STL10_PROJECT summary

- Images classification of STL-10 dataset's selected classes with one and multi GPU processing

In [1]:
```
import my_functions
import pandas as pd
```

1. Dataset

STL-10 dataset, containing of 5000 train labelled images and 8000 test labelled images.

2. Project folder structure

In [2]:
```
ls /home/artur/STL10/STL10_PROJECT
```

```
0_Project_summary.ipynb             Module_19_model_7.ipynb
appendix_env.ipynb                  Module_1_model_0.ipynb
inst_history.txt                    Module_2_Data_augmentation.ipynb
Models/                             Module_3_model_0_augm.ipynb
Module_0_Data_preprocessing.ipynb   Module_4_model_1.ipynb
Module_10_Vis_Filters.ipynb         Module_5_model_2.ipynb
Module_11_vgg16_filters.ipynb       Module_6_model_3.ipynb
Module_12_model_5.ipynb             Module_7_model_4.ipynb
Module_13_model_5.ipynb             Module_8_Vis_Activations.ipynb
Module_14_model_5.ipynb             Module_9_Vis_Weights.ipynb
Module_15_model_6.ipynb             my_functions.py
Module_16_model_7.ipynb             __pycache__/
Module_17_model_7.ipynb             read_me.txt
Module_18_model_7.ipynb             results_summary.txt
Module_19a_model7.ipynb             Used_data/
```

In [3]:
```
ls /home/artur/STL10/STL10_PROJECT/Used_data
```

```
TEST_IMAGES.npy  TRAIN_IMAGES.npy  TRAIN_X_EXTD.npy  VALID_IMAGES.npy
TEST_LABELS.npy  TRAIN_LABELS.npy  TRAIN_Y_EXTD.npy  VALID_LABELS.npy
```

In [4]:
```
ls /home/artur/STL10/STL10_PROJECT/Models
```

```
classifier_0.h5                 model_2_8-100-25-RMSp-trd.h5
classifier_1.h5                 model_2_8-30-20-adlt-trd.h5
classifier_2.h5                 model_2_8-30-40-sgd-trd.h5
classifier_3.h5                 model_2_8-30-9-adlt-trd.h5
classifier_4.h5                 model_2.h5
classifier_5.h5                 model_3_8-100-100-adlt-trd.h5
conv_base_0.h5                  model_3_8-100-100-sgd-trd.h5
conv_base_1.h5                  model_3_8-100-20-adlt-trd.h5
conv_base_2.h5                  model_3_8-100-30-RMSp-trd.h5
conv_base_3.h5                  model_3.h5
conv_base_4.h5                  model_4_8-100-100-sgd-trd.h5
model_0_0-100-100-RMSp-trd.h5   model_4_8-100-20-adlt-trd.h5
model_0_0-100-100-sgd-trd.h5    model_4_8-100-30-RMSp-trd.h5
model_0_0-100-120-rmsp-trd.h5   model_4_8-100-40-RMSp-trd.h5
model_0_0-100-150-adlt-trd.h5   model_4.h5
model_0_0-100-150-sgd-trd.h5    model_5_8-100-25-RMSp-trd.h5
model_0_0-15-15-rmsp-trd.h5     model_5_8-100-50-sgd-trd.h5
model_0_0-15-25-RMSp-trd.h5     model_5_8-100-70-sgd-trd.h5
model_0_0-15-40-sgd-trd.h5      model_6_8-100-100-sgd-trd.h5
model_0_0-30-100-sgd-trd.h5     model_6_8-100-17-adlt-trd.h5
model_0_8-100-100-sgd-trd.h5    model_6_8-100-25-RMSp-trd.h5
model_0_8-100-150-sgd-trd.h5    model_6_8-100-30-RMSp-trd.h5
model_0_8-100-21-RMSp-trd.h5    model_6_8-100-32-RMSp-trd.h5
model_0_8-100-30-adlt-trd.h5    model_6_8-100-40-adlt-trd.h5
model_0_8-100-30-RMSp-trd.h5    model_6_8-100-50-sgd-trd.h5
model_0_8-30-100-sgd-trd.h5     model_6.h5
model_0_8-30-30-adlt-trd.h5     model_7_8-100-100-0vgg16-trd.h5
model_0_8-30-32-sgd-trd.h5      model_7_8-100-12-0vgg16-trd.h5
model_0_8-30-8-adlt-trd.h5      model_7_8-100-12-1vgg16R-trd.h5
model_0_8-30-8-RMSp-trd.h5      model_7_8-100-20-1vgg16s-trd.h5
model_0.h5                      model_7_8-100-20-2vgg16s-trd.h5
model_1_8-100-100-sgd-trd.h5    model_7_8-100-20-3vgg16s-trd.h5
model_1_8-100-30-RMSp-trd.h5    model_7_8-100-25-0vgg16-trd.h5
model_1_8-30-32-sgd-trd.h5      model_7_8-100-30-0vgg16-trd.h5
model_1_8-30-40-sgd-trd.h5      model_7_8-100-5-1vgg16s-trd.h5
model_1_8-30-9-adlt-trd.h5      model_7_8-100-5-2vgg16s-trd.h5
model_1.h5                      model_7_8-100-5-3vgg16s-trd_a.h5
model_2_8-100-100-sgd-trd.h5    model_7_8-100-5-3vgg16s-trd.h5
```

