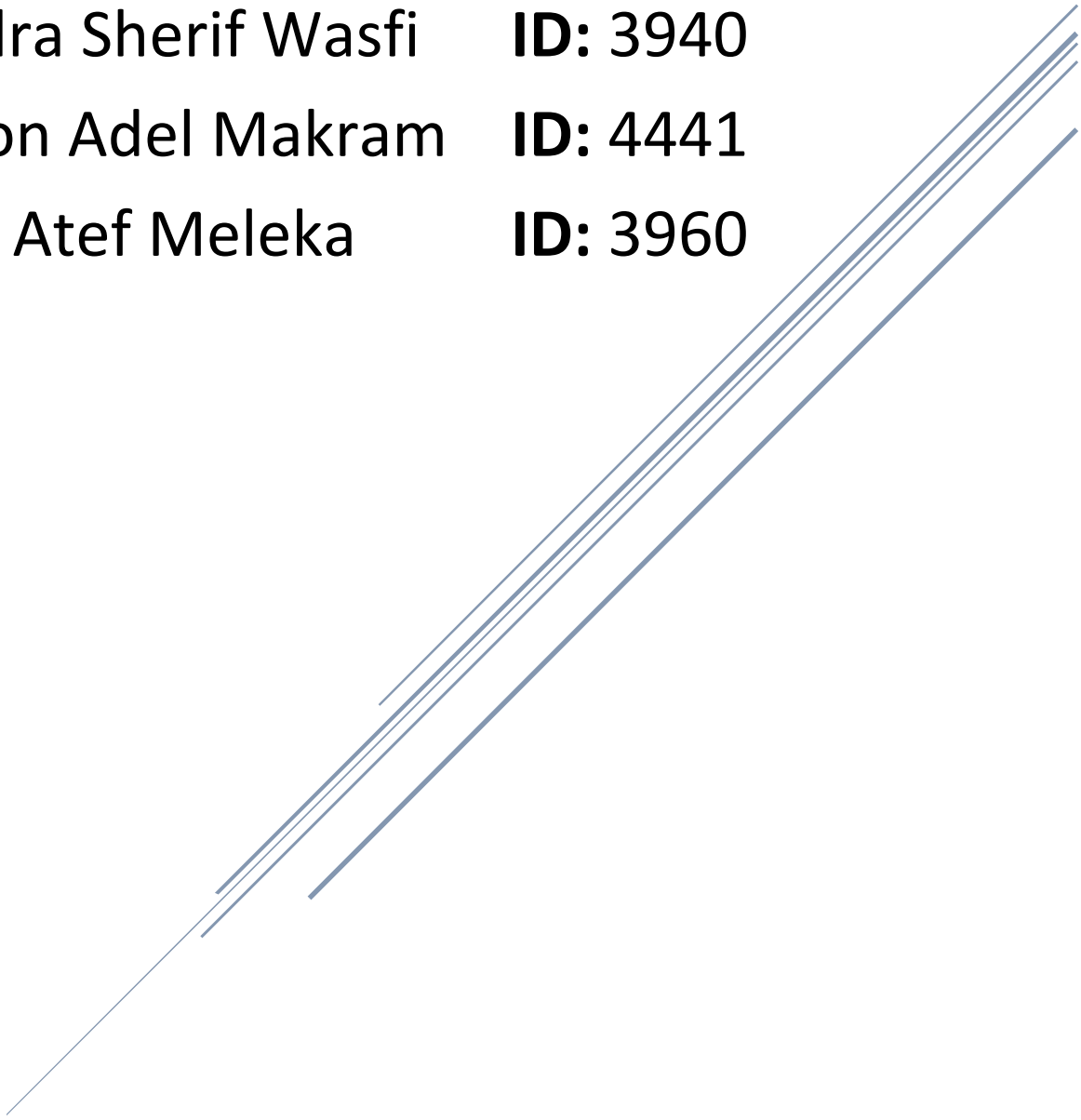
CIFAR-10 IMAGE CLASSIFICATION

Sandra Sherif Wasfi **ID:** 3940

Rimon Adel Makram **ID:** 4441

Adel Atef Meleka **ID:** 3960



INTRODUCTION

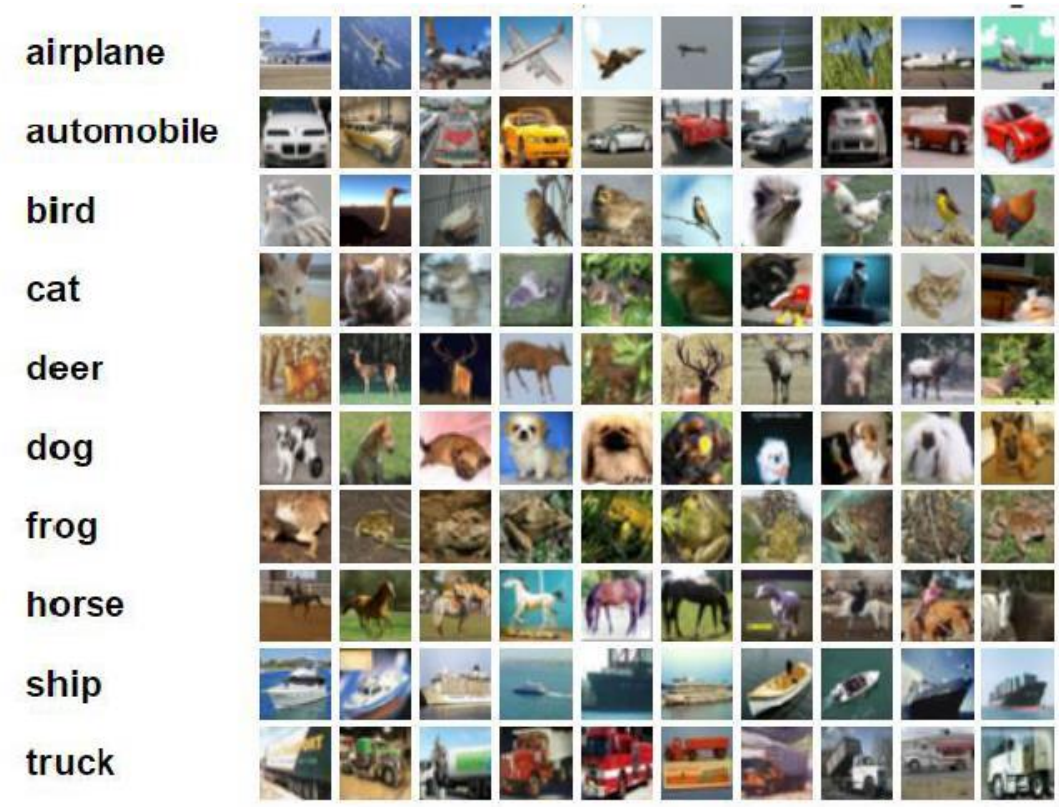
- In Convolutional Neural Networks (CNNs), a large part of the work is to choose the right layer to apply, among the most common options (1x1 filter, 3x3 filter 5x5 filter or max pooling). All we need is to find the optimal local construction and to repeat it spatially.

- As these “Inception modules” are stacked on top of each other, their output correlation statistics are bound to vary: as features of higher abstraction are captured by higher layers, their spatial concentration is expected to decrease suggesting that the ratio of 3×3 and 5×5 convolutions should increase as we move to higher layers.

DATA LOAD & PREPROCESSING

- The CIFAR-10 dataset consists of 60,000 32x32 color images in 10 classes, with 6000 images per class. The classes are completely mutually exclusive. There is no overlap between automobiles and trucks. "Automobile" includes sedans, SUVs, things of that sort. "Truck" includes only big trucks. Neither includes pickup trucks. You can download the dataset from here.

- Here are the classes in the dataset, as well as 10 random images from each:



- We implemented some data preprocessing means:
  1. Normalize all pictures of the training & test sets
  2. Convert training and test label to 1 hot encoding.

CONVOLUTION NEURAL NETWROK (CNN)

- Convolutional neural networks (CNNs) are a special kind of neural networks for processing data that has a known grid-like topology such as time-series data and image data. The general structure of a CNN consists of

one or more convolutional layers followed by one or more fully connected layers as in a standard multi-layer neural network as illustrated in Figure 2. This constrained architecture allows the CNN to leverage the spatial and temporal structure of the input and allows the network to learn more complex features from different parts of the input.

