000 examples and a test set of 10,000 examples. Each example is a 28x28 **grey-scale image**, associated with a label from 10 classes (e.g. coat, shirt).

Bag of Virtual Words (BoVW)- SIFT Based

SIFT (Scale Invariant Feature Detection) is a fast and efficient algorithm to find keypoints (and descriptors) for a given image. The algorithm uses only a monochrome intensity image.

Create a BoVW system:

- Compute the features (SIFT descriptors) for each image from the training set
- Cluster the feature with K-Means
- Create the histograms for each image in the training/test (based on cluster and SIFT)
- Create the model (MLP, KNN ..)
- Fit the model



SIFT based results

Dataset	Classifier	Accuracy	Precision	Recall	Time (s)
CIFAR 10	SVM	0.287	0.28	0.29	862
	K-Nearest Neighbors	0.180	0.18	0.18	107
	Decision Tree	0.173	0.17	0.17	2.5
	Multi Layer Perceptron	0.255	0.25	0.25	388
MNIST	SVM	0.664	0.66	0.66	3600
	K-Nearest Neighbor	0.476	0.50	0.48	726
	Decision Tree	0.590	0.58	0.58	206
	Multi Layer Perceptron	0.609	0.61	0.61	919

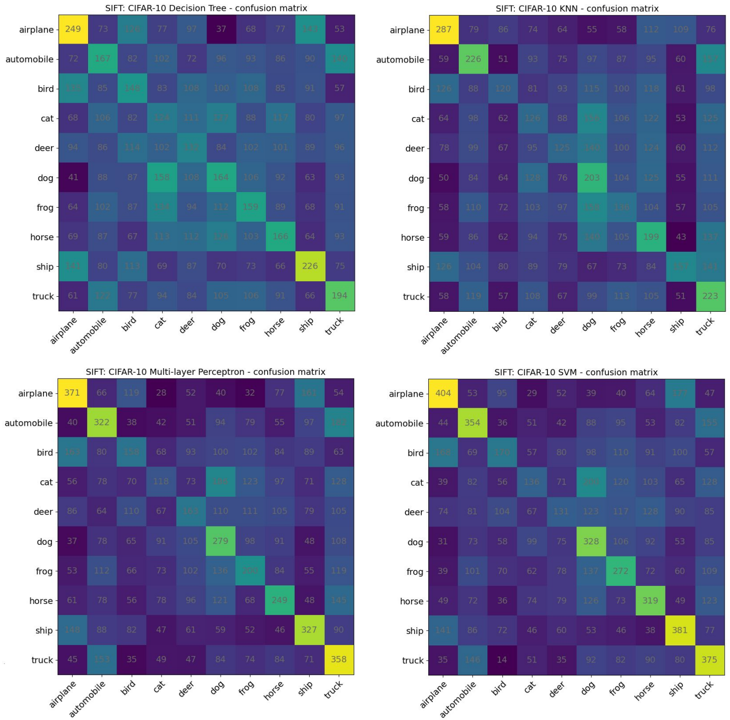
Table 3.1: Classifiers' performance for the two data sets.

Data set	Extraction	Clustering	Train Hist	Test Hist	Overall (s)
CIFAR 10	47	42	357	76	**522**
MNIST	57	57	295	60	**469**

- Bad performance over CIFAR-10 probably for the presence of background or too much differences in the classes.
- Reasonable performance for MNIST probably because the images are more "static" (e.g. same view for all the images)
- SVM performed better on both datasets (but very high running time), followed by MLP.
- <u>Running time:</u> BoVW system not too much "expensive". Approximately 10 minutes of computation for both datasets.

