Batch Normalization Cifar10 Dataset

*Fadilla Zennifa

***Abstract*—Established the most optimum architecture in supervised-unsupervised system becomes challenging. There are several parameters such as the number of layers, batch size, the number of epochs need to be considered. This report proposed a method to established the best model for classifies Cifar 10 dataset. Proposed model used convolutional neural network (CNN) as classifier. After testing 21 models, I found that decided the number of batches is important for gaining the accuracy of model performance. Based on that condition, I proposed to used Normalization batch size. Proposed models got accuracy performance in training data set up to 95.42% in with Normalization batch.**

***Keywords*— CNN, Cifar 10, tensorflow**

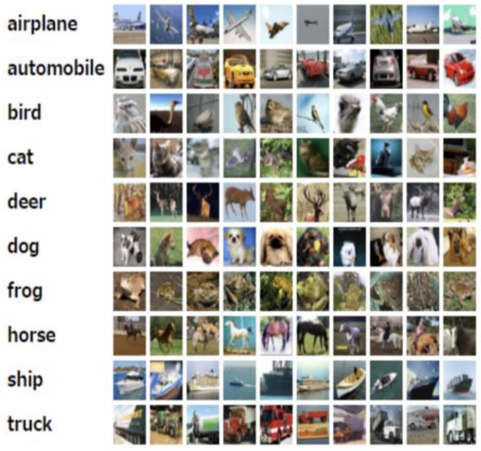
I. Introduction

This report is as a report for ridge-I hiring system. Cifar 10 dataset became dataset that has to be used in current report. To creating a model for supervised – unsupervised, CNN has been chosen as a classifier. The usage of CNN commonly has been done by using python. Considering the performance of Python IDLE, in this report Jupyter notebook with tensorflow library has been used. There are several trials and error when establishing the proposed method. Finally, this report mentioned to use naturalized batch size when establish the model for Cifar 10 dataset.

II. Methodology

A. Environment

I built this architecture by using Lenovo Yoga 710, Inter core i5, windows 10 and 64 bit Ram 8GB. Environment that has been used to establish the architecture is Anaconda, Jupyter notebook, tensorflow library and keras. Dataset that has been used in this report is Cifar10 dataset. There are 10 classes of Cifar 10 dataset. These classes can be seen in Fig1.



0 = airplan, 1= automobile, 2 =bird, 3= cat, 4= deer, 5= dog, 6 = frog, 7= horse, 8= ship, 9 = truck.

Fig 1 Cifar 10 Dataset[1]

This report chose to train 50% of all class in training data to make dataset become balance (can be seen in Fig 2.) rather than use different size of class in training dataset. This action has been chosen to reduce the overfitting conditions..

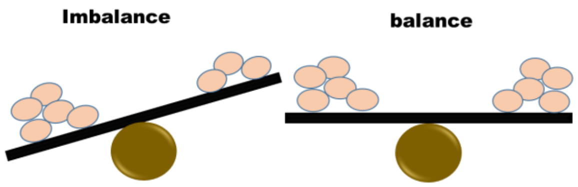


Fig 2 Balancing data

B. CNN (Convolutional Neural network)

A CNN is a feed-forward neural network that extracts image features using multiple convolutional layers and puts them into fully connected layers with a softmax function[2][3]. CNN has been chosen as classifier because in this report the data are images data.

III. Result

I applied 23 models to find the best models to be applied in Cifar 10 dataset (Can be seen on Appendix). The results can be seen as follow:

A. Layer decision

Layer is collection of 'nodes' operating together at a specific depth within a neural network (NN). Here, layer effects toward model's performance has been investigated. From Table 1, six number of layer shown the highest performance compare with other layers.

Table 1. Layer effects toward model's performance

Number of layer	Layer performance	
	Training	Testing
3	96.04%	61.88%
3+ max pooling	91.82%	69.24%
5+ max pooling	77.80%	75.49%
6+ max pooling	95.42%	81.96%

B. Batch size selections

This study tried to find the best model for Cifar 10 dataset. After applying 23 models, Table 2 shown normalized batch size has the highest performance compare with other conditions.

Table 2. Batch size investigation

Number of Epoch	Batch size	Batch performance	
		Training	Testing
25	10	68.44%	67.66%
10	32	84.59%	68.78%
100	64	87.00%	75.87%
40	502	67.90%	70.02%
100	512	77.80%	75.49%
10	1024	48.19%	52.69%
10	2048	43.69%	45.15%
20	Normalization	85.71%	81.59%
100	Normalization	95.42%	81.96%

C. *Overfitting and underfitting*

Training a models sometimes can be tricky. This situation is happened because sometimes high accuracy of training dataset will not show good performance in testing data. Sometimes the condition is in opposite, testing performance will be higher than training data set, where this condition is called as underfitting. From several models that has been trained in this report, there are some models shown overfitting (Fig 3).

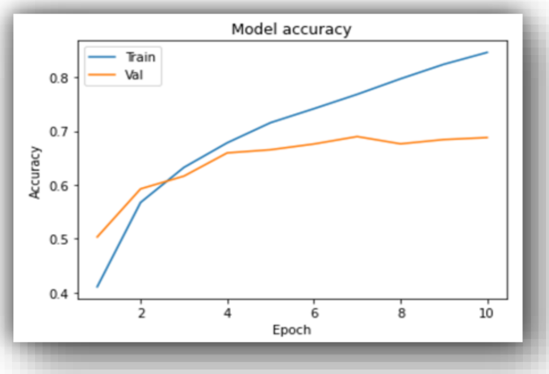
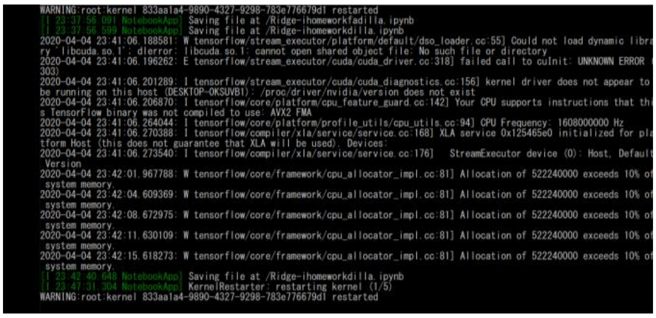


Fig 3 Cifar 10 Dataset[1]

Models with overfitting or underfitting become this report consideration. Based on that condition, when the training condition and validation condition has value more than 80%, that model will be considered as future proposed model.