- Typically, a convolutional layer consists of three stages:

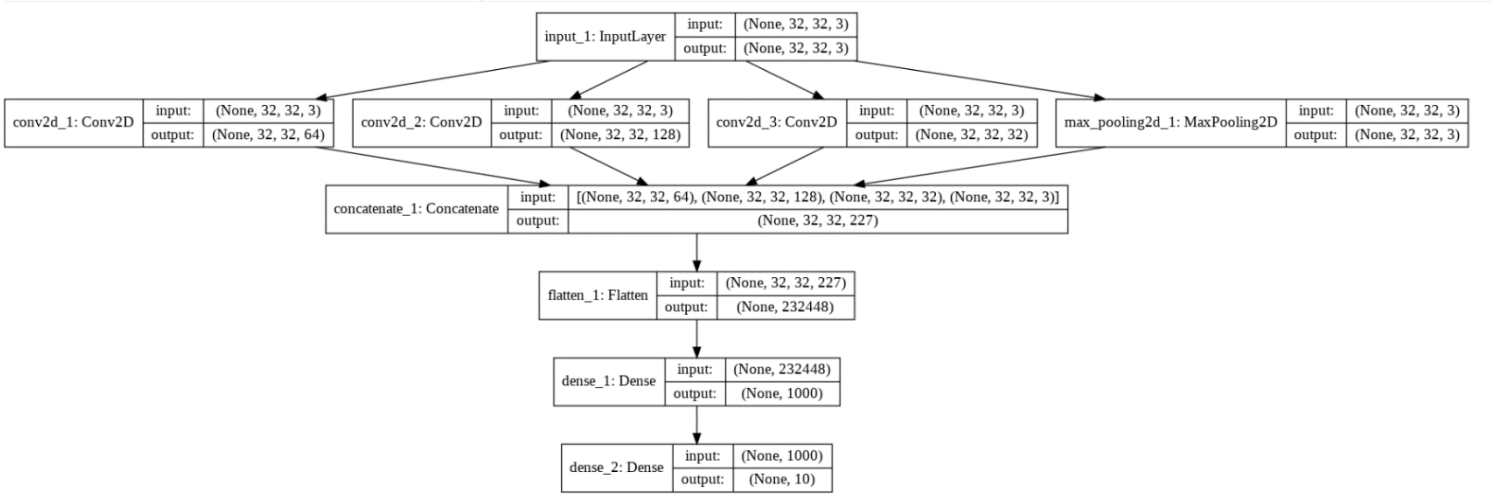
  1. In the first stage, the layer applies a kernel to its input by performing several convolutions in parallel to produce a set of linear activations.
  2. In the second stage, each linear activation is run through a nonlinear activation function, such as the recti ed linear activation function.
  3. In the third stage, a pooling function is used to further modify the output of the layer. Pooling functions replace the output of the net at a certain location with a summary statistic of the nearby outputs. For example, the max pooling operation reports the maximum output within a rectangular neighborhood. Other popular pooling functions include the average of a rectangular neighborhood or a weighted average based on the distance from the central pixel.

INCEPTION V1: NAÏVE VERSION

- Having filters with multiple sizes operate on the same level? The network essentially would get a bit “wider” rather than “deeper”. The authors designed the inception module to reflect the same.

- The “naive” inception module performs convolution on an input, with 3 different sizes of filters (1x1, 3x3, 5x5). Additionally, max pooling is also performed. The outputs are concatenated and sent to the next inception module.