3. Activities performed as parts of the project

- Hardware and software environment preparation

It includes GPU cards installation in serwer HP DL-585, nVidia CUDA and cuDNN packages installation and configuration, installation of all needed python libraries including TensorFlow and Keras.

Notebook: **appendix_env.ipynb**

- Data preprocessing

An original data is a STL-10 dataset, containing of 5000 train labelled images and 8000 test labelled images. Although this dataset is described on https://cs.stanford.edu/~acoates/stl10/ (https://cs.stanford.edu/~acoates/stl10/) in this project more general case has been assumed where the

data structure in unknown and requires of detailed recognition. It has been done in 'Module_0' notebook, where the original data were finally transformed into two classes of images - planes and birds only - in totally 2600 images set, splitted up into 1600 train, 500 validation and 500 test ones.

Notebook: **Module_0_Data_preprocessing.ipynb**

- Basic model configuration and its training

Although there are only two classes this training and the rest ones in this project has been proceed not as binary classification mode but as multiclass classification, as more general approach. Training with one GPU.

Notebook: **Module_1_model_0.ipynb**

- The training set augmentation from 1600 into 8000

Due to the small quantity of training data the 'off-line' (into a file on hdd) augmentation has been done.

Data after augmentation:

In [5]:
```
my_functions.code_block_0()
```

Out[5]:

	set name	shape	dtype	data sight
0	TRAIN_X	(8000, 96, 96, 3)	uint8	[87, 73, 37]
1	TRAIN_Y	(8000,)	uint8	2
2	VALID_X	(500, 96, 96, 3)	uint8	[204, 205, 210]
3	VALID_Y	(500,)	uint8	1
4	TEST_X	(500, 96, 96, 3)	uint8	[132, 167, 203]
5	TEST_Y	(500,)	uint8	1

Notebook: **Module_2_Data_augmentation.ipynb**

- Models' training

Different own models configurations' check. Training with one GPU.

Notebooks:

Module_3_model_0_augm.ipynb

Module_4_model_1.ipynb

Module_5_model_2.ipynb

Module_6_model_3.ipynb

Module_7_model_4.ipynb

- Visualization of learning effects

Layers' activation visualization ----- notebook -- **Module_8_Vis_Activations.ipynb**

Weights' visualization ---------- notebook -- **Module_9_Vis_Weights.ipynb**

Convolutional filters' visualization - notebook -- **Module_10_Vis_Filters.ipynb**

- Trained VGG16 network overview

Notebook:

Module_11_vgg16_filters.ipynb

- The use of convolutional base from vgg16 network to the training

Training of the all vgg16 layers from the beginning with own classifier. Training with multi GPU mode.

Notebooks:

Module_12_model_5.ipynb

Module_13_model_5.ipynb

Module_14_model_5.ipynb

- Reduction of own model and training

The return to previous own model and decreasing of it depth. Training with one GPU.

Notebook:

Module_15_model_6.ipynb

- The use of trained weights from vgg16 network

Training of all frozen vgg16 convolutional layers with own classifier, trainings last one / last two / last three layers and the rest frozen + classifier. Training with multi GPU mode except of Module_19a.

