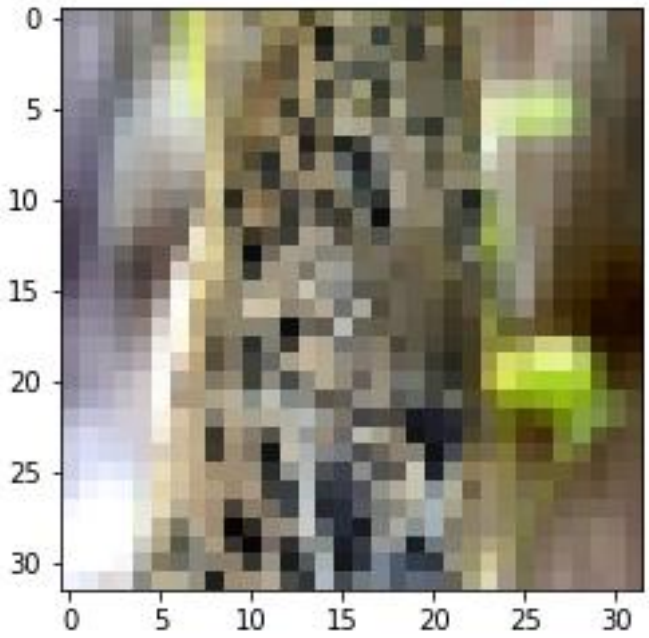
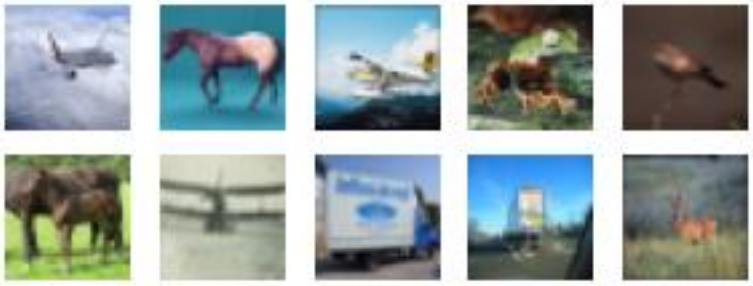
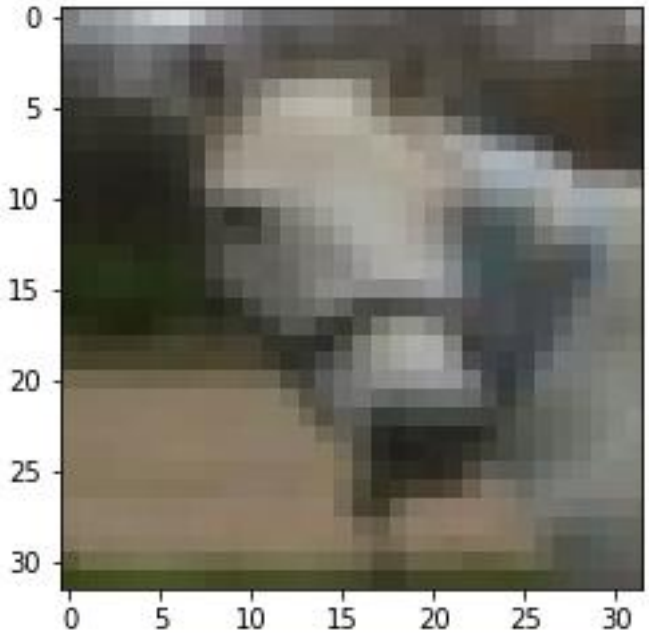
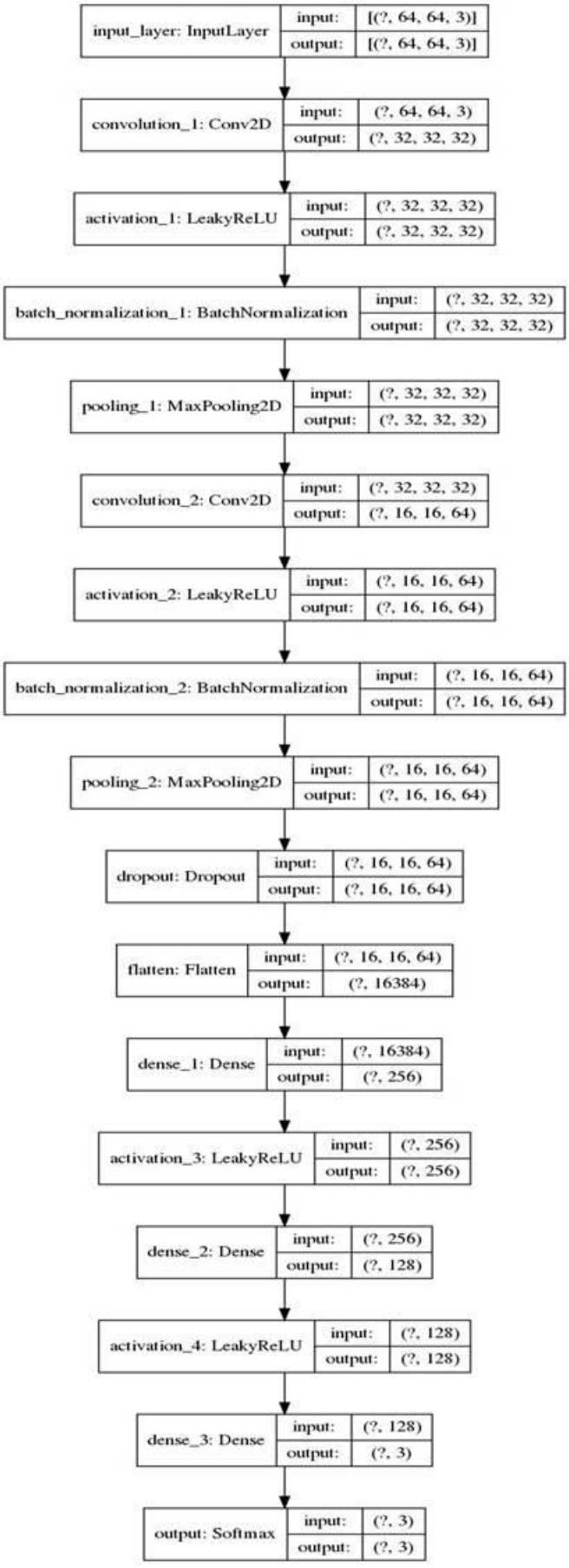
Chapter 1: Getting Started with TensorFlow 2.x for Computer Vision