IV. Discussion

When applying deep learning, usually batch 32 become the most recommended batch to be used. But in this study, rather than recommend to use batch size 32, normalized batch size became a recommendation. The reason to choose the same size of classes in this report because author consider with the data balancing. Using different size of data could cause overfitting problems [3],[4]. The author also would like to report that by using tf.concat function to combine the classes in tensorflow libraries makes the performance slow.



```
WARNING:root:kernel 833aa1a4-9890-4327-9298-783e776679d1 restarted
[I 23:37:56.091 NotebookApp] Saving file at /Ridge-ihomeworkfadilla.ipynb
[I 23:37:56.599 NotebookApp] Saving file at /Ridge-ihomeworkdilla.ipynb
2020-04-04 23:41:06.188581: W tensorflow/stream_executor/platform/default/dso_loader.cc:55] Could not load dynamic libra
ry 'libcuda.so.1'; dlerror: libcuda.so.1: cannot open shared object file: No such file or directory
2020-04-04 23:41:06.196262: E tensorflow/stream_executor/cuda/cuda_driver.cc:318] failed call to cuInit: UNKNOWN ERROR (
303)
2020-04-04 23:41:06.201289: I tensorflow/stream_executor/cuda/cuda_diagnostics.cc:156] kernel driver does not appear to
be running on this host (DESKTOP-OKSUVB1): /proc/driver/nvidia/version does not exist
2020-04-04 23:41:06.206870: I tensorflow/core/platform/cpu_feature_guard.cc:142] Your CPU supports instructions that thi
s TensorFlow binary was not compiled to use: AVX2 FMA
2020-04-04 23:41:06.264044: I tensorflow/core/platform/profile_utils/cpu_utils.cc:94] CPU Frequency: 1608000000 Hz
2020-04-04 23:41:06.270388: I tensorflow/compiler/xla/service/service.cc:168] XLA service 0x125465e0 initialized for pla
tform Host (this does not guarantee that XLA will be used). Devices:
2020-04-04 23:41:06.273540: I tensorflow/compiler/xla/service/service.cc:176]   StreamExecutor device (0): Host, Default
 Version
2020-04-04 23:42:01.967788: W tensorflow/core/framework/cpu_allocator_impl.cc:81] Allocation of 522240000 exceeds 10% of
 system memory.
2020-04-04 23:42:04.609369: W tensorflow/core/framework/cpu_allocator_impl.cc:81] Allocation of 522240000 exceeds 10% of
 system memory.
2020-04-04 23:42:08.672975: W tensorflow/core/framework/cpu_allocator_impl.cc:81] Allocation of 522240000 exceeds 10% of
 system memory.
2020-04-04 23:42:11.630109: W tensorflow/core/framework/cpu_allocator_impl.cc:81] Allocation of 522240000 exceeds 10% of
 system memory.
2020-04-04 23:42:15.618273: W tensorflow/core/framework/cpu_allocator_impl.cc:81] Allocation of 522240000 exceeds 10% of
 system memory.
[I 23:42:40.648 NotebookApp] Saving file at /Ridge-ihomeworkdilla.ipynb
[I 23:47:31.304 NotebookApp] KernelRestarter: restarting kernel (1/5)
WARNING:root:kernel 833aa1a4-9890-4327-9298-783e776679d1 restarted
```

Fig 4 Crash when used tf.concat library

This report still limited with the 50% training data. It is because the specification of personal computer that has been used in this report was not a super computer. Where usually to perform deep learning, 16 GB RAM become the minimum requirement. But this study only used 8 GB RAM. The author really hope can investigate more about artificial intelligence by joining Ridge-I company.

V. Conclussion

After several trials and error, I found that the highest performance accuracy > 80 % and validation >75%.

References

[1.] A. Krizhevsky, V. Nair, G. Hinton, CIFAR-10 Dataset, [online] Available: https://www.cs.toronto.edu/kriz/cifar.htm

[2.] The Y. Weng, T. Zhou, L. Liu and C. Xia, "Automatic Convolutional Neural Architecture Search for Image Classification Under Different Scenes," in *IEEE Access*, vol. 7, pp. 38495-38506, 2019.

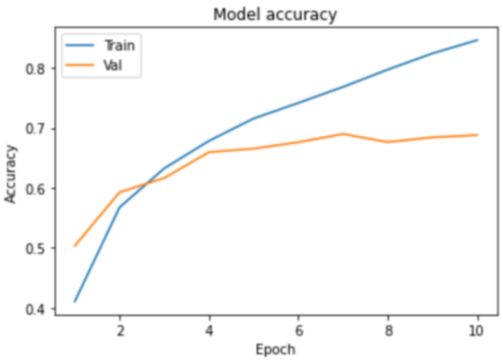
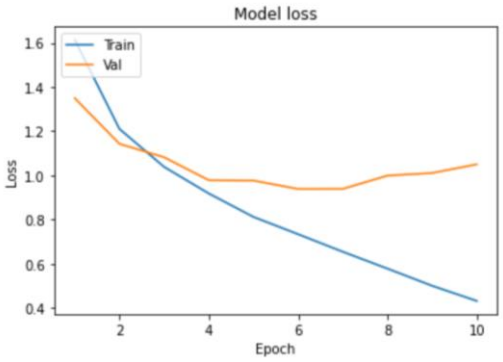
[3.] C. Yu, X. He, H. Ma, X. Qi, J. Lu and Y. Zhao, "S-DenseNet: A DenseNet Compression Model Based on Convolution Grouping Strategy Using Skyline Method," in IEEE Access, vol. 7, pp. 183604-183613, 2019.