Notebooks:

Module_16_model_7.ipynb

Module_17_model_7.ipynb

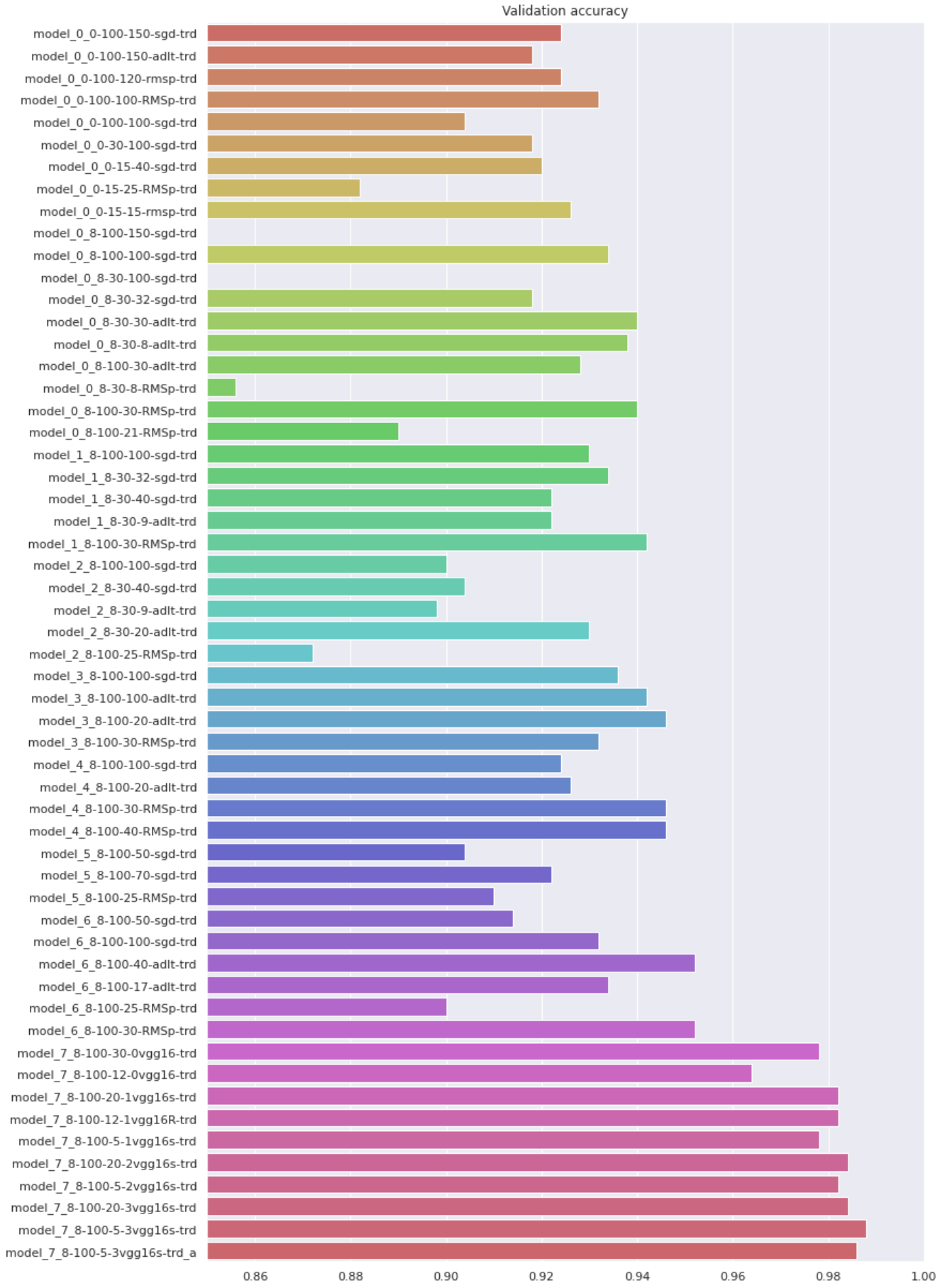
Module_18_model_7.ipynb

Module_19_model_7.ipynb

Module_19a_model7.ipynb

4. Results summary

In [6]:
```
my_functions.summary_barplot()
```



In [7]:
```
summary = pd.DataFrame(my_functions.results_preview())
summary.index = summary.index + 1
summary
```

Out[7]:

	model_name	model_description	notebook	dataset_size	batch_size	epochs	optimizer	acc
1	model_0_0-100-150-sgd-trd	32-64-128-256-512	Module_1	1600	100	150	sgd	0.924
2	model_0_0-100-150-adlt-trd	32-64-128-256-512	Module_1	1600	100	150	adadelta	0.918
3	model_0_0-100-120-rmsp-trd	32-64-128-256-512	Module_1	1600	100	120	rmsprop	0.924
4	model_0_0-100-100-RMSp-trd	32-64-128-256-512	Module_1	1600	100	100	RMSprop(lr=1e-4)	0.932
5	model_0_0-100-100-sgd-trd	32-64-128-256-512	Module_1	1600	100	100	sgd	0.904
6	model_0_0-30-100-sgd-trd	32-64-128-256-512	Module_1	1600	30	100	sgd	0.918
7	model_0_0-15-40-sgd-trd	32-64-128-256-512	Module_1	1600	15	40	sgd	0.920
8	model_0_0-15-25-RMSp-trd	32-64-128-256-512	Module_1	1600	15	25	RMSprop(lr=1e-4)	0.882
9	model_0_0-15-15-rmsp-trd	32-64-128-256-512	Module_1	1600	15	15	rmsprop	0.926
10	model_0_8-100-150-sgd-trd	32-64-128-256-512	Module_3	8000	100	150	sgd	0.522
11	model_0_8-100-100-sgd-trd	32-64-128-256-512	Module_3	8000	100	100	sgd	0.934
12	model_0_8-30-100-sgd-trd	32-64-128-256-512	Module_3	8000	30	100	sgd	0.522
13	model_0_8-30-32-sgd-trd	32-64-128-256-512	Module_3	8000	30	32	sgd	0.918
14	model_0_8-30-30-adlt-trd	32-64-128-256-512	Module_3	8000	30	30	adadelta	0.940
15	model_0_8-30-8-adlt-trd	32-64-128-256-512	Module_3	8000	30	8	adadelta	0.938
16	model_0_8-100-30-adlt-trd	32-64-128-256-512	Module_3	8000	100	30	adadelta	0.928
17	model_0_8-30-8-RMSp-trd	32-64-128-256-512	Module_3	8000	30	8	RMSprop(lr=1e-4)	0.856
18	model_0_8-100-30-RMSp-trd	32-64-128-256-512	Module_3	8000	100	30	RMSprop(lr=1e-4)	0.940
19	model_0_8-100-21-RMSp-trd	32-64-128-256-512	Module_3	8000	100	21	RMSprop(lr=1e-4)	0.890

	model_name	model_description	notebook	dataset_size	batch_size	epochs	optimizer	acc
20	model_1_8-100-100-sgd-trd	model_0 + Dropout	Module_4	8000	100	100	sgd	0.930
21	model_1_8-30-32-sgd-trd	model_0 + Dropout	Module_4	8000	30	32	sgd	0.934
22	model_1_8-30-40-sgd-trd	model_0 + Dropout	Module_4	8000	30	40	sgd	0.922
23	model_1_8-30-9-adlt-trd	model_0 + Dropout	Module_4	8000	30	9	adadelta	0.922
24	model_1_8-100-30-RMSp-trd	model_0 + Dropout	Module_4	8000	100	30	RMSprop(lr=1e-4)	0.942
25	model_2_8-100-100-sgd-trd	model_0 + reg. L2	Module_5	8000	100	100	sgd	0.900
26	model_2_8-30-40-sgd-trd	model_0 + reg. L2	Module_5	8000	30	40	sgd	0.904
27	model_2_8-30-9-adlt-trd	model_0 + reg. L2	Module_5	8000	30	9	adadelta	0.898
28	model_2_8-30-20-adlt-trd	model_0 + reg. L2	Module_5	8000	30	20	adadelta	0.930
29	model_2_8-100-25-RMSp-trd	model_0 + reg. L2	Module_5	8000	100	25	RMSprop(lr=1e-4)	0.872
30	model_3_8-100-100-sgd-trd	32+64+128+128+256	Module_6	8000	100	100	sgd	0.936
31	model_3_8-100-100-adlt-trd	32+64+128+128+256	Module_6	8000	100	100	adadelta	0.942
32	model_3_8-100-20-adlt-trd	32+64+128+128+256	Module_6	8000	100	20	adadelta	0.946
33	model_3_8-100-30-RMSp-trd	32+64+128+128+256	Module_6	8000	100	30	RMSprop(lr=1e-4)	0.932
34	model_4_8-100-100-sgd-trd	32+64+128+128+128	Module_7	8000	100	100	sgd	0.924
35	model_4_8-100-20-adlt-trd	32+64+128+128+128	Module_7	8000	100	20	adadelta	0.926
36	model_4_8-100-30-RMSp-trd	32+64+128+128+128	Module_7	8000	100	30	RMSprop(lr=1e-4)	0.946
37	model_4_8-100-40-RMSp-trd	32+64+128+128+128	Module_7	8000	100	40	RMSprop(lr=1e-4)	0.946
38	model_5_8-100-50-sgd-trd	vgg16 conv base None	Module_12	8000	100	50	sgd	0.904
39	model_5_8-100-70-sgd-trd	vgg16 conv base None	Module_13	8000	100	70	sgd	0.922
40	model_5_8-100-25-RMSp-trd	vgg16 conv base None	Module_14	8000	100	25	RMSprop(lr=1e-4)	0.910