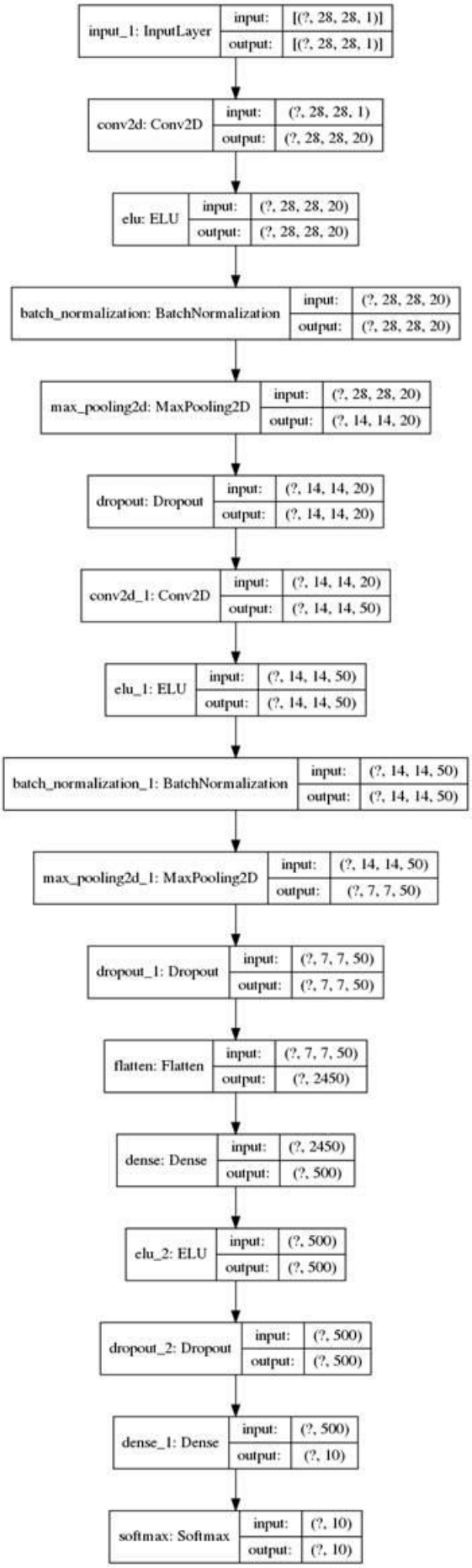


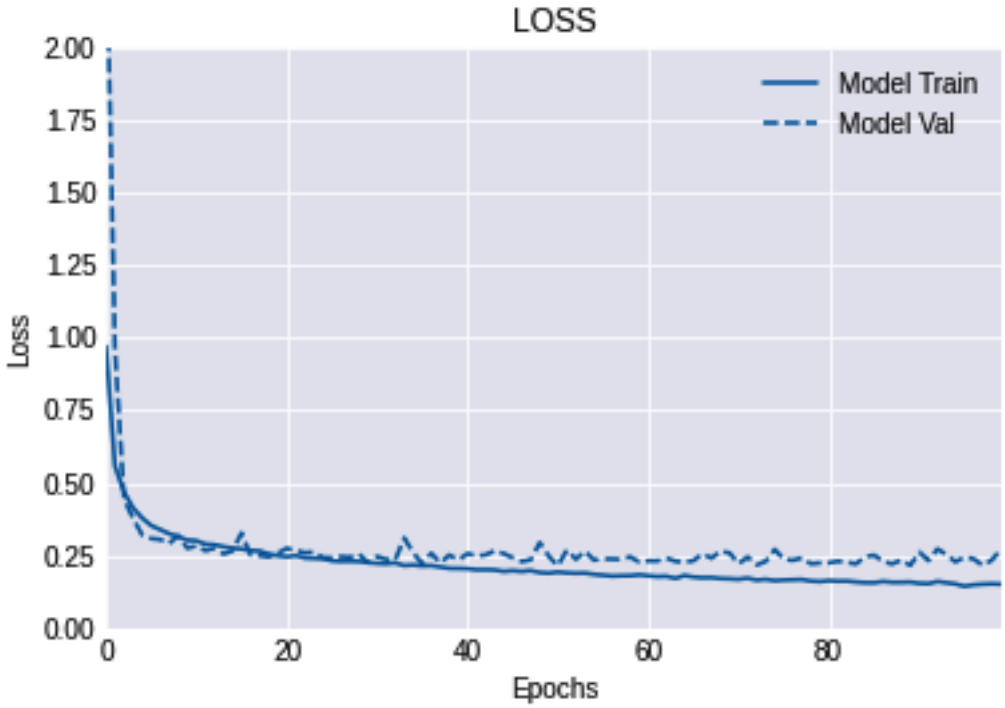


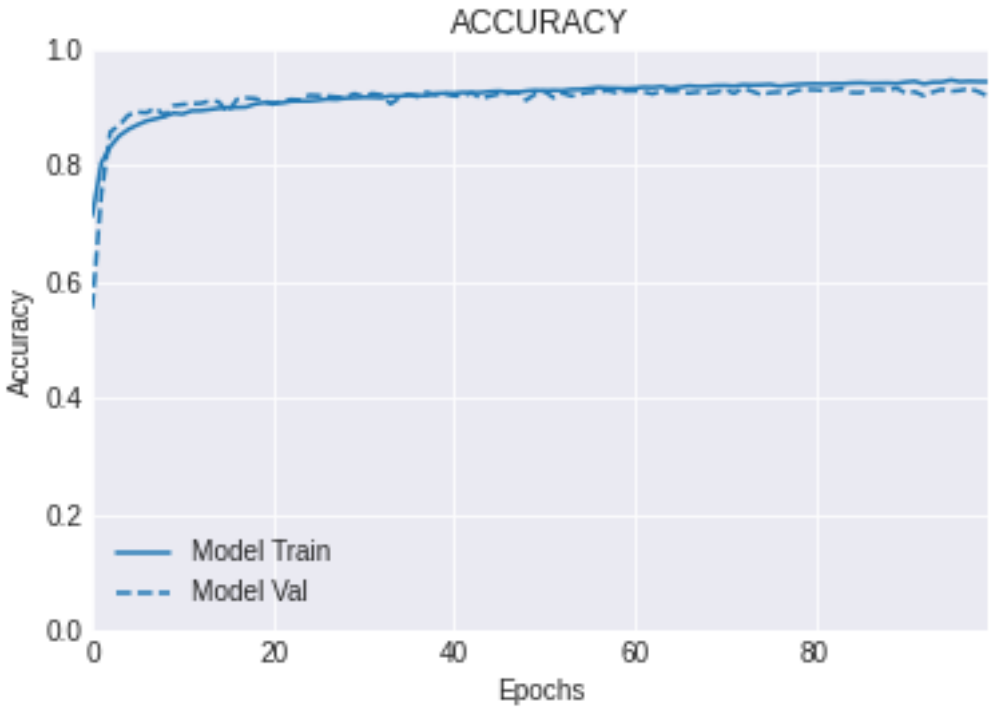

0
5
10
15
20
25
30
0
5
10
15
20
25
30

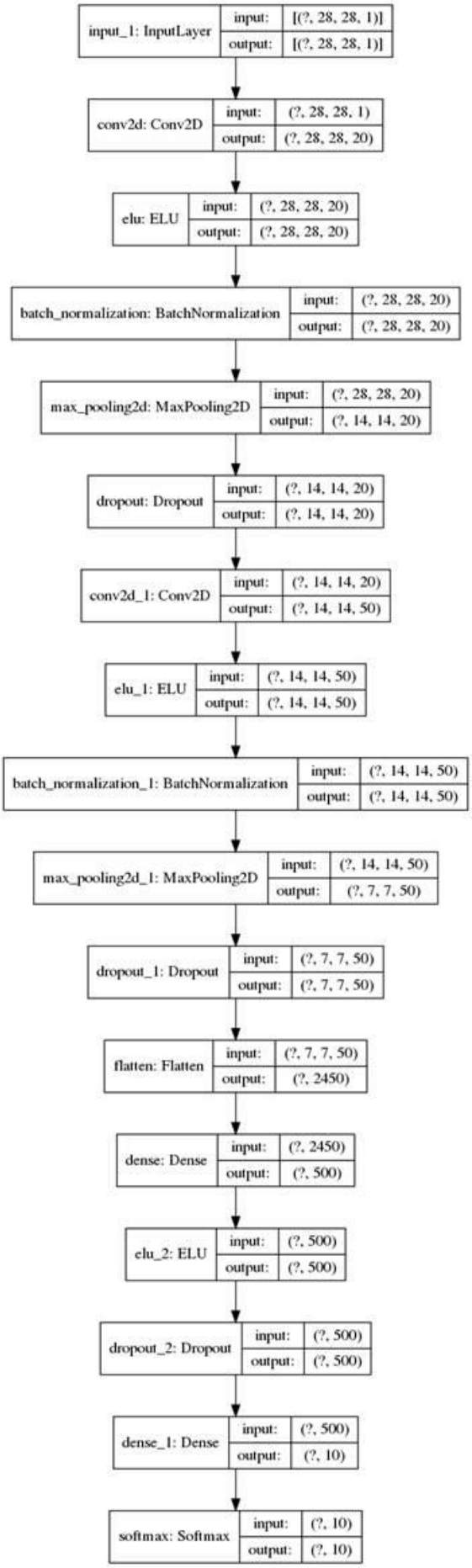
```
Model: "my_model"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_layer (InputLayer)     [(None, 64, 64, 3)]       0
_________________________________________________________________
convolution_1 (Conv2D)       (None, 32, 32, 32)        416
_________________________________________________________________
activation_1 (LeakyReLU)     (None, 32, 32, 32)        0
_________________________________________________________________
batch_normalization_1 (Batch (None, 32, 32, 32)        128
_________________________________________________________________
pooling_1 (MaxPooling2D)     (None, 32, 32, 32)        0
_________________________________________________________________
convolution_2 (Conv2D)       (None, 16, 16, 64)        8256
_________________________________________________________________
activation_2 (LeakyReLU)     (None, 16, 16, 64)        0
_________________________________________________________________
batch_normalization_2 (Batch (None, 16, 16, 64)        256
_________________________________________________________________
pooling_2 (MaxPooling2D)     (None, 16, 16, 64)        0
_________________________________________________________________
dropout (Dropout)            (None, 16, 16, 64)        0
_________________________________________________________________
flatten (Flatten)            (None, 16384)             0
_________________________________________________________________
dense_1 (Dense)              (None, 256)               4194560
_________________________________________________________________
activation_3 (LeakyReLU)     (None, 256)               0
_________________________________________________________________
dense_2 (Dense)              (None, 128)               32896
_________________________________________________________________
activation_4 (LeakyReLU)     (None, 128)               0
_________________________________________________________________
dense_3 (Dense)              (None, 3)                 387
_________________________________________________________________
output (Softmax)             (None, 3)                 0
=================================================================
Total params: 4,236,899
Trainable params: 4,236,707
Non-trainable params: 192
_________________________________________________________________
```


input_layer: InputLayer
input: [(?, 64, 64, 3)]
output: [(?, 64, 64, 3)]
convolution_1: Conv2D
input: (?, 64, 64, 3)
output: (?, 32, 32, 32)
activation_1: LeakyReLU
input: (?, 32, 32, 32)
output: (?, 32, 32, 32)
batch_normalization_1: BatchNormalization
input: (?, 32, 32, 32)
output: (?, 32, 32, 32)
pooling_1: MaxPooling2D
input: (?, 32, 32, 32)
output: (?, 32, 32, 32)
convolution_2: Conv2D
input: (?, 32, 32, 32)
output: (?, 16, 16, 64)
activation_2: LeakyReLU
input: (?, 16, 16, 64)
output: (?, 16, 16, 64)
batch_normalization_2: BatchNormalization
input: (?, 16, 16, 64)
output: (?, 16, 16, 64)
pooling_2: MaxPooling2D
input: (?, 16, 16, 64)
output: (?, 16, 16, 64)
dropout: Dropout
input: (?, 16, 16, 64)
output: (?, 16, 16, 64)
flatten: Flatten
input: (?, 16, 16, 64)
output: (?, 16384)
dense_1: Dense
input: (?, 16384)
output: (?, 256)
activation_3: LeakyReLU
input: (?, 256)
output: (?, 256)
dense_2: Dense
input: (?, 256)
output: (?, 128)
activation_4: LeakyReLU
input: (?, 128)
output: (?, 128)
dense_3: Dense
input: (?, 128)
output: (?, 3)
output: Softmax
input: (?, 3)
output: (?, 3)

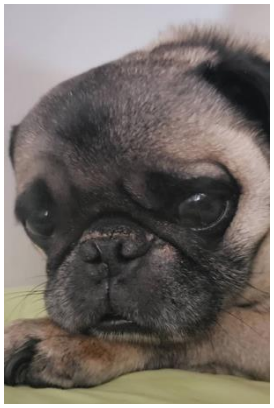

input_1: InputLayer
input: [(?, 28, 28, 1)]
output: [(?, 28, 28, 1)]
conv2d: Conv2D
input: (?, 28, 28, 1)
output: (?, 28, 28, 20)
elu: ELU
input: (?, 28, 28, 20)
output: (?, 28, 28, 20)
batch_normalization: BatchNormalization
input: (?, 28, 28, 20)
output: (?, 28, 28, 20)
max_pooling2d: MaxPooling2D
input: (?, 28, 28, 20)
output: (?, 14, 14, 20)
dropout: Dropout
input: (?, 14, 14, 20)
output: (?, 14, 14, 20)
conv2d_1: Conv2D
input: (?, 14, 14, 20)
output: (?, 14, 14, 50)
elu_1: ELU
input: (?, 14, 14, 50)
output: (?, 14, 14, 50)
batch_normalization_1: BatchNormalization
input: (?, 14, 14, 50)
output: (?, 14, 14, 50)
max_pooling2d_1: MaxPooling2D
input: (?, 14, 14, 50)
output: (?, 7, 7, 50)
dropout_1: Dropout
input: (?, 7, 7, 50)
output: (?, 7, 7, 50)
flatten: Flatten
input: (?, 7, 7, 50)
output: (?, 2450)
dense: Dense
input: (?, 2450)
output: (?, 500)
elu_2: ELU
input: (?, 500)
output: (?, 500)
dropout_2: Dropout
input: (?, 500)
output: (?, 500)
dense_1: Dense
input: (?, 500)
output: (?, 10)
softmax: Softmax
input: (?, 10)
output: (?, 10)

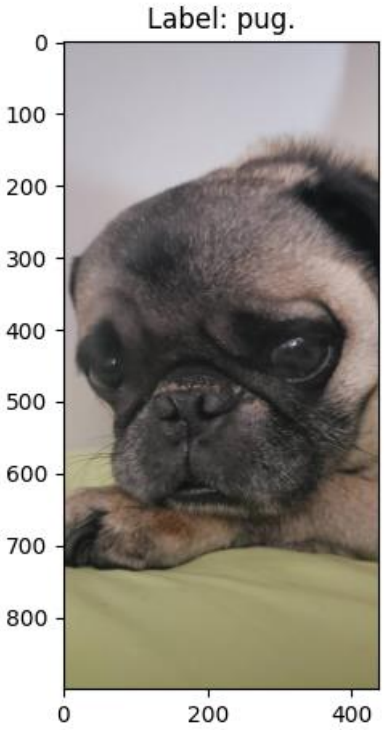

LOSS
Model Train
Model Val
Loss
Epochs
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
0
20
40
60
80


ACCURACY
Model Train
Model Val
Accuracy
Epochs
1.0
0.8
0.6
0.4
0.2
0.0
0
20
40
60
80