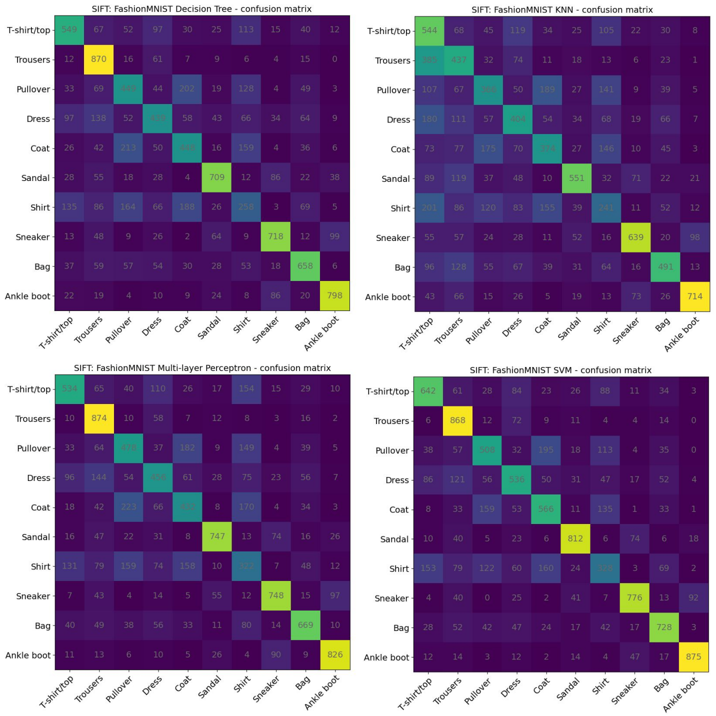
Confusion Matrix - CIFAR-10

- All the matrices have a common pattern. The most confused classes were Bird, Cat and Deer:
    - Cat was confused with Dog
    - Bird was confused with Airplane
- Reasonable performance with the remaining classes



Confusion Matrix - MNIST

- The BoVW system perform well on Fashion MNIST dataset
- All the classes with a good number of True Positives
- Common error over “Shirt” class:
  - Often confused with similar clothes as Coat, Pullover or T-Shirt



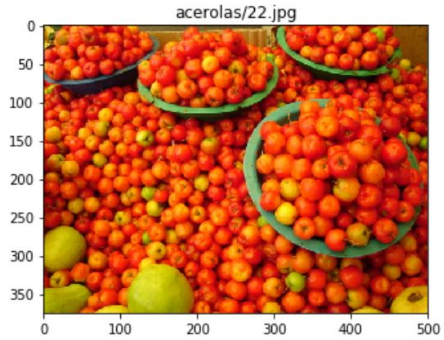
Color Histogram based approach - Theory

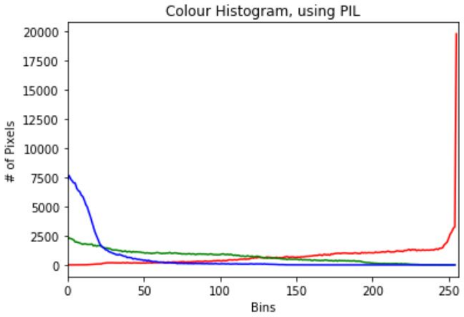
Color Histograms were calculated according to the provided script.

Idea: Calculate the frequency of use for each color in each image.

Improvement: Combining the three color channels and calculating frequencies of all possible combinations of the three color values for each pixel.

(i.e. frequency of red having value A, green having value B and blue having value C in each pixel, where A,B,C ∈ [0,256], for all possible combinations of values for A, B and C)





Example images taken from provided script

Color Histogram based approach - Results

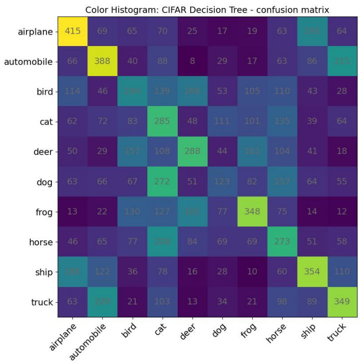
Dataset	Classifier	**Accuracy**	**Recall**	**Time (s)**
CIFAR 10	SVM	0.52	0.16	8199
	K-Nearest Neighbors	0.40	0.39	577
	Decision Tree	0.30	0.30	4
	Multi Layer Perceptron	0.35	0.36	177
MNIST	SVM	0.56	0.57	3053
	K-Nearest Neighbor	0.54	0.51	347
	Decision Tree	0.46	0.46	6
	Multi Layer Perceptron	0.54	0.56	306