	model_name	model_description	notebook	dataset_size	batch_size	epochs	optimizer	acc
41	model_6_8-100-50-sgd-trd	32+64+128+128	Module_15	8000	100	50	sgd	0.914
42	model_6_8-100-100-sgd-trd	32+64+128+128	Module_15	8000	100	100	sgd	0.932
43	model_6_8-100-40-adlt-trd	32+64+128+128	Module_15	8000	100	40	adadelta	0.952
44	model_6_8-100-17-adlt-trd	32+64+128+128	Module_15	8000	100	17	adadelta	0.934
45	model_6_8-100-25-RMSp-trd	32+64+128+128	Module_15	8000	100	25	RMSprop(lr=1e-4)	0.900
46	model_6_8-100-30-RMSp-trd	32+64+128+128	Module_15	8000	100	30	RMSprop(lr=1e-4)	0.952
47	model_7_8-100-30-0vgg16-trd	full vgg16 base frozen	Module_16	8000	100	30	sgd	0.978
48	model_7_8-100-12-0vgg16-trd	full vgg16 base frozen	Module_16	8000	100	12	sgd	0.964
49	model_7_8-100-20-1vgg16s-trd	last vgg16 layer training	Module_17	8000	100	20	sgd	0.982
50	model_7_8-100-12-1vgg16R-trd	last vgg16 layer training	Module_17	8000	100	12	RMSprop(lr=1e-4)	0.982
51	model_7_8-100-5-1vgg16s-trd	last vgg16 layer training	Module_17	8000	100	5	sgd	0.978
52	model_7_8-100-20-2vgg16s-trd	2 last vgg16 layers tr.	Module_18	8000	100	20	sgd	0.984
53	model_7_8-100-5-2vgg16s-trd	2 last vgg16 layers tr.	Module_18	8000	100	5	sgd	0.982
54	model_7_8-100-20-3vgg16s-trd	3 last vgg16 layers tr.	Module_19	8000	100	20	sgd	0.984
55	model_7_8-100-5-3vgg16s-trd	3 last vgg16 layers tr.	Module_19	8000	100	5	sgd	0.988
56	model_7_8-100-5-3vgg16s-trd_a	3 last vgg16 layers tr.	Module_19a	8000	100	5	sgd	0.986

5. Selected models' evaluation

Checking the models on test data.

In [8]:
```
train_x_name = 'TRAIN_X_EXTD.npy'
train_y_name = 'TRAIN_Y_EXTD.npy'
TRAIN_X, VALID_X, TEST_X, TRAIN_Y, VALID_Y, TEST_Y = my_functions.data_conversion(train_

my_functions.code_block_1(TEST_X, TEST_Y)
```

```
LOADED DATA (before conversion):
  set name              shape  dtype       data sight
0  TRAIN_X  (8000, 96, 96, 3)  uint8     [87, 73, 37]
1  TRAIN_Y            (8000,)  uint8                2
2  VALID_X   (500, 96, 96, 3)  uint8  [204, 205, 210]
3  VALID_Y             (500,)  uint8                1
4   TEST_X   (500, 96, 96, 3)  uint8  [132, 167, 203]
5   TEST_Y             (500,)  uint8                1

Using TensorFlow backend.


DATA FOR TRAINING (after conversion):
  set name              shape    dtype                              data sight
0  TRAIN_X  (8000, 96, 96, 3)  float32  [0.34117648, 0.28627452, 0.14509805]
1  TRAIN_Y          (8000, 2)  float32                              [0.0, 1.0]
2  VALID_X   (500, 96, 96, 3)  float32           [0.8, 0.8039216, 0.8235294]
3  VALID_Y           (500, 2)  float32                              [1.0, 0.0]
4   TEST_X   (500, 96, 96, 3)  float32     [0.5176471, 0.654902, 0.79607844]
5   TEST_Y           (500, 2)  float32                              [1.0, 0.0]

500/500 [==============================] - 2s 4ms/step

Model no. 33 (Module_6) , validation accuracy = 0.932 , test accuracy =  0.93599998950
95825

500/500 [==============================] - 0s 720us/step

Model no. 37 (Module_7) , validation accuracy = 0.946 , test accuracy =  0.94599997997
28394

500/500 [==============================] - 0s 686us/step

Model no. 46 (Module_15) , validation accuracy = 0.952 , test accuracy =  0.9340000152
587891

500/500 [==============================] - 3s 5ms/step

Model no. 56 (Module_19a) , validation accuracy = 0.986 , test accuracy =  0.981999993
3242798
```

In [ ]:
```

```