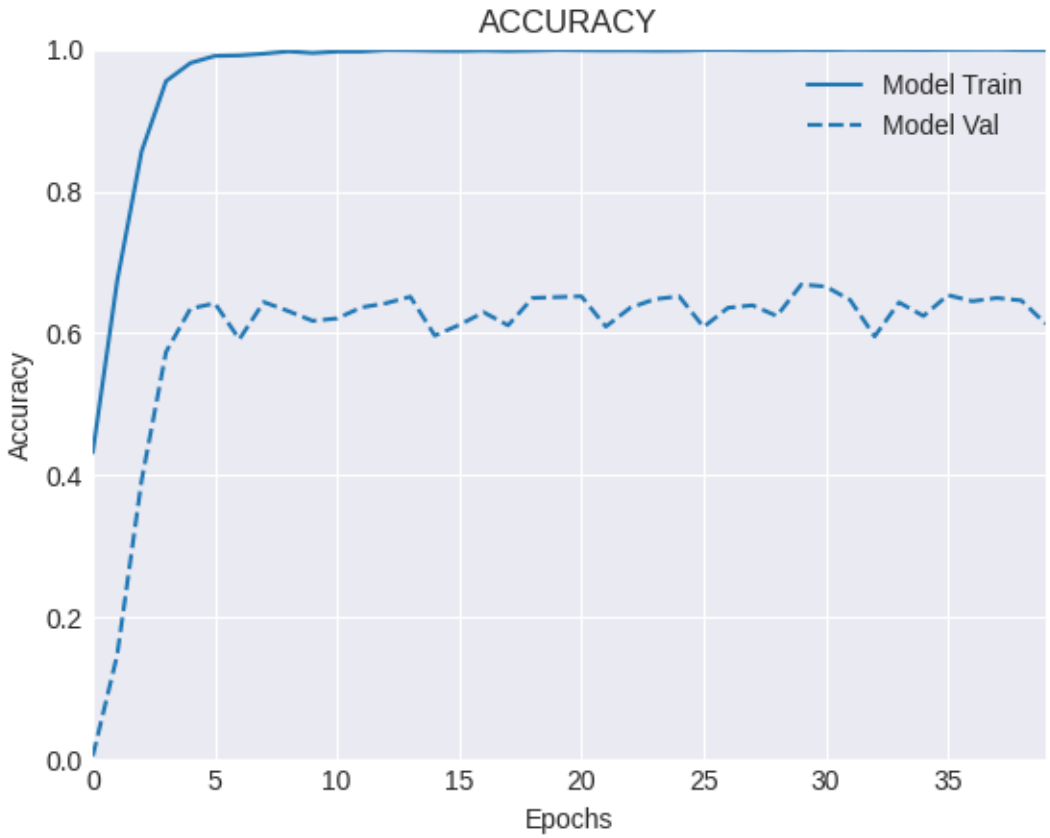

input_1: InputLayer
input: [(?, 28, 28, 1)]
output: [(?, 28, 28, 1)]
conv2d: Conv2D
input: (?, 28, 28, 1)
output: (?, 28, 28, 20)
elu: ELU
input: (?, 28, 28, 20)
output: (?, 28, 28, 20)
batch_normalization: BatchNormalization
input: (?, 28, 28, 20)
output: (?, 28, 28, 20)
max_pooling2d: MaxPooling2D
input: (?, 28, 28, 20)
output: (?, 14, 14, 20)
dropout: Dropout
input: (?, 14, 14, 20)
output: (?, 14, 14, 20)
conv2d_1: Conv2D
input: (?, 14, 14, 20)
output: (?, 14, 14, 50)
elu_1: ELU
input: (?, 14, 14, 50)
output: (?, 14, 14, 50)
batch_normalization_1: BatchNormalization
input: (?, 14, 14, 50)
output: (?, 14, 14, 50)
max_pooling2d_1: MaxPooling2D
input: (?, 14, 14, 50)
output: (?, 7, 7, 50)
dropout_1: Dropout
input: (?, 7, 7, 50)
output: (?, 7, 7, 50)
flatten: Flatten
input: (?, 7, 7, 50)
output: (?, 2450)
dense: Dense
input: (?, 2450)
output: (?, 500)
elu_2: ELU
input: (?, 500)
output: (?, 500)
dropout_2: Dropout
input: (?, 500)
output: (?, 500)
dense_1: Dense
input: (?, 500)
output: (?, 10)
softmax: Softmax
input: (?, 10)
output: (?, 10)

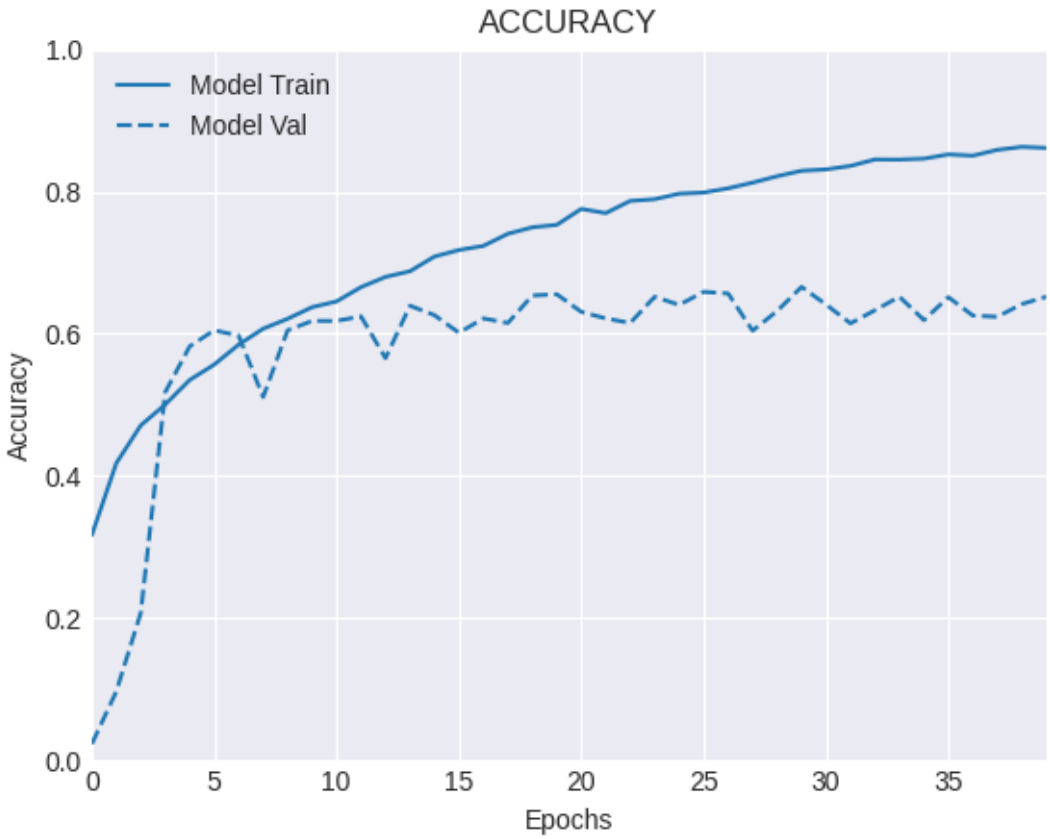
Chapter 2: Performing Image Classification

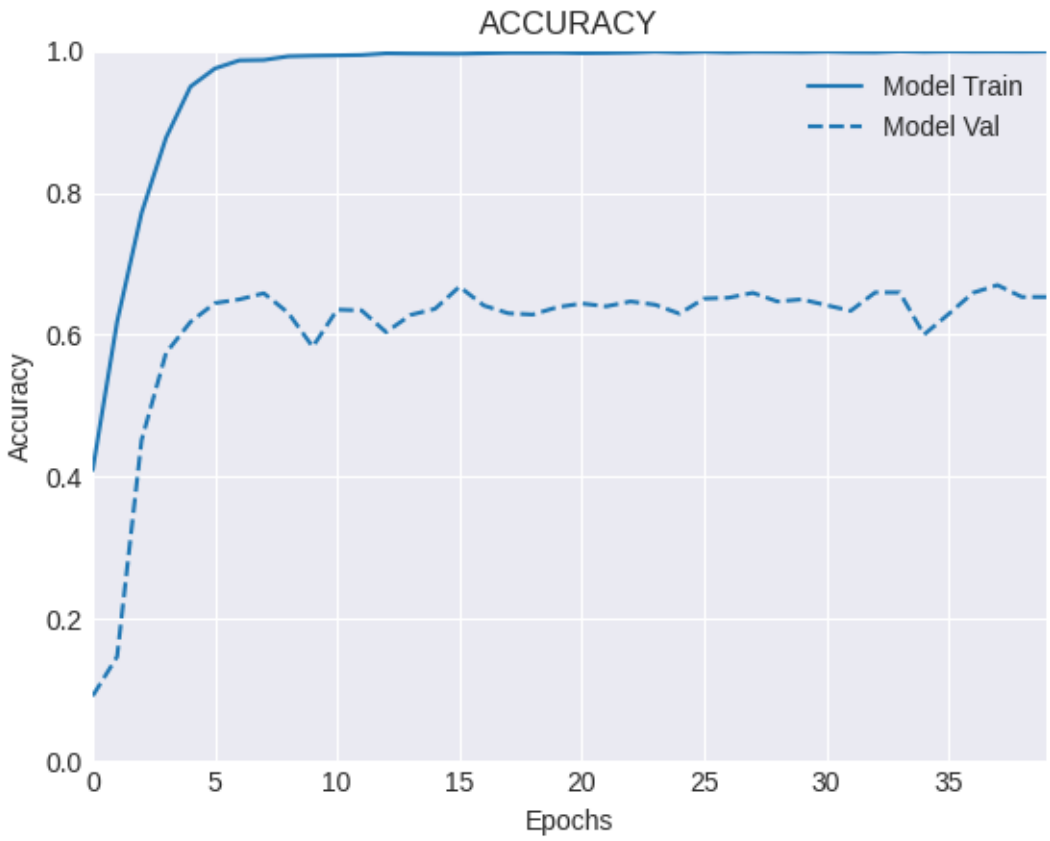


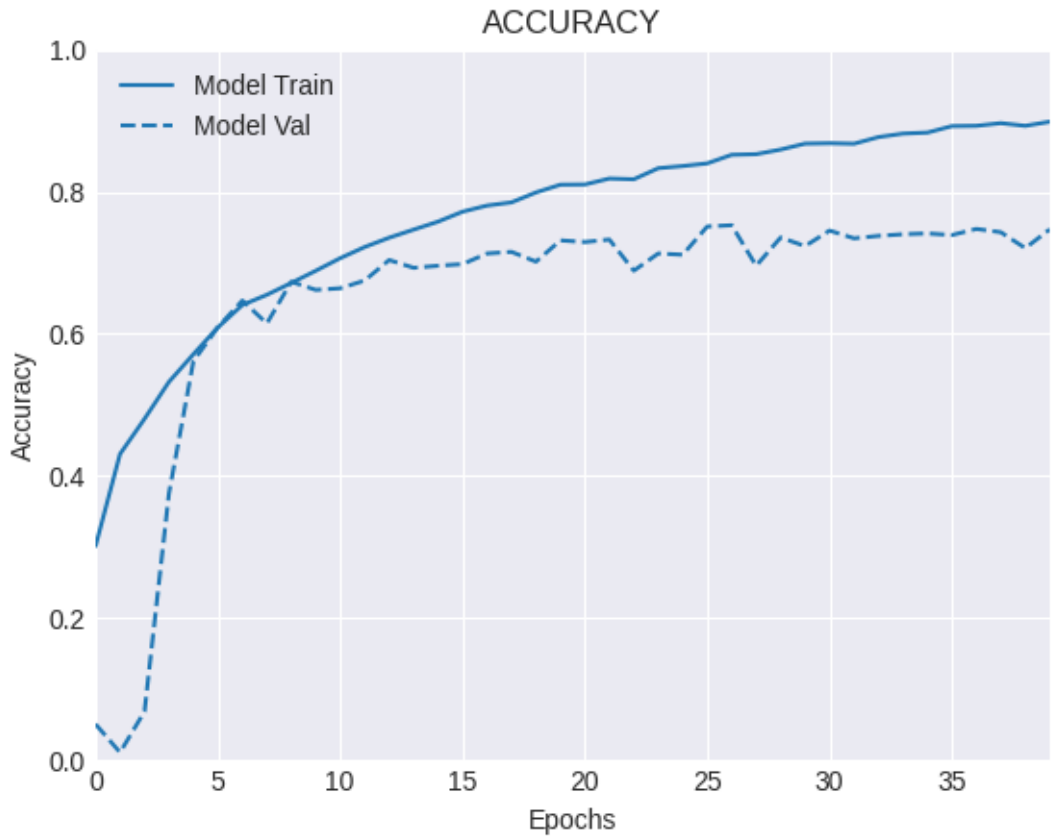

Label: pug.
0
100
200
300
400
500
600
700
800
0
200
400


Label: convertible.
0
500
1000
1500
2000
0
500
1000
1500
2000
2500
3000
3500
4000


ACCURACY
Model Train
Model Val
Accuracy
Epochs
1.0
0.8
0.6
0.4
0.2
0.0
0
5
10
15
20
25
30
35


ACCURACY
Model Train
Model Val
Accuracy
Epochs
1.0
0.8
0.6
0.4
0.2
0.0
0
5
10
15
20
25
30
35