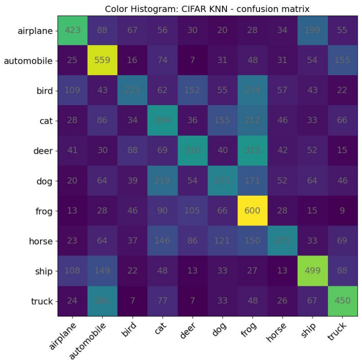
For CIFAR-10 the predictions mediocre, however a little better than our SIFT results for this data set

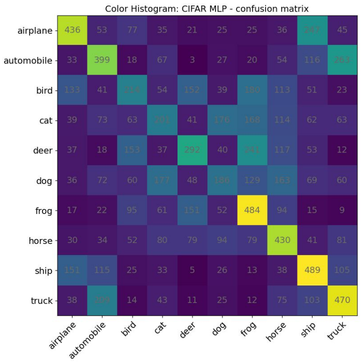
For FashionMNIST the results were better despite only having one color channel, SIFT was better however.

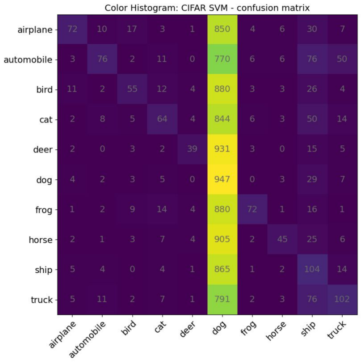
➜ Still surprisingly good results for such a simple and quick to compute descriptor

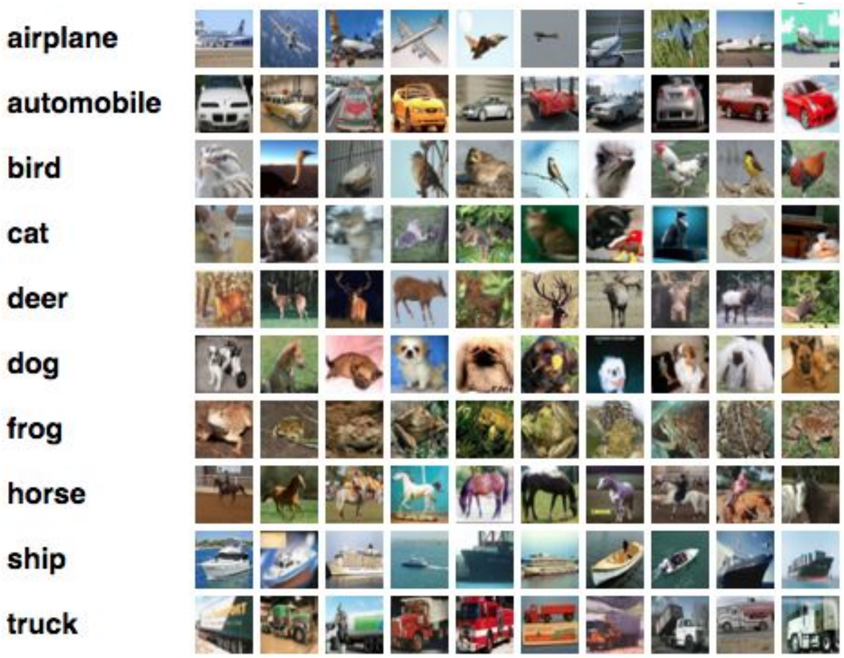
Color Histogram based approach - Confusion Matrix - CIFAR-10