[4.] F. Zennifa,; Ageno, S.; Hatano, S.; Iramina, K. Hybrid System for Engagement Recognition During Cognitive Tasks Using a CFS + KNN Algorithm. *Sensors* **2018**, *18*, 3691

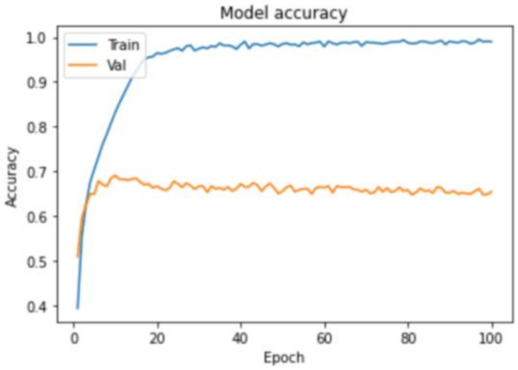
[5.] F. Zennifa and K. Iramina, "Quantitative Formula of Blink Rates-Pupillometry for Attention Level Detection in Supervised Machine Learning," in *IEEE Access*, vol. 7, pp. 96263-96271, 2019.

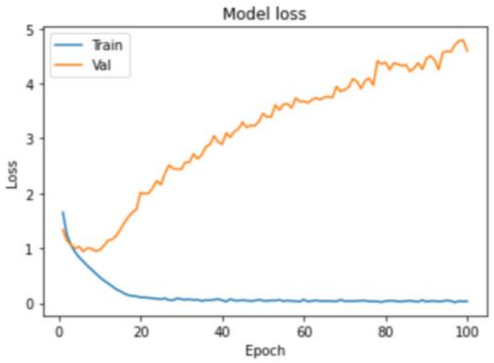
Appendix

No	Model fitting	Training Validation accuracy	Training Validation Loss

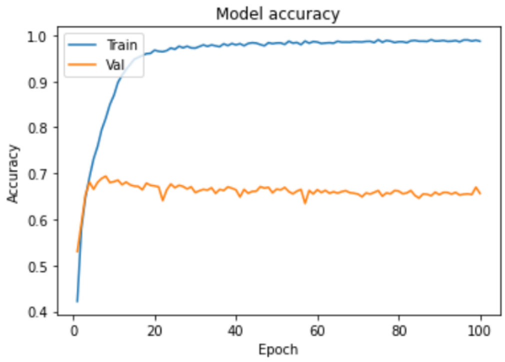
1.	Batch size = 32
	Epoch =10
	Verbose=1
			