- Model Architecture:



```
    OPERATION          DATA DIMENSIONS   WEIGHTS(N)   WEIGHTS(%)

        Input   #####     32   32    3
    InputLayer    |   --------------------        0     0.0%
                #####     32   32    3
       Conv2D    \|/  --------------------      256     0.0%
         relu   #####     32   32   64
       Conv2D    \|/  --------------------     3584     0.0%
         relu   #####     32   32  128
       Conv2D    \|/  --------------------     2432     0.0%
         relu   #####     32   32   32
    MaxPooling2D   Y max --------------------        0     0.0%
                #####     32   32    3
     Concatenate   ????? --------------------        0     0.0%
                #####     32   32  227
         Flatten   ||||| --------------------        0     0.0%
                #####      232448
           Dense   XXXXX --------------------  232449000   100.0%
         relu   #####        1000
           Dense   XXXXX --------------------     10010     0.0%
       softmax   #####          10
```

➢ Model Results:

```
40000/40000 [==============================] - 222s 6ms/step - loss: 1.8379 - acc: 0.3395 - val_loss: 1.6550 - val_acc: 0.4111
Epoch 2/10
40000/40000 [==============================] - 212s 5ms/step - loss: 1.5573 - acc: 0.4449 - val_loss: 1.4866 - val_acc: 0.4762
Epoch 3/10
40000/40000 [==============================] - 211s 5ms/step - loss: 1.4143 - acc: 0.4927 - val_loss: 1.3949 - val_acc: 0.5068
Epoch 4/10
40000/40000 [==============================] - 211s 5ms/step - loss: 1.3176 - acc: 0.5305 - val_loss: 1.3100 - val_acc: 0.5325
Epoch 5/10
40000/40000 [==============================] - 211s 5ms/step - loss: 1.2406 - acc: 0.5557 - val_loss: 1.2676 - val_acc: 0.5540
Epoch 6/10
40000/40000 [==============================] - 211s 5ms/step - loss: 1.1767 - acc: 0.5809 - val_loss: 1.2955 - val_acc: 0.5385
Epoch 7/10
 4928/40000 [==>...........................] - ETA: 2:58 - loss: 1.1184 - acc: 0.5972
```

```
10000/10000 [==============================] - 8s 803us/step

Test acc: 56.05%
```

INCEPTION V1: DIMENSION REDUCTION

- Deep neural networks are computationally expensive. To make it cheaper, the authors limit the number of input channels by adding an extra 1x1 convolution before the 3x3 and 5x5 convolutions.

- Though adding an extra operation may seem counterintuitive, 1x1 convolutions are far cheaper than 5x5 convolutions, and the reduced number of input channels also help. Do note that however, the 1x1 convolution is introduced after the max pooling layer, rather than before.

- Model Architecture:

Model: "model_2"