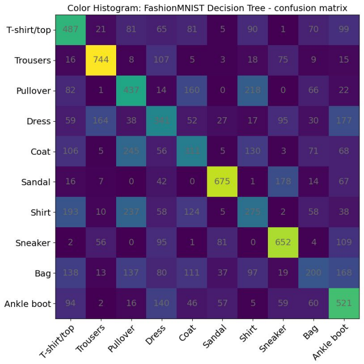


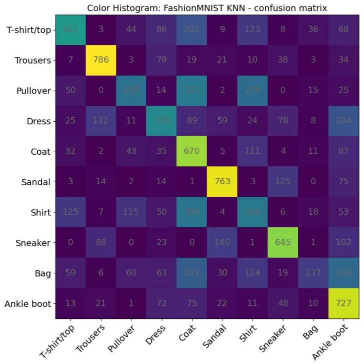


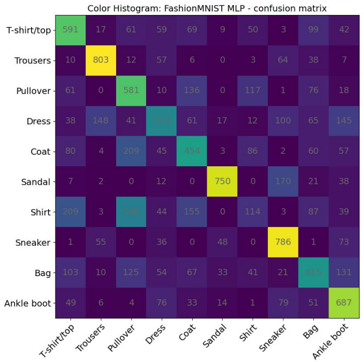


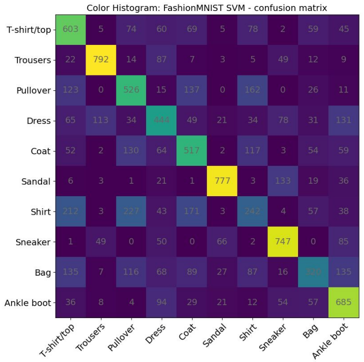


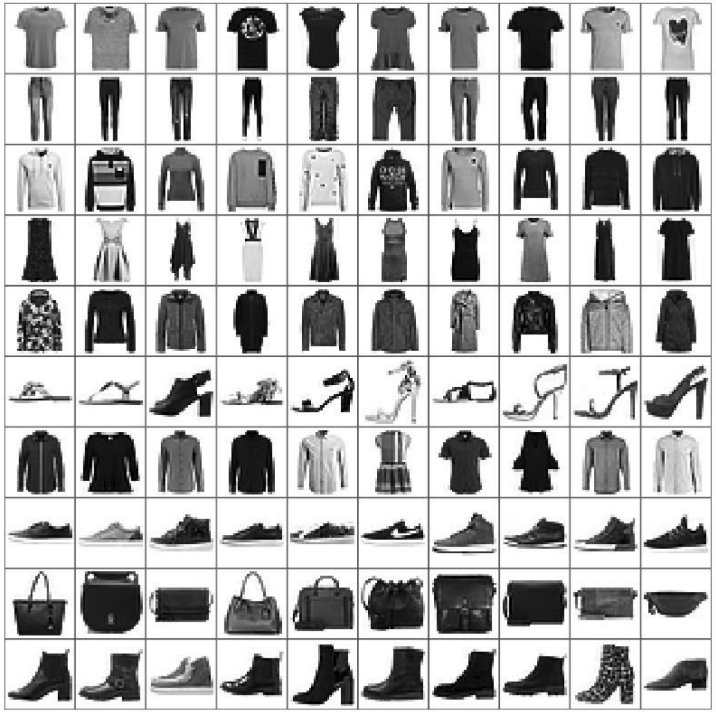
Color Histogram based approach - Confusion Matrix - FashionMNIST











Convolutional Neural Networks

Architectures

Simple:

- Convolution. Input = 28x28x1.
- SubSampling (Max Pooling).
- Dropout.
- Fully Connected #1. Output = 128.
- Output 10.

***Lenet5*:**

- Convolution #1. Input = 28x28x1.
- SubSampling (Max Pooling) #1.
- Convolution #2.
- SubSampling (Max Pooling) #2.
- Fully Connected #1. Output = 256.
- Fully Connected #2. Output = 84.
- Output 10.

Plus:

- Convolution #1. Input = 28x28x1.
- SubSampling (Max Pooling) #1.
- Batch Normalisation #1.
- Convolution #2.
- SubSampling (Max Pooling) #2.
- Batch Normalisation #2.
- Dropout #1.
- Fully Connected #1. Output = 128.
- Dropout #2.
- Fully Connected #2. Output = 50.
- Output 10.