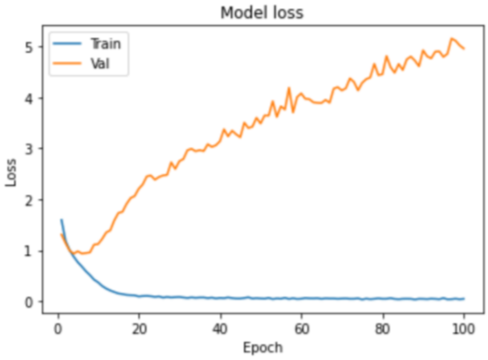
2. Batch size = 64
Epoch =100
Verbose=1



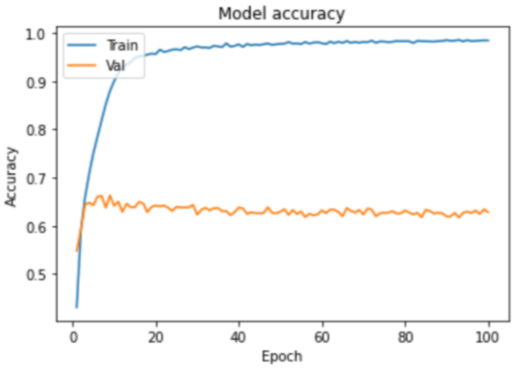


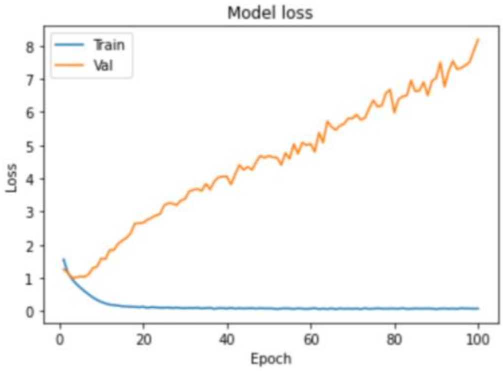
3. Batch size = 32
Epoch =100
Verbose=1



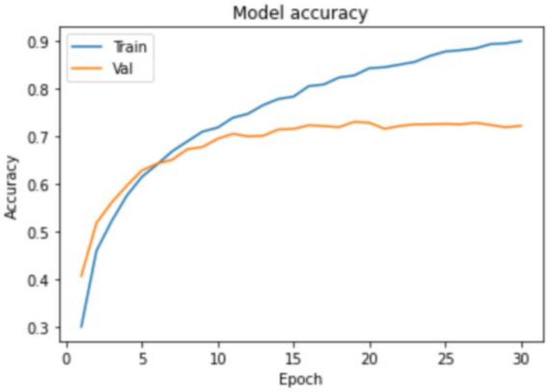


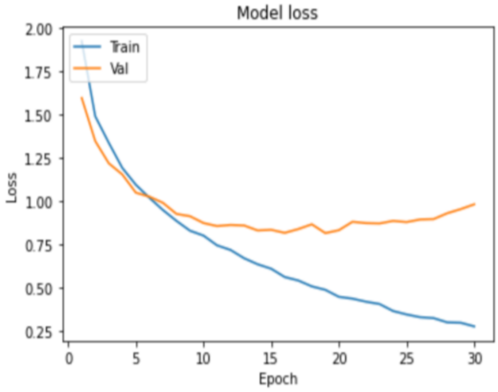
4. Batch size = 10
Epoch =100
Verbose=1



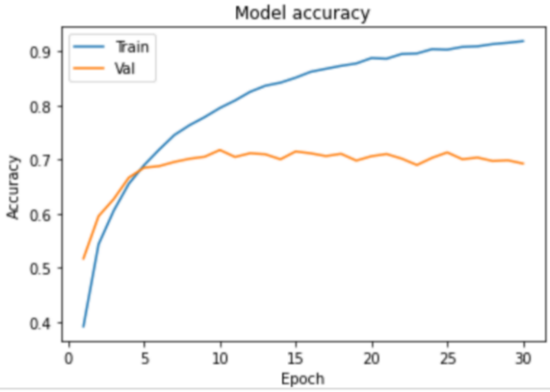


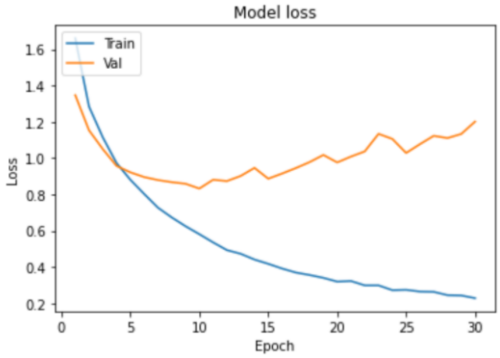
5. Batch size = 256
Epoch = 30
Verbose=1



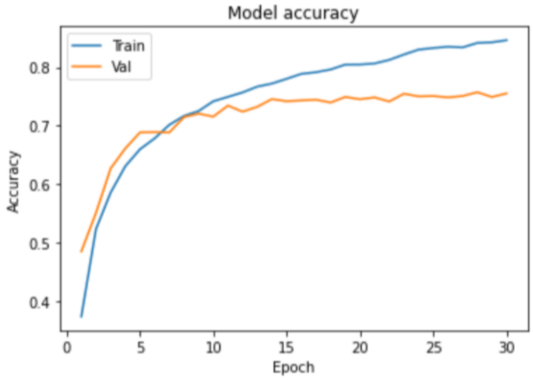


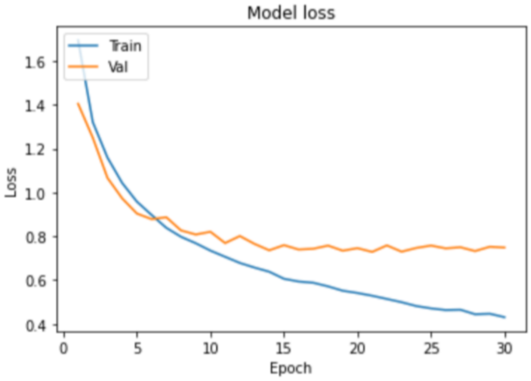
6. Batch size =32
Epoch = 30
Verbose=1



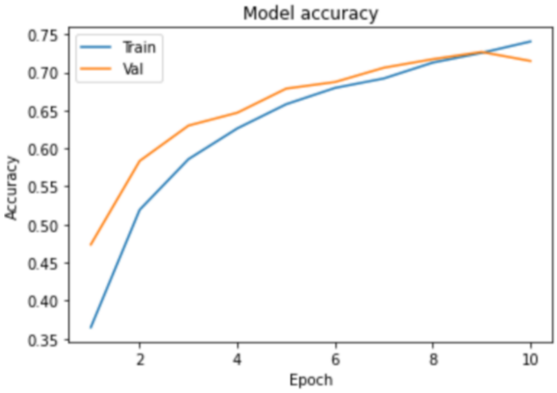


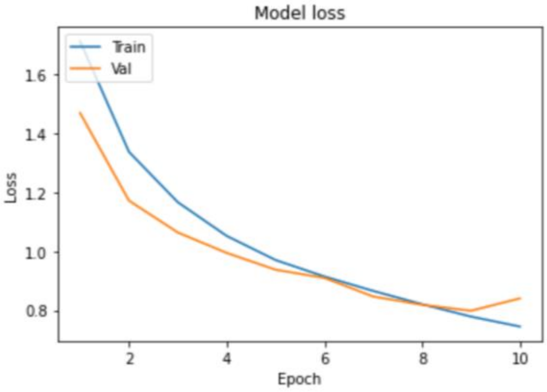
7 Batch size =32
Epoch = 30
Verbose=1
Filtering
96,96,192