Layer (type)	Output Shape	Param #	Connected to
input_5 (InputLayer)	(None, 32, 32, 3)	0	
conv2d_157 (Conv2D)	(None, 13, 13, 64)	9472	input_5[0][0]
conv2d_159 (Conv2D)	(None, 13, 13, 96)	6240	conv2d_157[0][0]
conv2d_161 (Conv2D)	(None, 13, 13, 16)	1040	conv2d_157[0][0]
max_pooling2d_39 (MaxPooling2D)	(None, 13, 13, 64)	0	conv2d_157[0][0]
conv2d_158 (Conv2D)	(None, 13, 13, 64)	4160	conv2d_157[0][0]
conv2d_160 (Conv2D)	(None, 13, 13, 128)	12416	conv2d_159[0][0]
conv2d_162 (Conv2D)	(None, 13, 13, 32)	12832	conv2d_161[0][0]
conv2d_163 (Conv2D)	(None, 13, 13, 32)	2080	max_pooling2d_39[0][0]
concatenate_25 (Concatenate)	(None, 13, 13, 256)	0	conv2d_158[0][0] conv2d_160[0][0] conv2d_162[0][0] conv2d_163[0][0]
conv2d_165 (Conv2D)	(None, 13, 13, 128)	32896	concatenate_25[0][0]
conv2d_167 (Conv2D)	(None, 13, 13, 32)	8224	concatenate_25[0][0]
max_pooling2d_40 (MaxPooling2D)	(None, 13, 13, 256)	0	concatenate_25[0][0]
conv2d_164 (Conv2D)	(None, 13, 13, 128)	32896	concatenate_25[0][0]
conv2d_166 (Conv2D)	(None, 13, 13, 192)	24768	conv2d_165[0][0]
conv2d_168 (Conv2D)	(None, 13, 13, 96)	76896	conv2d_167[0][0]
conv2d_169 (Conv2D)	(None, 13, 13, 64)	16448	max_pooling2d_40[0][0]
concatenate_26 (Concatenate)	(None, 13, 13, 480)	0	conv2d_164[0][0] conv2d_166[0][0] conv2d_168[0][0] conv2d_169[0][0]
max_pooling2d_41 (MaxPooling2D)	(None, 6, 6, 480)	0	concatenate_26[0][0]
conv2d_171 (Conv2D)	(None, 6, 6, 96)	46176	max_pooling2d_41[0][0]
conv2d_173 (Conv2D)	(None, 6, 6, 16)	7696	max_pooling2d_41[0][0]
max_pooling2d_42 (MaxPooling2D)	(None, 6, 6, 480)	0	max_pooling2d_41[0][0]
conv2d_170 (Conv2D)	(None, 6, 6, 192)	92352	max_pooling2d_41[0][0]
conv2d_172 (Conv2D)	(None, 6, 6, 208)	20176	conv2d_171[0][0]
conv2d_174 (Conv2D)	(None, 6, 6, 48)	19248	conv2d_173[0][0]
conv2d_175 (Conv2D)	(None, 6, 6, 64)	30784	max_pooling2d_42[0][0]
concatenate_27 (Concatenate)	(None, 6, 6, 512)	0	conv2d_170[0][0] conv2d_172[0][0] conv2d_174[0][0] conv2d_175[0][0]
conv2d_177 (Conv2D)	(None, 6, 6, 112)	57456	concatenate_27[0][0]
conv2d_179 (Conv2D)	(None, 6, 6, 24)	12312	concatenate_27[0][0]
max_pooling2d_43 (MaxPooling2D)	(None, 6, 6, 512)	0	concatenate_27[0][0]
conv2d_176 (Conv2D)	(None, 6, 6, 160)	82080	concatenate_27[0][0]
conv2d_178 (Conv2D)	(None, 6, 6, 224)	25312	conv2d_177[0][0]
conv2d_180 (Conv2D)	(None, 6, 6, 64)	38464	conv2d_179[0][0]
conv2d_181 (Conv2D)	(None, 6, 6, 64)	32832	max_pooling2d_43[0][0]
concatenate_28 (Concatenate)	(None, 6, 6, 512)	0	conv2d_176[0][0] conv2d_178[0][0]

conv2d_183 (Conv2D)	(None, 6, 6, 128)	65664	concatenate_28[0][0]
conv2d_185 (Conv2D)	(None, 6, 6, 24)	12312	concatenate_28[0][0]
max_pooling2d_44 (MaxPooling2D)	(None, 6, 6, 512)	0	concatenate_28[0][0]
conv2d_182 (Conv2D)	(None, 6, 6, 128)	65664	concatenate_28[0][0]
conv2d_184 (Conv2D)	(None, 6, 6, 256)	33024	conv2d_183[0][0]
conv2d_186 (Conv2D)	(None, 6, 6, 64)	38464	conv2d_185[0][0]
conv2d_187 (Conv2D)	(None, 6, 6, 64)	32832	max_pooling2d_44[0][0]
concatenate_29 (Concatenate)	(None, 6, 6, 512)	0	conv2d_182[0][0] conv2d_184[0][0] conv2d_186[0][0] conv2d_187[0][0]
conv2d_189 (Conv2D)	(None, 6, 6, 144)	73872	concatenate_29[0][0]
conv2d_191 (Conv2D)	(None, 6, 6, 32)	16416	concatenate_29[0][0]
max_pooling2d_45 (MaxPooling2D)	(None, 6, 6, 512)	0	concatenate_29[0][0]
conv2d_188 (Conv2D)	(None, 6, 6, 112)	57456	concatenate_29[0][0]
conv2d_190 (Conv2D)	(None, 6, 6, 288)	41760	conv2d_189[0][0]
conv2d_192 (Conv2D)	(None, 6, 6, 64)	51264	conv2d_191[0][0]
conv2d_193 (Conv2D)	(None, 6, 6, 64)	32832	max_pooling2d_45[0][0]
concatenate_30 (Concatenate)	(None, 6, 6, 528)	0	conv2d_188[0][0] conv2d_190[0][0] conv2d_192[0][0] conv2d_193[0][0]
conv2d_195 (Conv2D)	(None, 6, 6, 160)	84640	concatenate_30[0][0]
conv2d_197 (Conv2D)	(None, 6, 6, 32)	16928	concatenate_30[0][0]