Plusplus:

- Convolution #1. Input = 28x28x1.
- Convolution #2.
- SubSampling (Max Pooling) #1.
- Batch Normalisation #1.
- Dropout #1.
- Convolution #3.
- Convolution #4.
- SubSampling (Max Pooling) #2.
- Batch Normalisation #2.
- Dropout #2.
- Convolution #5.
- Convolution #6.
- SubSampling (Max Pooling) #3.
- Batch Normalisation #3.
- Dropout #3.
- Fully Connected #1. Output = 512.
- Dropout #2.
- Output 10.

FashionMNIST - Learning

Architecture	Data Augmentation	Accuracy	Loss	Processing Time
MLP	No	88.46%	0.333	74.88 s
CNN - Simple	No	91.82%	0.266	439.44 s (~7 min)
CNN - Simple	Yes	81.72%	0.489	646.90 s (~11 min)
CNN - *Lenet5*	No	91.69%	0.319	1046.09 s (~17 min)
CNN - Plus	No	91.58%	0.237	730.83 s (~12 min)
CNN - Plus Plus	No	92.70%	0.205	2573.60 s (~42 min)

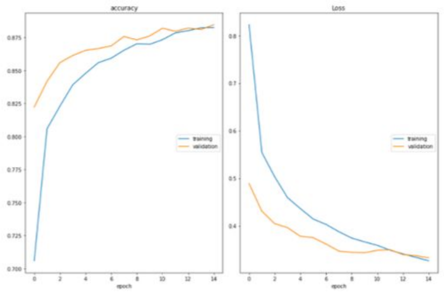


Figure 5.1: Evolution of Accuracy and Loss in FashionMNIST using MLPs

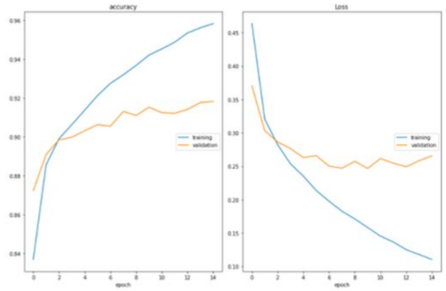


Figure 5.2: Evolution of Accuracy and Loss in FashionMNIST using CNNs (architecture - simple)

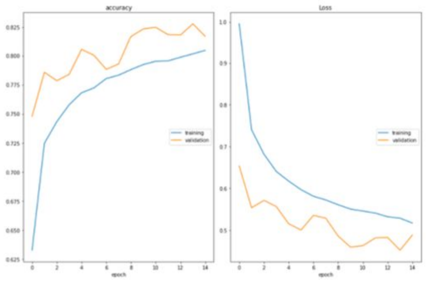


Figure 5.3: Evolution of Accuracy and Loss in FashionMNIST using CNNs and applying Data Augmentation (architecture - simple)

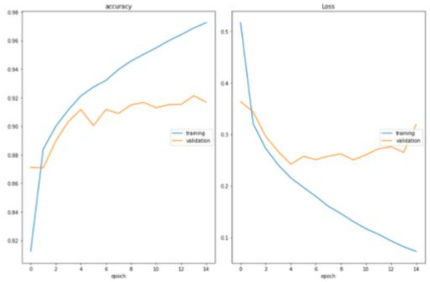


Figure 5.4: Evolution of Accuracy and Loss in FashionMNIST using CNNs (architecture - *Lenet5*)

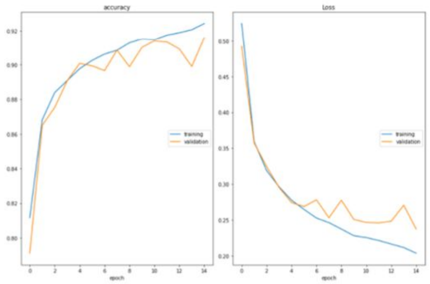


Figure 5.5: Evolution of Accuracy and Loss in FashionMNIST using CNNs (architecture - plus)

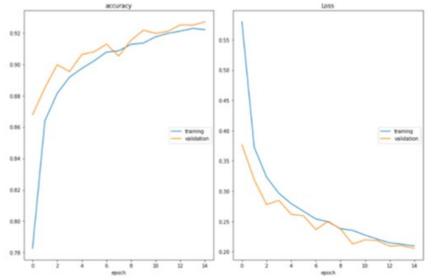


Figure 5.6: Evolution of Accuracy and Loss in FashionMNIST using CNNs (architecture - plus plus)