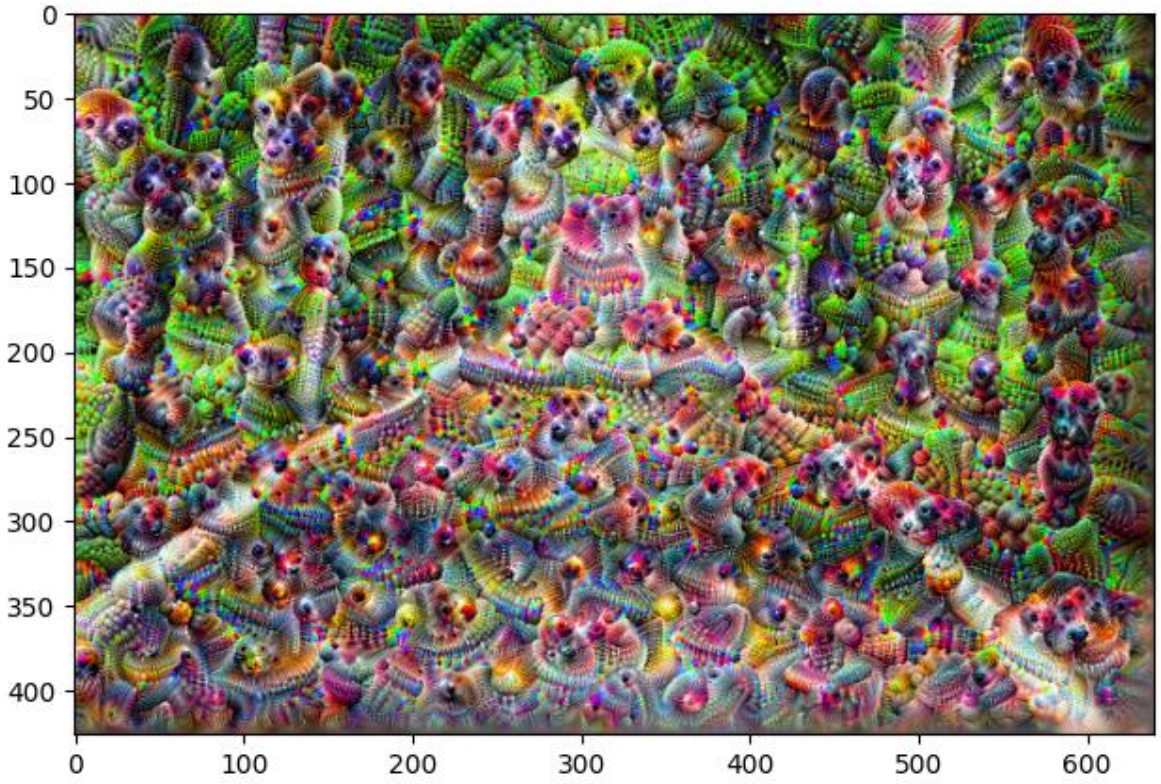

ACCURACY
Model Train
Model Val
Accuracy
Epochs
1.0
0.8
0.6
0.4
0.2
0.0
0
5
10
15
20
25
30
35

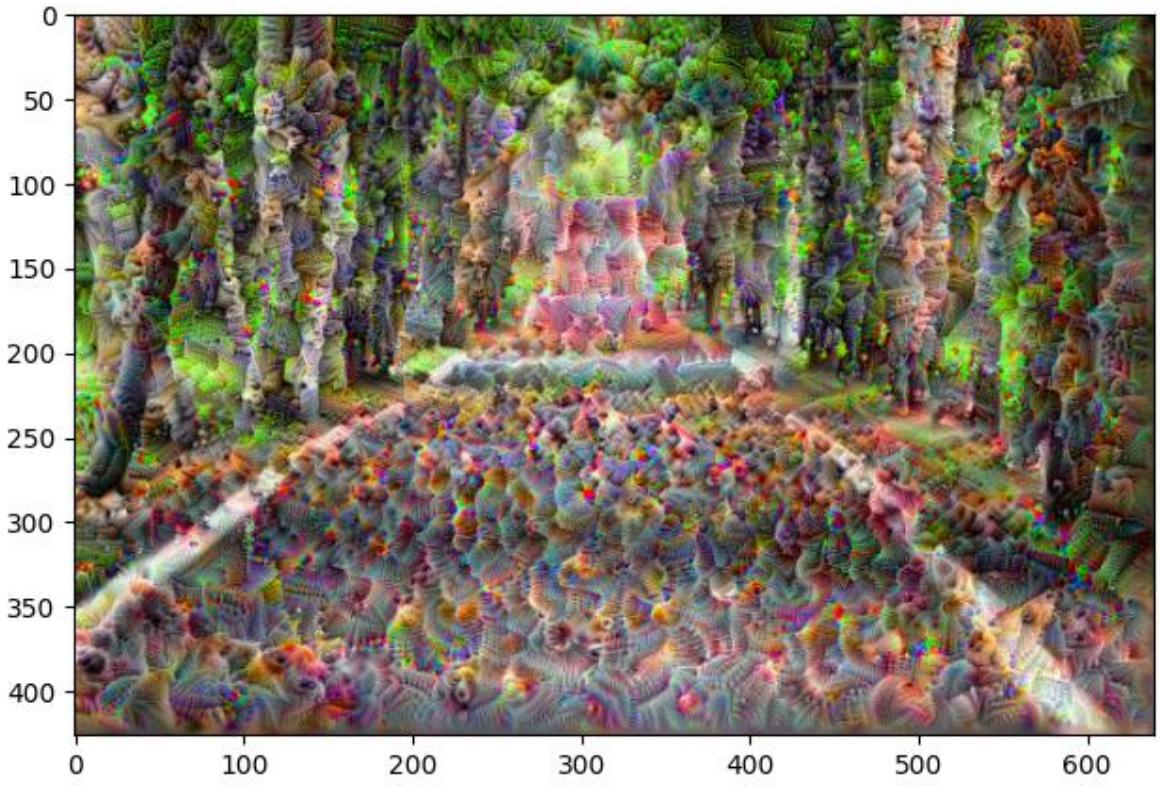

ACCURACY
Model Train
Model Val
Accuracy
Epochs
1.0
0.8
0.6
0.4
0.2
0.0
0
5
10
15
20
25
30
35

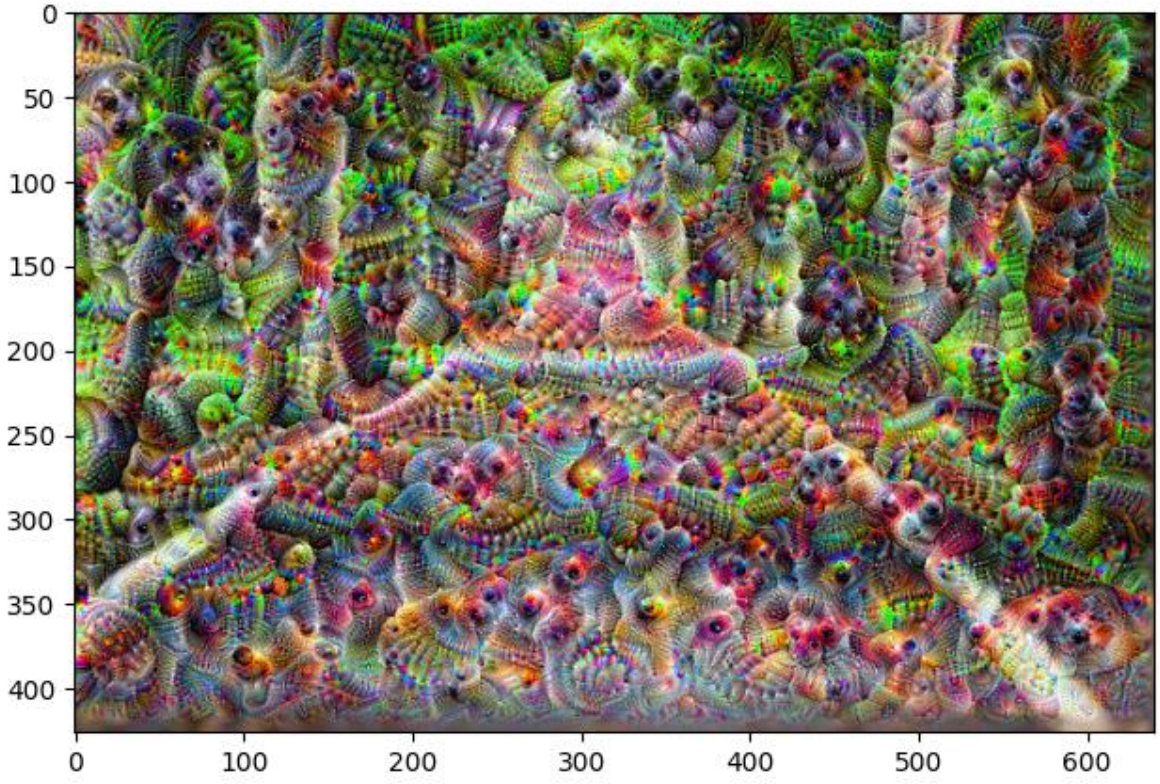
Chapter 3: Harnessing the Power of Pre-Trained Networks with Transfer Learning

Chapter 4: Enhancing and Styling Images with DeepDream, Neural Style Transfer and Image Super-Resolution