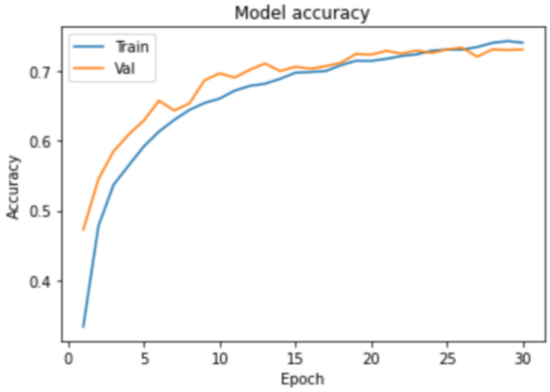


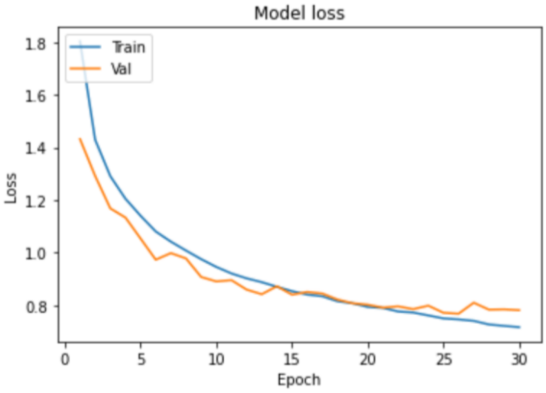
8. Batch size =32
Epoch = 10
Verbose=1
Filtering
96,96,192



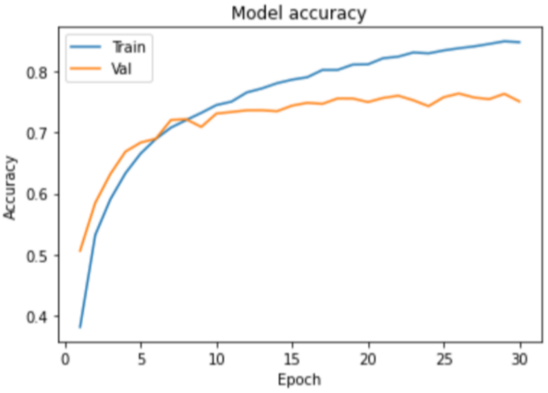


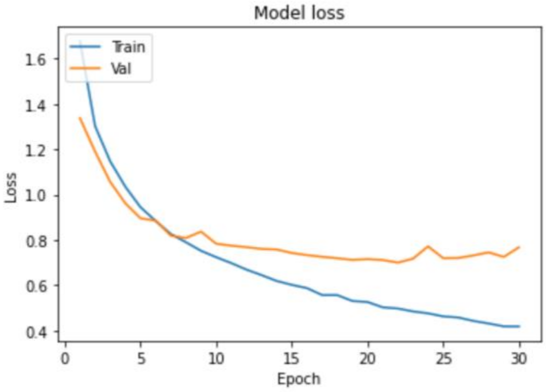
9. Batch size =32
Epoch = 30
Verbose=1
Filtering
64,32,64



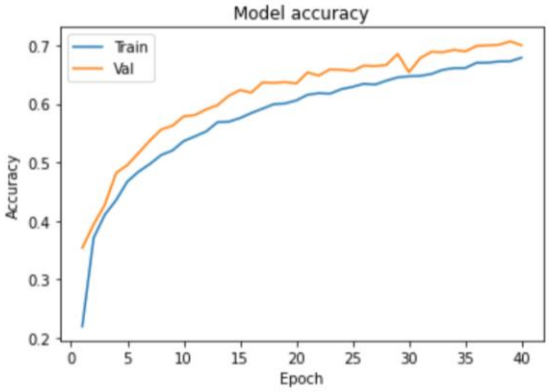


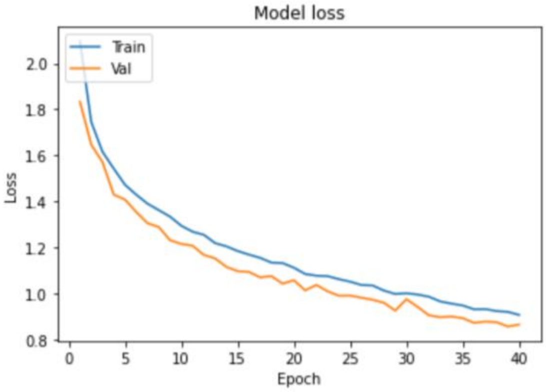
10. Batch size =32
Epoch = 30
Verbose=1
Filtering 64,92,192



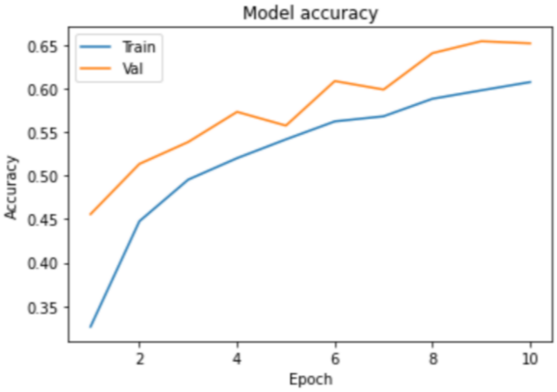


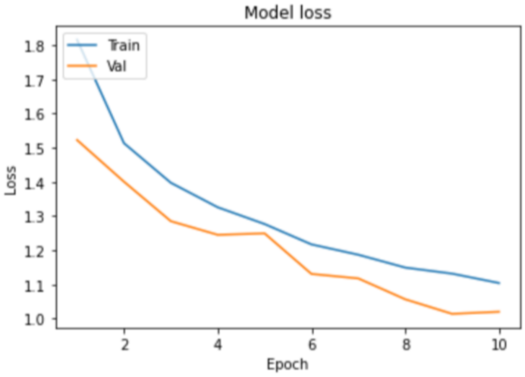
11. Batch size =502
Epoch = 40
Verbose=1
Filtering 64,92,192