➢

conv2d_195 (Conv2D)	(None, 6, 6, 160)	84640	concatenate_30[0][0]
conv2d_197 (Conv2D)	(None, 6, 6, 32)	16928	concatenate_30[0][0]
max_pooling2d_46 (MaxPooling2D)	(None, 6, 6, 528)	0	concatenate_30[0][0]
conv2d_194 (Conv2D)	(None, 6, 6, 256)	135424	concatenate_30[0][0]
conv2d_196 (Conv2D)	(None, 6, 6, 320)	51520	conv2d_195[0][0]
conv2d_198 (Conv2D)	(None, 6, 6, 128)	102528	conv2d_197[0][0]
conv2d_199 (Conv2D)	(None, 6, 6, 128)	67712	max_pooling2d_46[0][0]
concatenate_31 (Concatenate)	(None, 6, 6, 832)	0	conv2d_194[0][0] conv2d_196[0][0] conv2d_198[0][0] conv2d_199[0][0]
max_pooling2d_47 (MaxPooling2D)	(None, 2, 2, 832)	0	concatenate_31[0][0]
conv2d_201 (Conv2D)	(None, 2, 2, 160)	133280	max_pooling2d_47[0][0]
conv2d_203 (Conv2D)	(None, 2, 2, 32)	26656	max_pooling2d_47[0][0]
max_pooling2d_48 (MaxPooling2D)	(None, 2, 2, 832)	0	max_pooling2d_47[0][0]
conv2d_200 (Conv2D)	(None, 2, 2, 256)	213248	max_pooling2d_47[0][0]
conv2d_202 (Conv2D)	(None, 2, 2, 320)	51520	conv2d_201[0][0]
conv2d_204 (Conv2D)	(None, 2, 2, 128)	102528	conv2d_203[0][0]
conv2d_205 (Conv2D)	(None, 2, 2, 128)	106624	max_pooling2d_48[0][0]
concatenate_32 (Concatenate)	(None, 2, 2, 832)	0	conv2d_200[0][0] conv2d_202[0][0] conv2d_204[0][0] conv2d_205[0][0]
conv2d_207 (Conv2D)	(None, 2, 2, 192)	159936	concatenate_32[0][0]