0
50
100
150
200
250
300
350
400
0
100
200
300
400
500
600

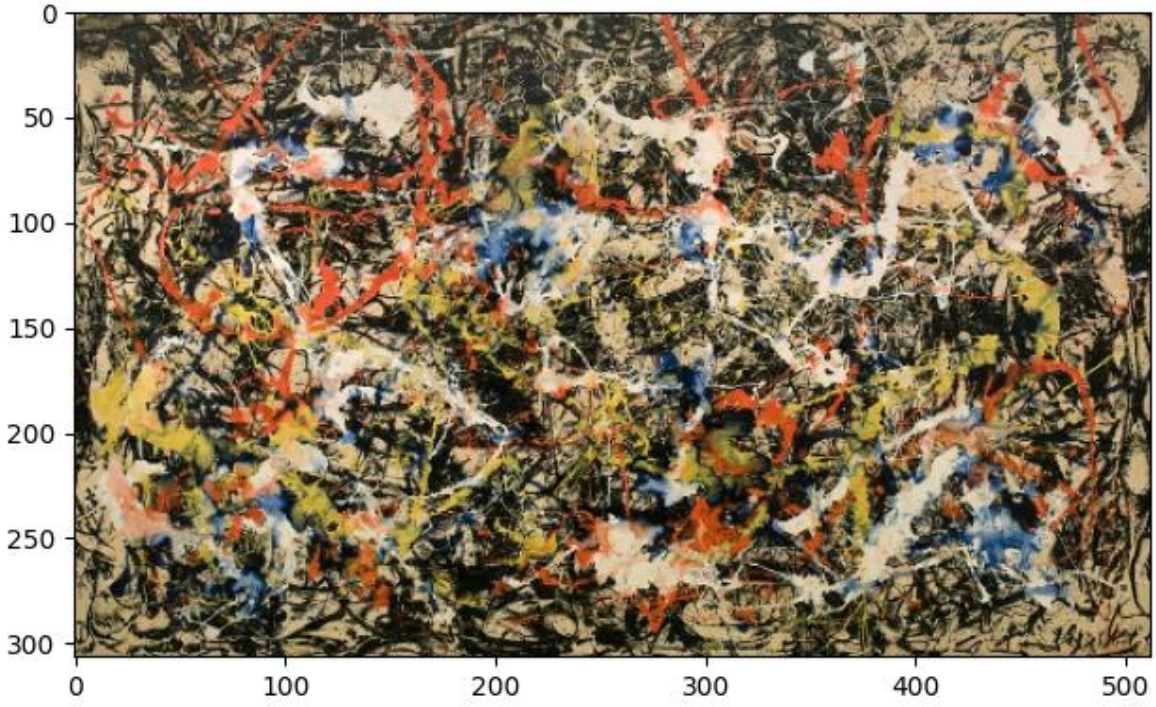

0
50
100
150
200
250
300
350
400
0
100
200
300
400
500
600

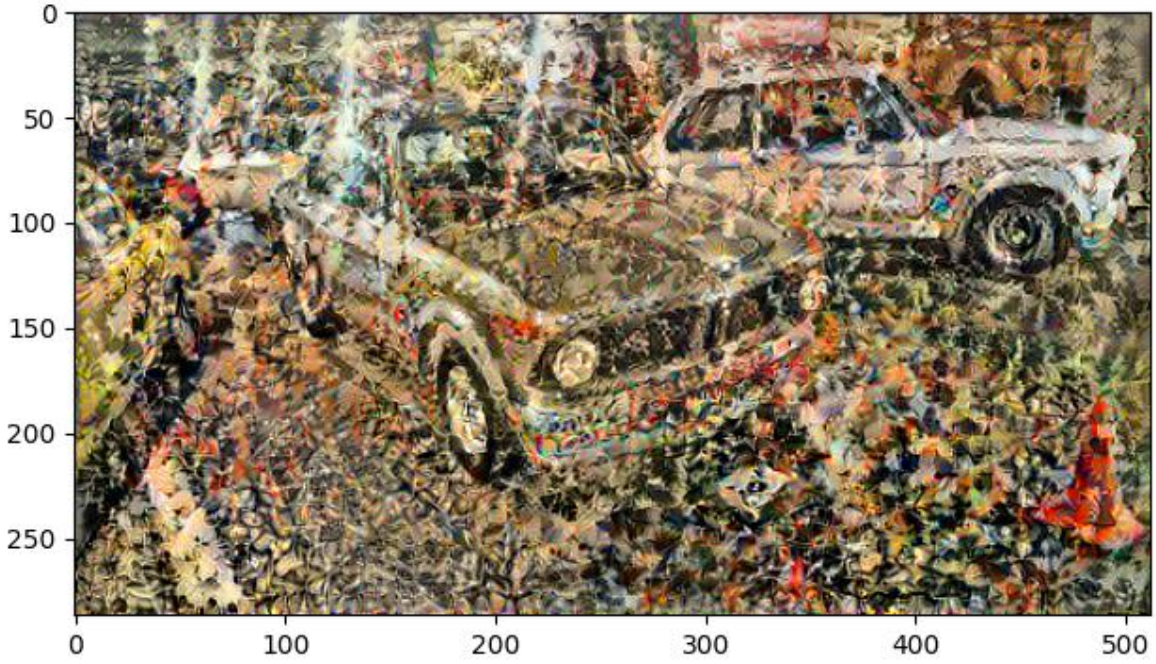

0
50
100
150
200
250
300
350
400
0
100
200
300
400
500
600

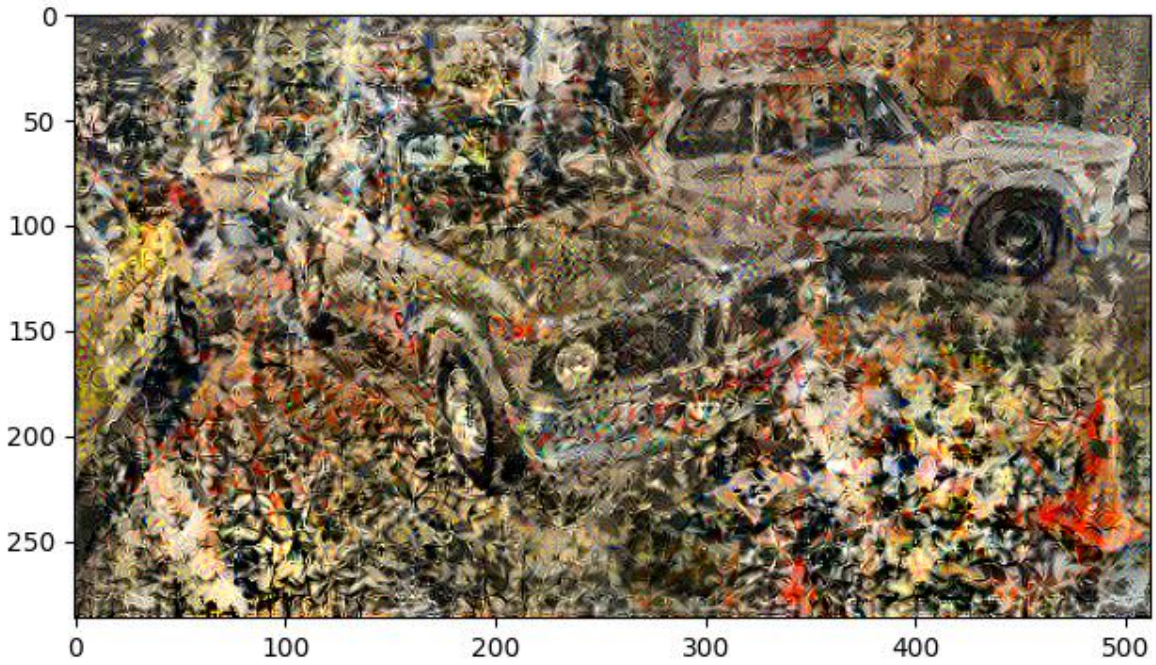

0
50
100
150
200
250
300
350
400
0
100
200
300
400
500
600


BMW 2002 Turbo
62


0
50
100
150
200
250
0
100
200
300
400
500

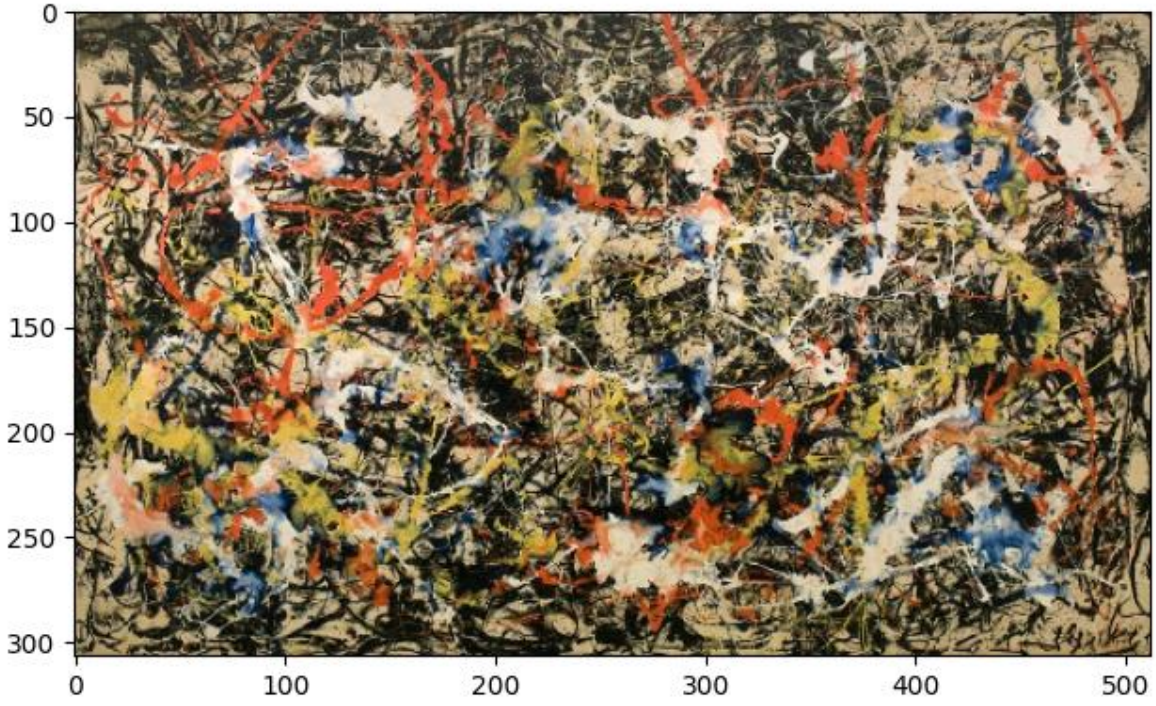

0
50
100
150
200
250
300
0
100
200
300
400
500

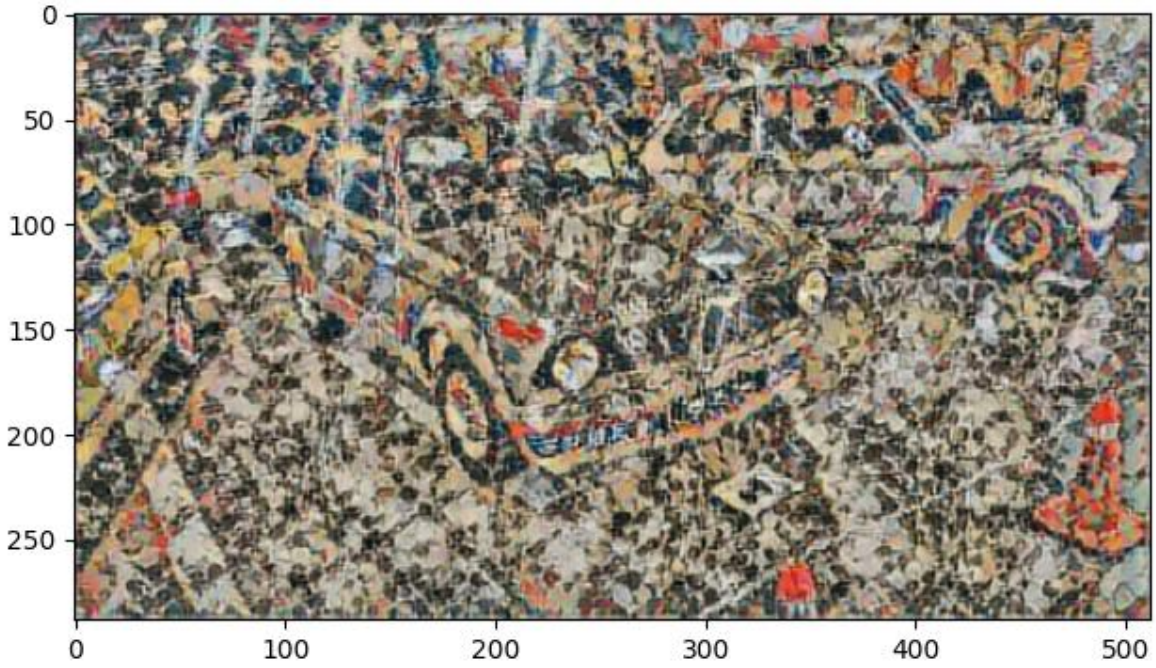

0
50
100
150
200
250
0
100
200
300
400
500

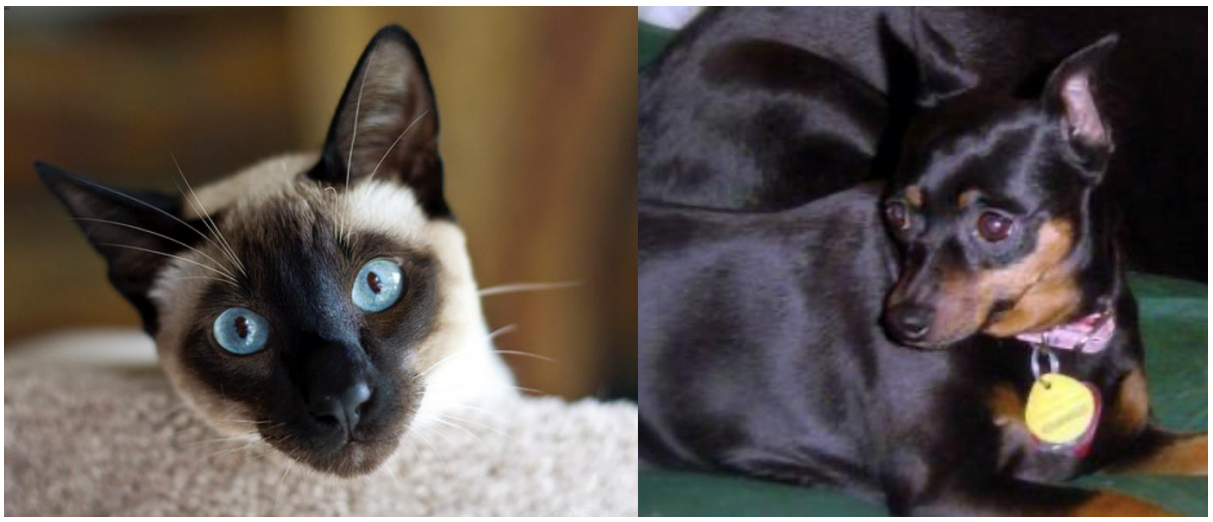

0
50
100
150
200
250
0
100
200
300
400
500


BMW 2002 Turbo
52

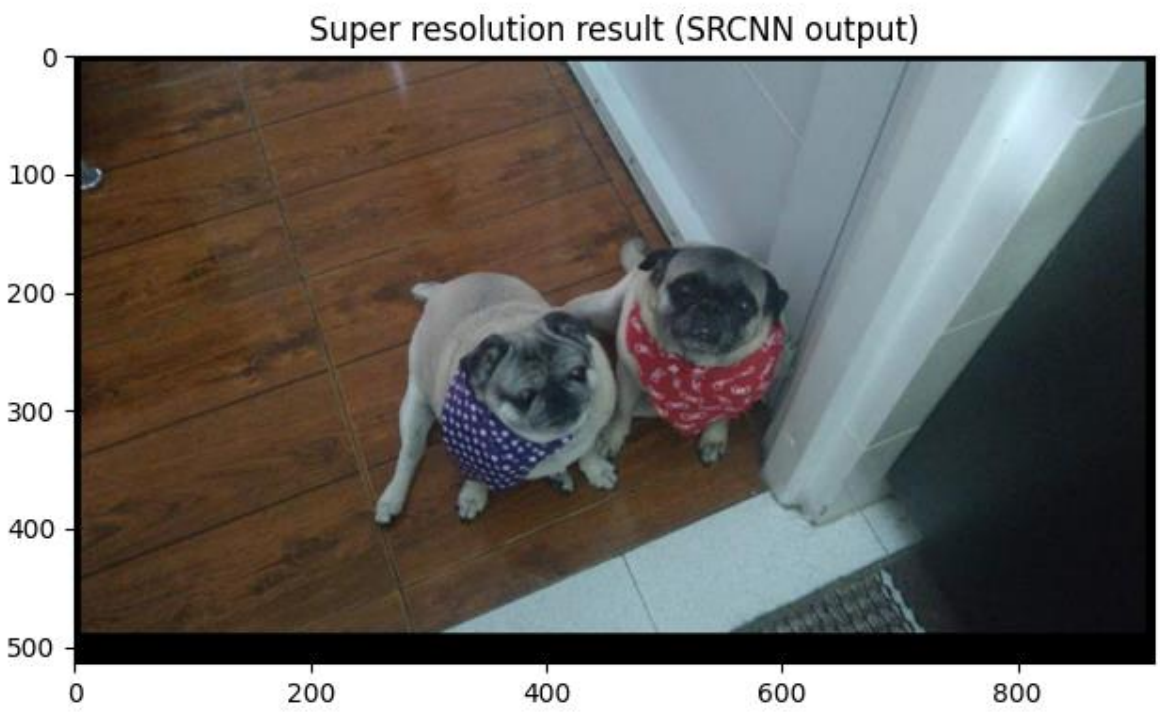

0
50
100
150
200
250
0
100
200
300
400
500


0
50
100
150
200
250
300
0
100
200
300
400
500

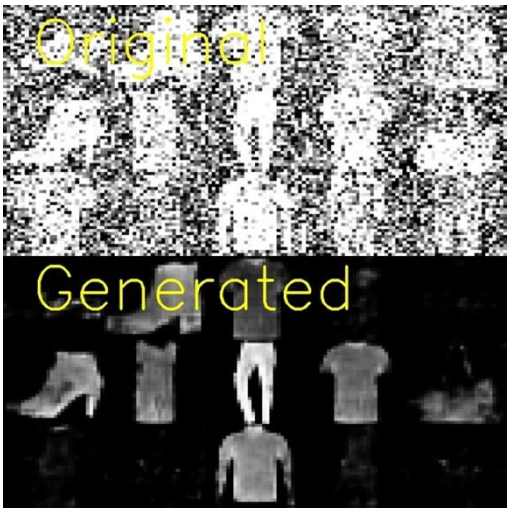

0
50
100
150
200
250
0
100
200
300
400
500

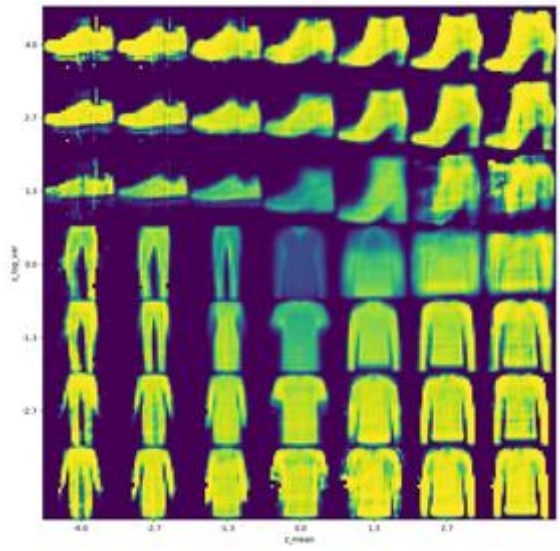



Low resolution image (Downsize + Upsize)
0
100
200
300
400
500
0
200
400
600
800


Super resolution result (SRCNN output)
0
100
200
300
400
500
0
200
400
600
800