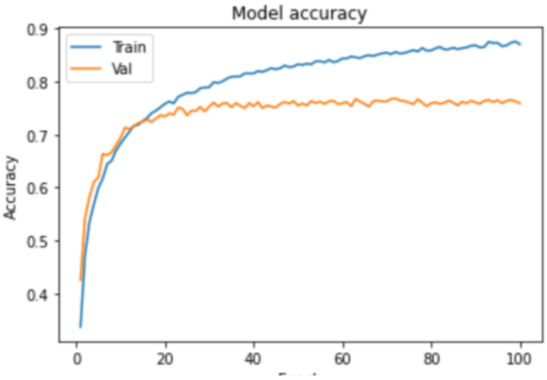


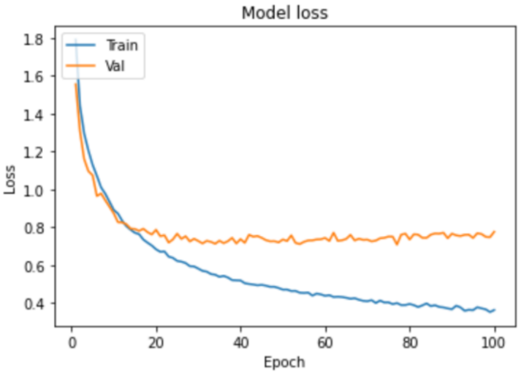
12 Epoch = 10
Verbose=1
Filtering 64,32,64,
dense unit 128

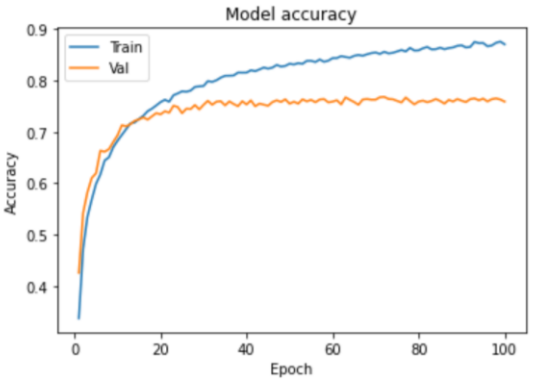
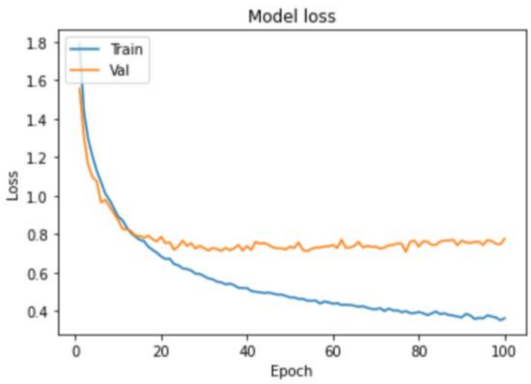


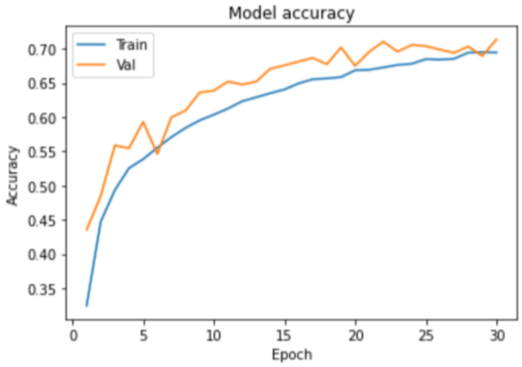
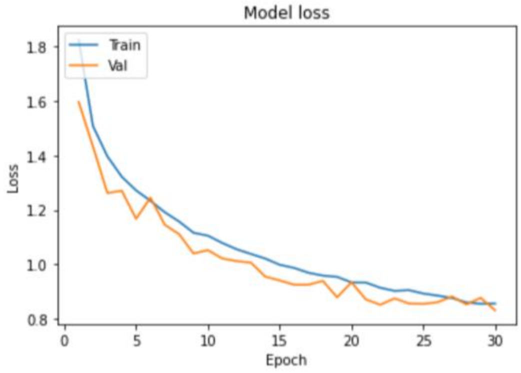


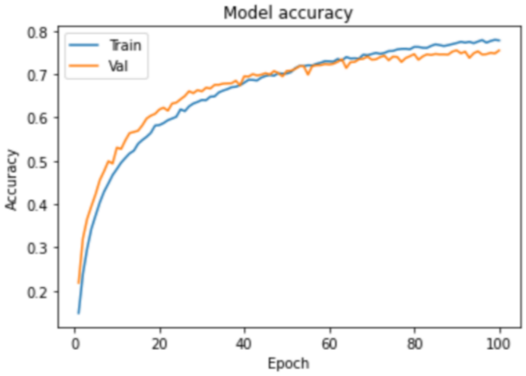
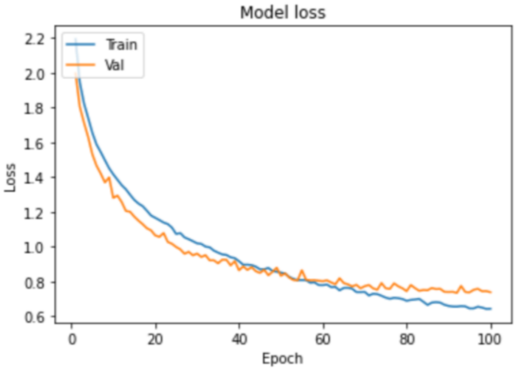
13 Batch size =64
Epoch = 100
Verbose=1
Filtering 64, 64, 64
dense 128