CIFAR-10 - Learning

Architecture	Data Augmentation	Accuracy	Loss	Processing Time
MLP (greyscale)	No	37.15%	1.786	88.15 s
CNN - Simple	No	70.07%	0.895	2094.20 s (∼35 min)
CNN - Simple	Yes	60.66%	1.097	2059.43 s (∼34 min)
CNN - *Lenet5*	No	67.80%	1.058	1083.34 s (∼18 min)
CNN - Plus	No	77.46%	0.746	3297.38 s (∼54 min)
CNN - Plus Plus	No	79.29%	0.613	4035.80 s (∼67 min)

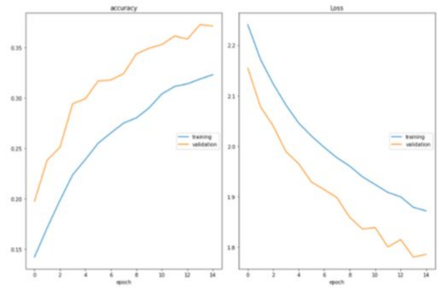


Figure 5.7: Evolution of Accuracy and Loss in CIFAR using MLPs

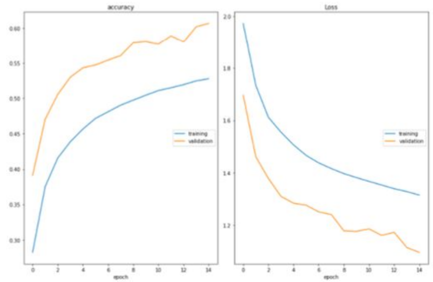


Figure 5.9: Evolution of Accuracy and Loss in CIFAR using CNNs and applying Data Augmentation (architecture - simple)

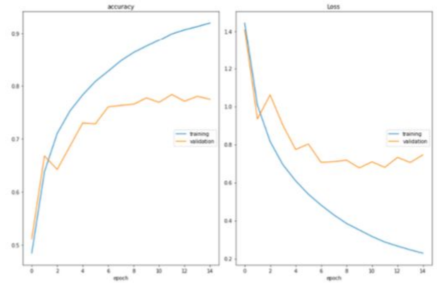


Figure 5.11: Evolution of Accuracy and Loss in CIFAR using CNNs (architecture - plus)

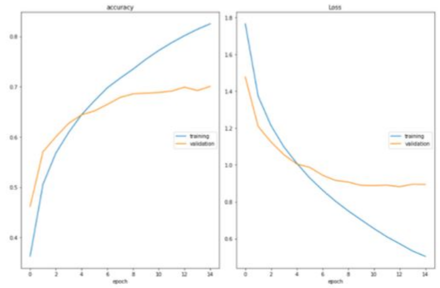


Figure 5.8: Evolution of Accuracy and Loss in CIFAR using CNNs (architecture - simple)

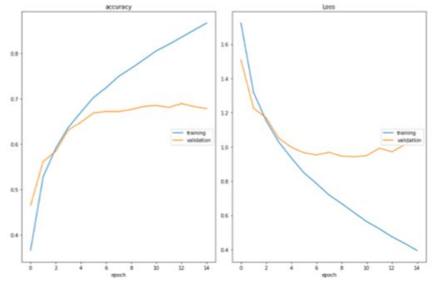


Figure 5.10: Evolution of Accuracy and Loss in CIFAR using CNNs (architecture - *Lenet5*)