Chapter 5: Reducing Noise with Autoencoders




4.0
2.7
1.3
0.0
-1.3
-2.7
z_log_var
-4.0
-2.7
-1.3
0.0
1.3
2.7
z_mean

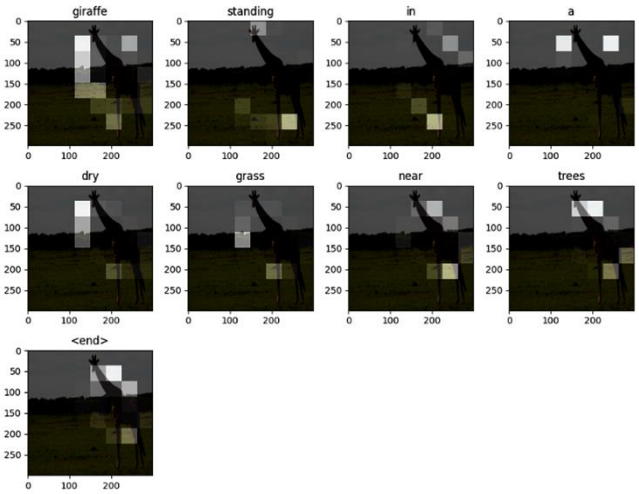
Chapter 6: Generative Models and Adversarial Attacks

Chapter 7: Captioning Images with CNNs and RNNs

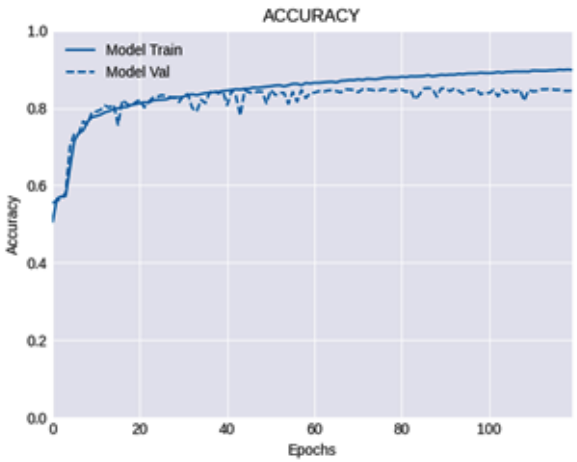

football player in red uniform in the air
0
50
100
150
200
250
300
0
100
200
300
400

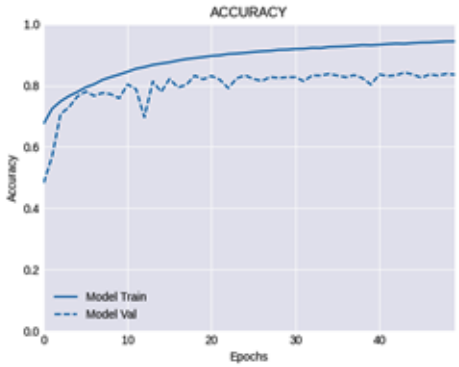

AIRFRANCE

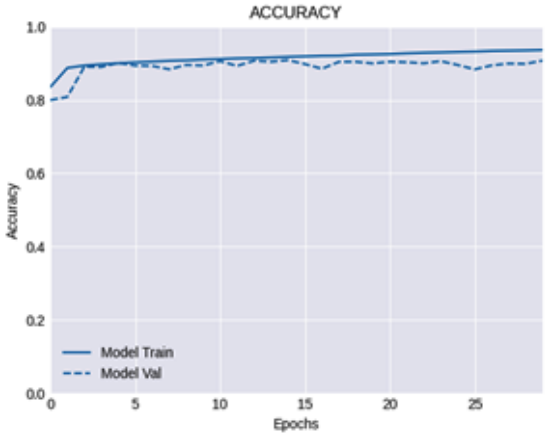

man in blue shirt is jumping on the street
0
50
100
150
200
250
300
350
0
100
200
300
400

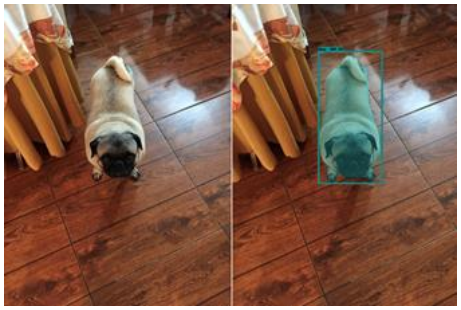

giraffe
standing
in
a
dry
grass
near
trees
<end>

Chapter 8: Fine-Grained Understanding of Images through Segmentation

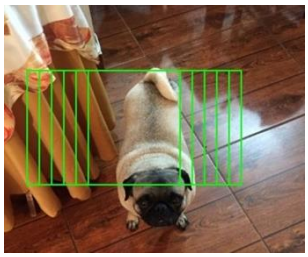


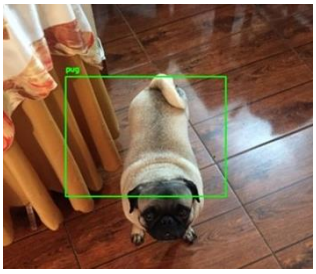

ACCURACY
Accuracy
Epochs
Model Train
Model Val
0.0
0.2
0.4
0.6
0.8
1.0
0
10
20
30
40

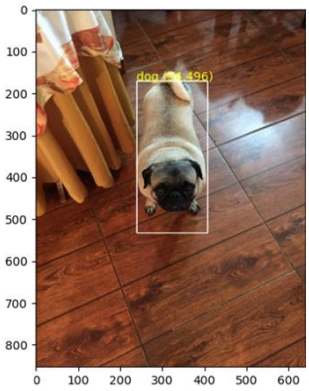

ACCURACY
Accuracy
Epochs
Model Train
Model Val
0.0
0.2
0.4
0.6
0.8
1.0
0
5
10
15
20
25

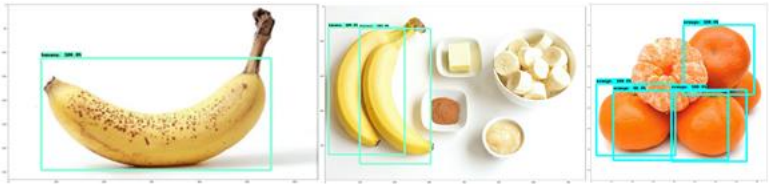


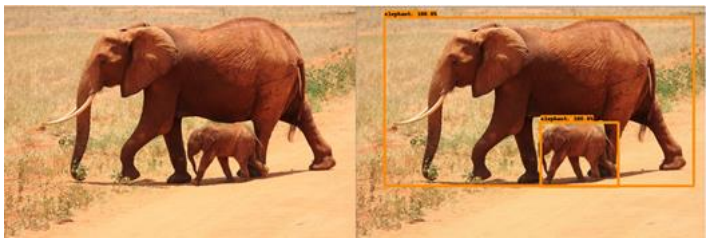
Chapter 9: Localizing Elements in Images with Object Detection