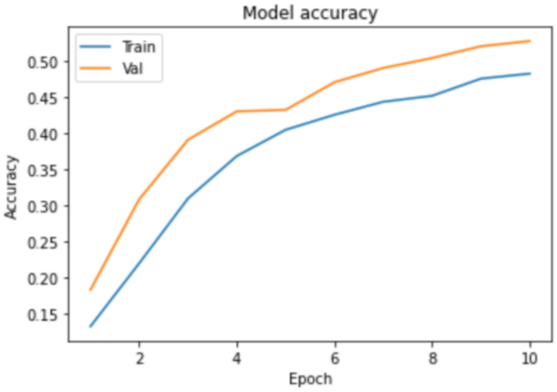
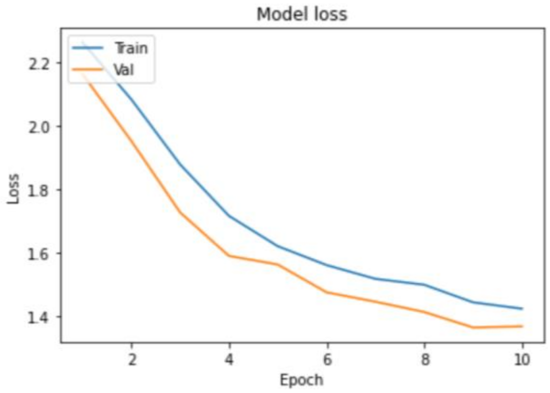




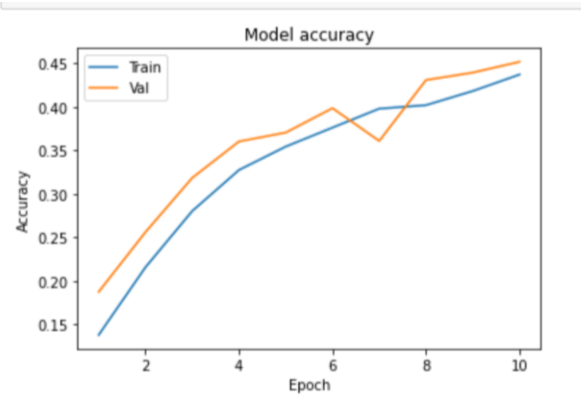
14	Batch size =64
	Epoch = 100
	Verbose=1
	Filtering 64, 64, 64 dense 128
			

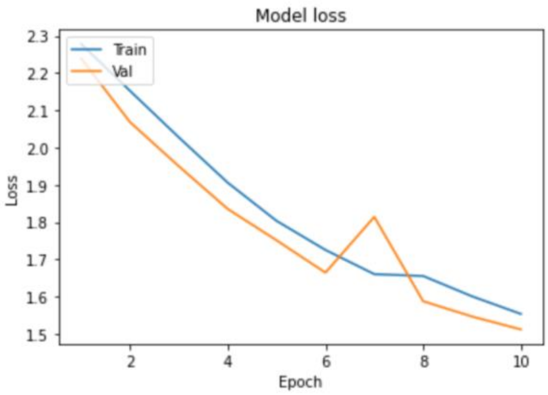
15	Epoch = 30
	Verbose=1
	Filtering 64, 32, 64 dense 128
			

16	Batch size =512
	Epoch = 100
	Verbose=1
	Filtering 64, 32, 64, 64, 64 dense 32
			

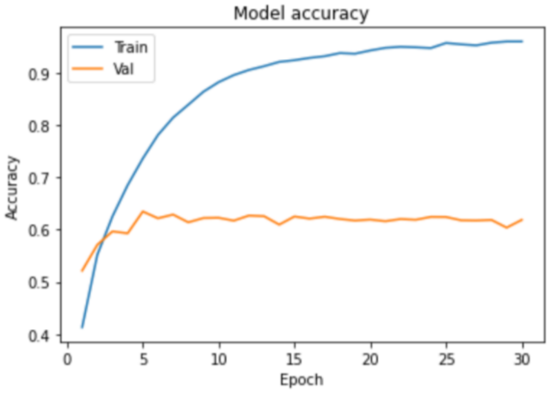
17	Batch size =1024
	Epoch = 10
	Verbose=1
	Filtering 64, 32, 64, dense 32
			

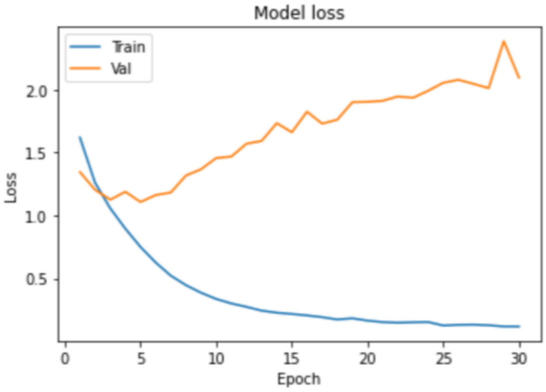
18 Batch size =2048
Epoch = 10
Verbose=1
Filtering 64, 32, 64, dense 32



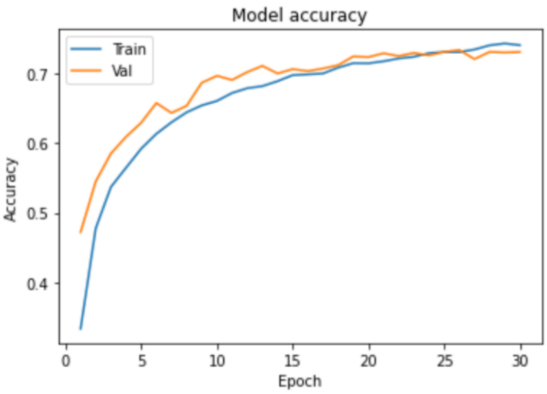


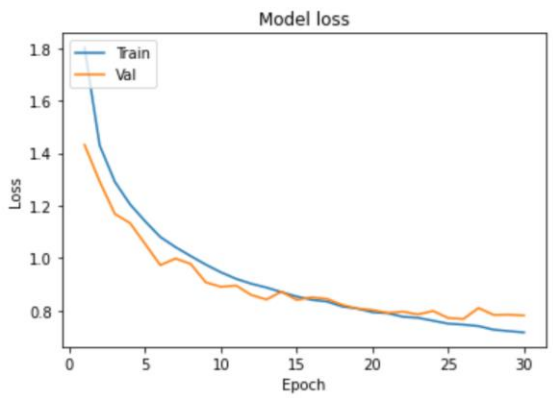
19 Batch size =32
Epoch = 30
Verbose=1
Filtering 64, 32, 64, dense 64, no max pooling in this case