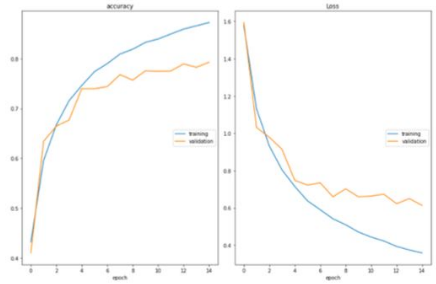
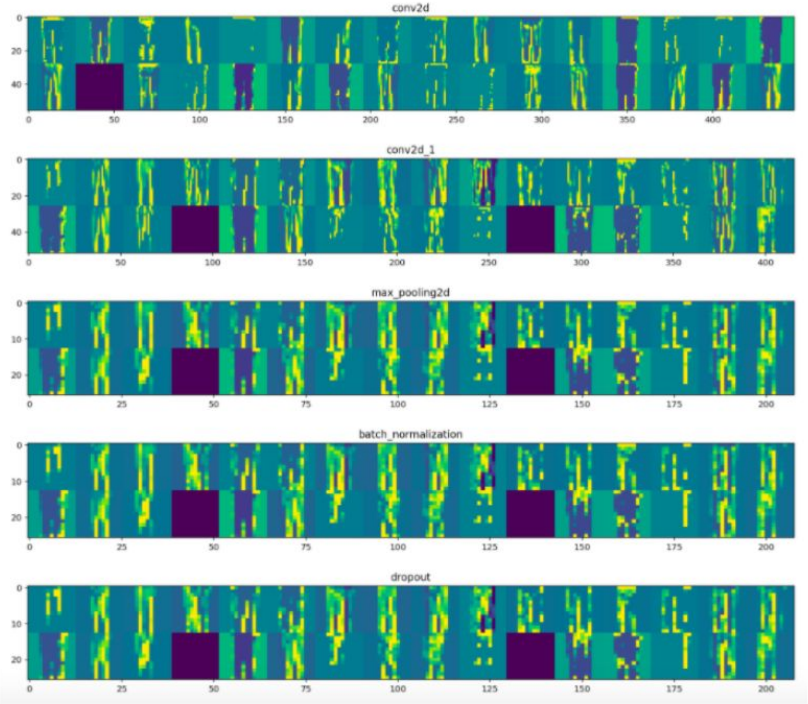


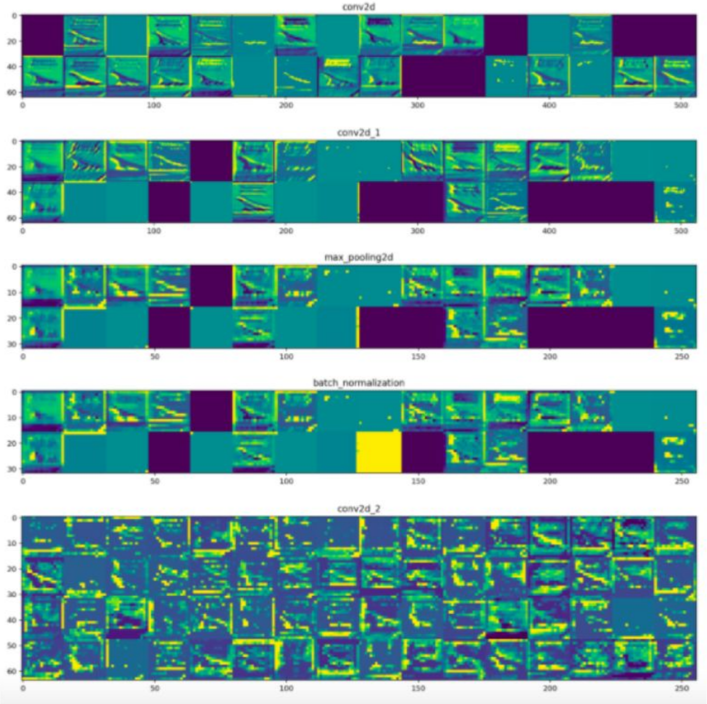
Figure 5.12: Evolution of Accuracy and Loss in CIFAR using CNNs (architecture - plus plus)

Featuremaps Visualization

FashionMNIST (Trousers)



CIFAR-10 (Frog)



Exercise 3 - Deep Learning

Machine Learning

2020/2021 | Technische Universität Wien

Q&A

Grupo 06
Diogo Braga (E12007525)
Enrico Coluccia (E12005483)
Matthias Kiss (01218325)