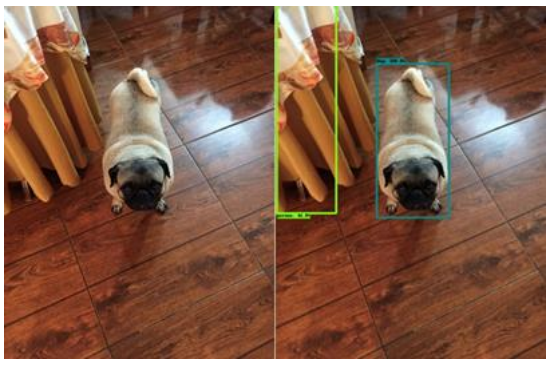










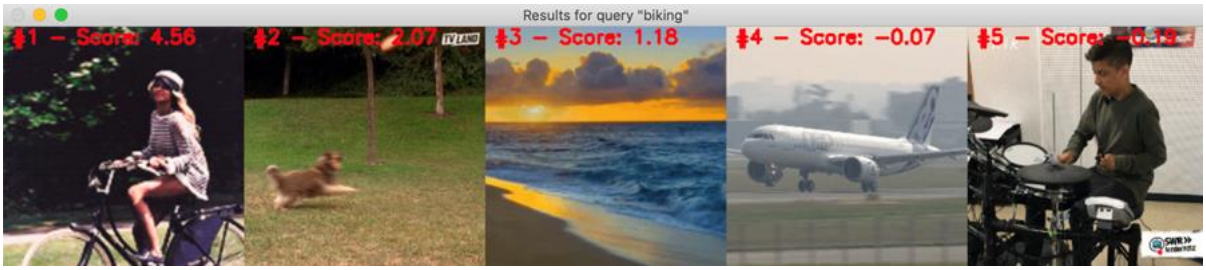


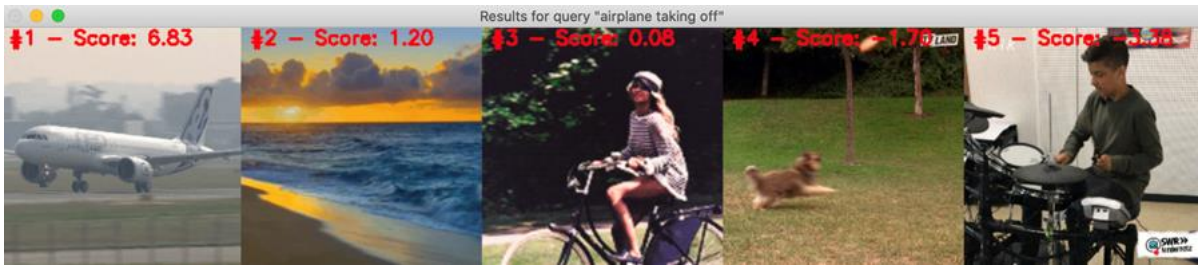
Chapter 10: Applying the Power of Deep Learning to Videos


Video 0: first frame
Video 0: last frame
Video 1: first frame
Video 1: last frame
Video 2: first frame
Video 2: last frame
Video 3: first frame
Video 3: last frame


Results for query "beach"
#1 – Score: 4.56
#2 – Score: 1.93
#3 – Score: 1.06
#4 – Score: 0.23
#5 – Score: -2.66

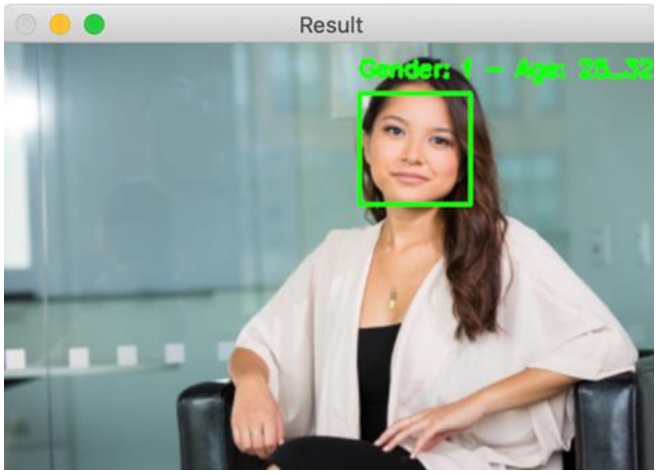

Results for query "playing drums"
#1 – Score: 7.35
#2 – Score: 1.55
#3 – Score: 0.83
#4 – Score: 0.77
#5 – Score: –1.72


Results for query "biking"
#1 – Score: 4.56
#2 – Score: 2.07
#3 – Score: 1.18
#4 – Score: –0.07
#5 – Score:


Results for query "airplane taking off"
#1 – Score: 6.83
#2 – Score: 1.20
#3 – Score: 0.08
#4 – Score:
#5 – Score:


Results for query "dog catching frisbee"
#1 – Score: 8.27
#2 – Score: 2.43
#3 – Score:
#4 – Score: –2.21
#5 – Score: –2.53

Chapter 11: Streamlining Network Implementation with AutoML



Index of /adiencedb/AdienceBenchmarkC

[ICO]	Name	Last modified	Size	Description
[PARENTDIR]	Parent Directory		-	
[TXT]	(old a) fold_0_data.txt	2014-11-20 16:36	355K	
[TXT]	(old a) fold_1_data.txt	2014-11-20 16:36	297K	
[TXT]	(old a) fold_2_data.txt	2014-11-20 16:36	310K	
[TXT]	(old a) fold_3_data.txt	2014-11-20 16:36	279K	
[TXT]	(old a) fold_4_data.txt	2014-11-20 16:36	307K	
[TXT]	(old a) fold_frontal_>	2014-11-20 16:36	253K	
[TXT]	(old a) fold_frontal_>	2014-11-20 16:36	242K	
[TXT]	(old a) fold_frontal_>	2014-11-20 16:36	190K	
[TXT]	(old a) fold_frontal_>	2014-11-20 16:36	202K	
[TXT]	(old a) fold_frontal_>	2014-11-20 16:36	192K	
[TXT]	LICENSE.txt	2016-11-22 20:35	1.8K	
[ ]	aligned.tar.gz	2014-06-18 16:51	2.6G	
[ ]	faces.tar.gz	2014-06-18 15:04	1.2G	
[TXT]	fold_0_data.txt	2014-12-15 09:57	355K	
[TXT]	fold_1_data.txt	2014-12-15 09:57	297K	
[TXT]	fold_2_data.txt	2014-12-15 09:57	310K	
[TXT]	fold_3_data.txt	2014-12-15 09:57	279K	
[TXT]	fold_4_data.txt	2014-12-15 09:57	307K	
[TXT]	fold_frontal_0_data.txt	2014-12-15 09:57	253K	
[TXT]	fold_frontal_1_data.txt	2014-12-15 09:57	242K	
[TXT]	fold_frontal_2_data.txt	2014-12-15 09:57	190K	
[TXT]	fold_frontal_3_data.txt	2014-12-15 09:57	202K	
[TXT]	fold_frontal_4_data.txt	2014-12-15 09:57	192K	

Index of /adiencedb

[ICO]	Name	Last modified	Size
[PARENTDIR]	Parent Directory		-
[DIR]	AdienceBenchmarkOfUn..>	2014-12-15 09:59	-

CSlab FTP SERVER

Tal Hassner's datasets are availble from (same username and password as FTP server)

Adience OUI Unfiltered faces for gender and age classification

Action Similarity Labeling benchmark (ASLAN)

Face frontalization MATLAB code and LFW3D

Violent Flows benchmark and data set

YouTube Faces (YTF) data set

Download

The data is stored on our FTP server: *agas.openu.ac.il*

Alternatively, direct access to the download directory is available through:
http://www.cslab.openu.ac.il/download/

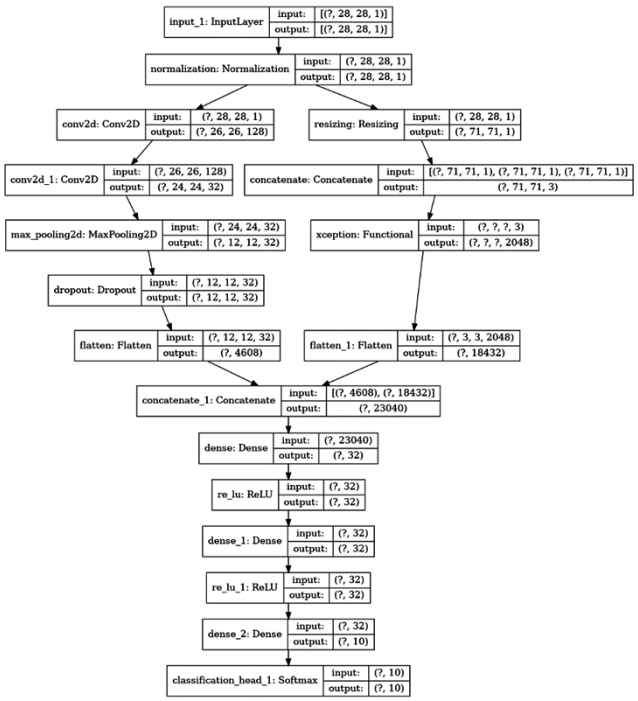
To get access to the data, please provide the details below. We ask for your details so we can keep in touch in case we find any need for a critical update or in case we would organize a dedicated workshop, etc. When done, you will immediately be able to see the FTP user name and password.