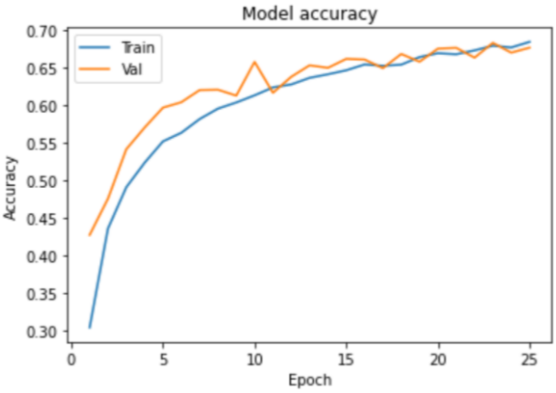


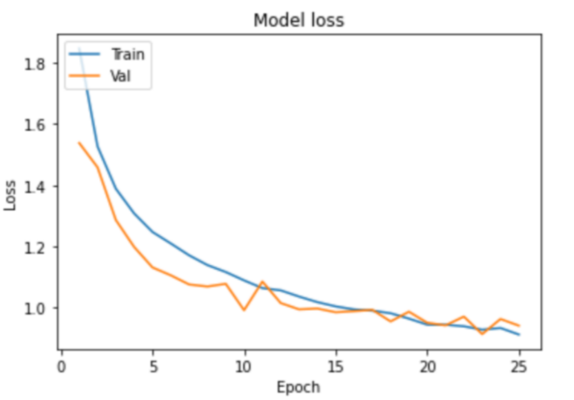
20 Batch size =32
Epoch = 30
Verbose=1
Filtering 64, 32, 64, dense 64

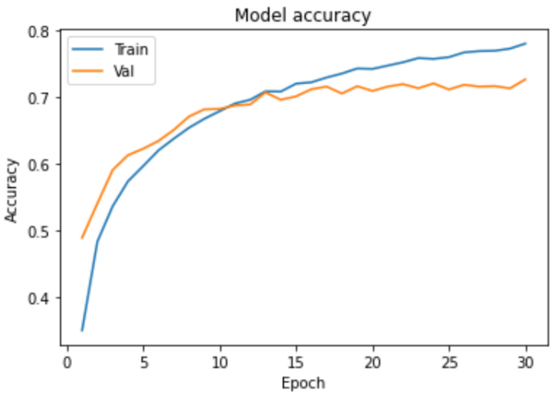
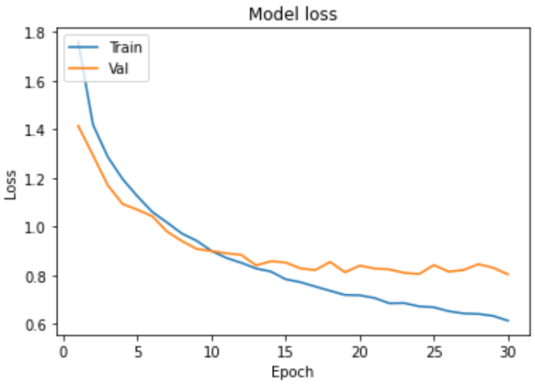


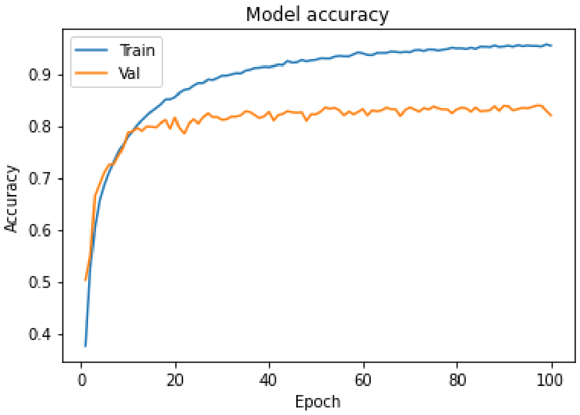
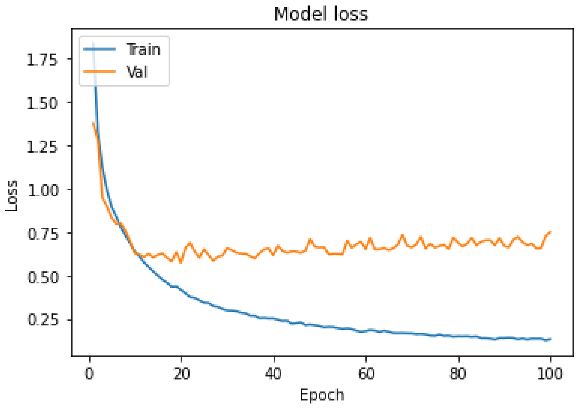


21 Batch size =10
Epoch = 25
Verbose=1
Filtering 64, 32, 64,,64,64 dense 32





22	Batch size =32
	Epoch =30
	Verbose=1
	Filtering 64, 32, 64, dense 64
			

23	Batch normalization
	Epoch = 100
	Verbose=1, 6 layers.
		 Model accuracy
		Train
		Val
		Accuracy
		0.4
		0.5
		0.6
		0.7
		0.8
		0.9
		Epoch
		0
		20
		40
		60
		80
		100
			 Model loss
			Train
			Val
			Loss
			0.25
			0.50
			0.75
			1.00
			1.25
			1.50
			1.75
			Epoch
			0
			20
			40
			60
			80
			100