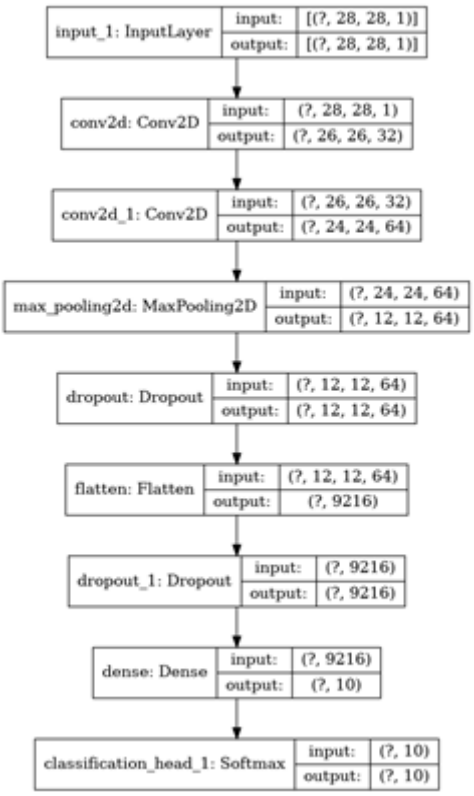
Your name * Jesús

Your e-mail * jesus@datasmarts.net

* Required field

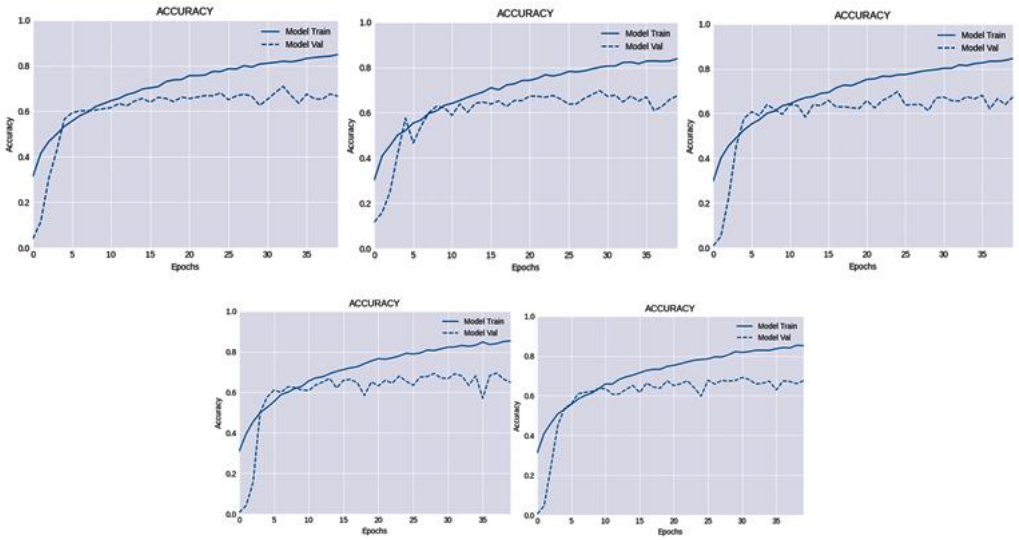
Submit

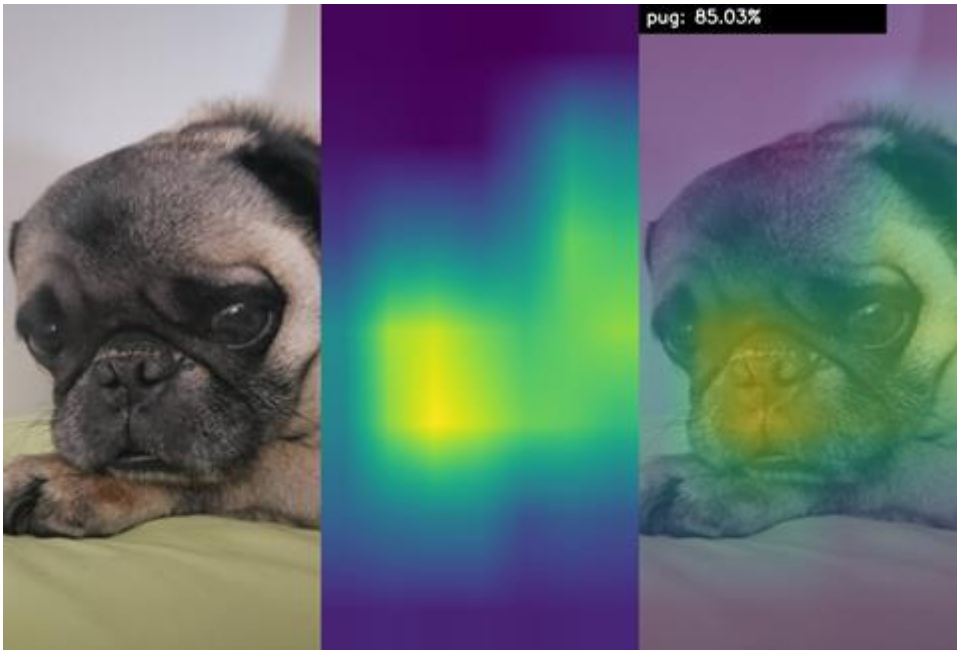




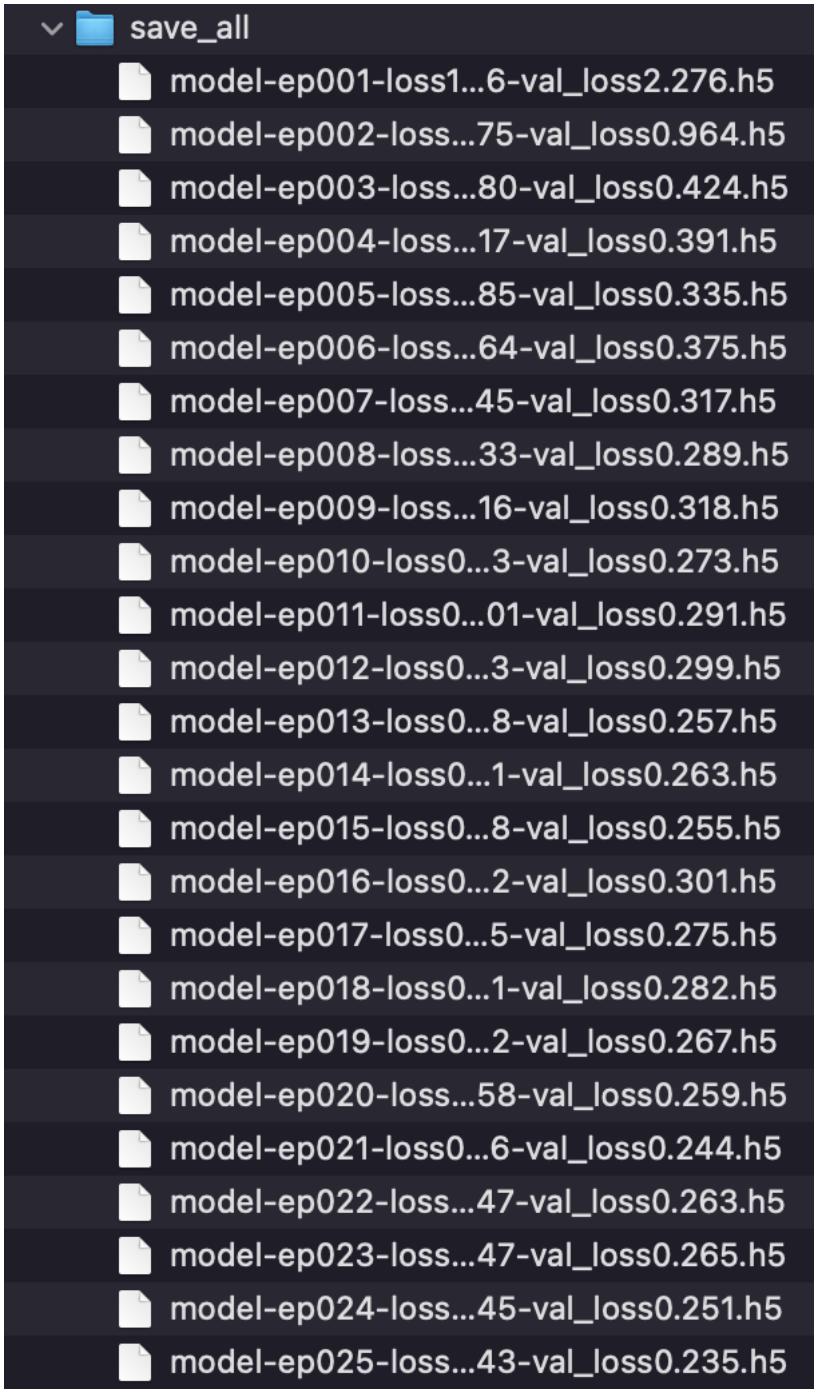

```
Model: "functional_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         [(None, 28, 28, 1)]       0
_________________________________________________________________
conv2d (Conv2D)              (None, 26, 26, 32)        320
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 24, 24, 64)        18496
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 12, 12, 64)        0
_________________________________________________________________
dropout (Dropout)            (None, 12, 12, 64)        0
_________________________________________________________________
flatten (Flatten)            (None, 9216)              0
_________________________________________________________________
dropout_1 (Dropout)          (None, 9216)              0
_________________________________________________________________
dense (Dense)                (None, 10)                92170
_________________________________________________________________
classification_head_1 (Softm (None, 10)                0
=================================================================
Total params: 110,986
Trainable params: 110,986
Non-trainable params: 0
_________________________________________________________________
None
```

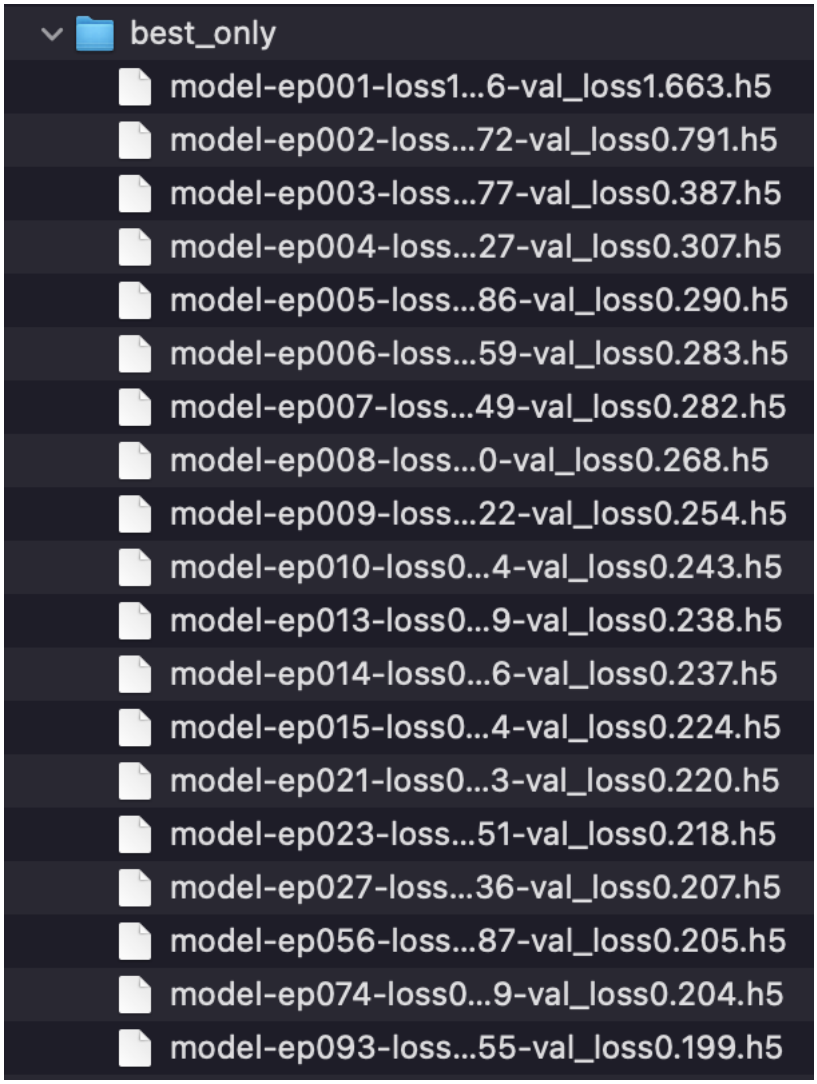
Chapter 12: Boosting Performance




pug: 85.03%


overwrite
model.h5


save_all
model-ep001-loss1...6-val_loss2.276.h5
model-ep002-loss...75-val_loss0.964.h5
model-ep003-loss...80-val_loss0.424.h5
model-ep004-loss...17-val_loss0.391.h5
model-ep005-loss...85-val_loss0.335.h5
model-ep006-loss...64-val_loss0.375.h5
model-ep007-loss...45-val_loss0.317.h5
model-ep008-loss...33-val_loss0.289.h5
model-ep009-loss...16-val_loss0.318.h5
model-ep010-loss0...3-val_loss0.273.h5
model-ep011-loss0...01-val_loss0.291.h5
model-ep012-loss0...3-val_loss0.299.h5
model-ep013-loss0...8-val_loss0.257.h5
model-ep014-loss0...1-val_loss0.263.h5
model-ep015-loss0...8-val_loss0.255.h5
model-ep016-loss0...2-val_loss0.301.h5
model-ep017-loss0...5-val_loss0.275.h5
model-ep018-loss0...1-val_loss0.282.h5
model-ep019-loss0...2-val_loss0.267.h5
model-ep020-loss...58-val_loss0.259.h5
model-ep021-loss0...6-val_loss0.244.h5
model-ep022-loss...47-val_loss0.263.h5
model-ep023-loss...47-val_loss0.265.h5
model-ep024-loss...45-val_loss0.251.h5
model-ep025-loss...43-val_loss0.235.h5


best_only
model-ep001-loss1...6-val_loss1.663.h5
model-ep002-loss...72-val_loss0.791.h5
model-ep003-loss...77-val_loss0.387.h5
model-ep004-loss...27-val_loss0.307.h5
model-ep005-loss...86-val_loss0.290.h5
model-ep006-loss...59-val_loss0.283.h5
model-ep007-loss...49-val_loss0.282.h5
model-ep008-loss...0-val_loss0.268.h5
model-ep009-loss...22-val_loss0.254.h5
model-ep010-loss0...4-val_loss0.243.h5
model-ep013-loss0...9-val_loss0.238.h5
model-ep014-loss0...6-val_loss0.237.h5
model-ep015-loss0...4-val_loss0.224.h5
model-ep021-loss0...3-val_loss0.220.h5
model-ep023-loss...51-val_loss0.218.h5
model-ep027-loss...36-val_loss0.207.h5
model-ep056-loss...87-val_loss0.205.h5
model-ep074-loss0...9-val_loss0.204.h5
model-ep093-loss...55-val_loss0.199.h5