➢

```
conv2d_209 (Conv2D)             (None, 2, 2, 48)     39984       concatenate_32[0][0]
__________________________________________________________________________________________________
max_pooling2d_49 (MaxPooling2D) (None, 2, 2, 832)    0           concatenate_32[0][0]
__________________________________________________________________________________________________
conv2d_206 (Conv2D)             (None, 2, 2, 384)    319872      concatenate_32[0][0]
__________________________________________________________________________________________________
conv2d_208 (Conv2D)             (None, 2, 2, 384)    74112       conv2d_207[0][0]
__________________________________________________________________________________________________
conv2d_210 (Conv2D)             (None, 2, 2, 128)    153728      conv2d_209[0][0]
__________________________________________________________________________________________________
conv2d_211 (Conv2D)             (None, 2, 2, 128)    106624      max_pooling2d_49[0][0]
__________________________________________________________________________________________________
concatenate_33 (Concatenate)    (None, 2, 2, 1024)   0           conv2d_206[0][0]
                                                                 conv2d_208[0][0]
                                                                 conv2d_210[0][0]
                                                                 conv2d_211[0][0]
__________________________________________________________________________________________________
average_pooling2d_11 (AveragePo (None, 1, 1, 1024)   0           concatenate_33[0][0]
__________________________________________________________________________________________________
average_pooling2d_9 (AveragePoo (None, 2, 2, 512)    0           concatenate_27[0][0]
__________________________________________________________________________________________________
average_pooling2d_10 (AveragePo (None, 2, 2, 528)    0           concatenate_30[0][0]
__________________________________________________________________________________________________
flatten_9 (Flatten)             (None, 1024)         0           average_pooling2d_11[0][0
__________________________________________________________________________________________________
flatten_7 (Flatten)             (None, 2048)         0           average_pooling2d_9[0][0]
__________________________________________________________________________________________________
flatten_8 (Flatten)             (None, 2112)         0           average_pooling2d_10[0][0
__________________________________________________________________________________________________
dropout_9 (Dropout)             (None, 1024)         0           flatten_9[0][0]
__________________________________________________________________________________________________
dropout_7 (Dropout)             (None, 2048)         0           flatten_7[0][0]
__________________________________________________________________________________________________
dropout_8 (Dropout)             (None, 2112)         0           flatten_8[0][0]
__________________________________________________________________________________________________
dense_9 (Dense)                 (None, 1000)         1025000     dropout_9[0][0]
__________________________________________________________________________________________________
dense_7 (Dense)                 (None, 1000)         2049000     dropout_7[0][0]
__________________________________________________________________________________________________
dense_8 (Dense)                 (None, 1000)         2113000     dropout_8[0][0]
__________________________________________________________________________________________________
loss3_classifier_act (Dense)    (None, 10)           10010       dense_9[0][0]
__________________________________________________________________________________________________
loss3_classifier_act (Dense)    (None, 10)           10010       dense_9[0][0]
__________________________________________________________________________________________________
loss1_classifier_act (Dense)    (None, 10)           10010       dense_7[0][0]
__________________________________________________________________________________________________
loss2_classifier_act (Dense)    (None, 10)           10010       dense_8[0][0]
==================================================================================================
Total params: 8,390,710
Trainable params: 8,390,710
Non-trainable params: 0
```

➢ Model Results:

```
# Evaluate the model on Test set

scores = googlenet.evaluate(x_test, {"loss3_classifier_act":y_test, "loss1_classifier_act":y_test, "loss2_classifier_act":y_test})
print("\nTest %s: %.2f%%" % (googlenet.metrics_names[1], scores[1]*100))
```

```
10000/10000 [==============================] - 7s 673us/step

Test loss3_classifier_act_loss: 94.44%
```

INCEPTION V2

➢ Model Architecture:

➢ Layer (type) Output Shape Param # Connected to
➢ ==========
==========
➢ input_6 (InputLayer) (None, 32, 32, 3) 0
➢ _____
_____
➢ conv2d_258 (Conv2D) (None, 15, 15, 32) 896 input_6[0][0]
➢ _____
_____
➢ conv2d_262 (Conv2D) (None, 15, 15, 128) 4224 conv2d_258[0][0]
➢ _____
_____
➢ conv2d_260 (Conv2D) (None, 15, 15, 128) 4224 conv2d_258[0][0]
➢ _____
_____
➢ conv2d_263 (Conv2D) (None, 15, 15, 128) 147584 conv2d_262[0][0]
➢ _____
_____
➢ max_pooling2d_30 (MaxPooling2D) (None, 15, 15, 32) 0 conv2d_258[0][0]
➢ _____
_____
➢ conv2d_259 (Conv2D) (None, 15, 15, 128) 4224 conv2d_258[0][0]
➢ _____
_____
➢ conv2d_261 (Conv2D) (None, 15, 15, 128) 147584 conv2d_260[0][0]
➢ _____
_____
➢ conv2d_264 (Conv2D) (None, 15, 15, 128) 147584 conv2d_263[0][0]
➢ _____
_____
➢ conv2d_265 (Conv2D) (None, 15, 15, 128) 4224 max_pooling2d_30[0][0]
➢ _____
_____
➢ concatenate_30 (Concatenate) (None, 15, 15, 512) 0 conv2d_259[0][0]
➢ conv2d_261[0][0]
➢ conv2d_264[0][0]
➢ conv2d_265[0][0]
➢ _____
_____
➢ conv2d_269 (Conv2D) (None, 15, 15, 128) 65664 concatenate_30[0][0]
➢ _____
_____
➢ conv2d_267 (Conv2D) (None, 15, 15, 128) 65664 concatenate_30[0][0]
➢ _____
_____

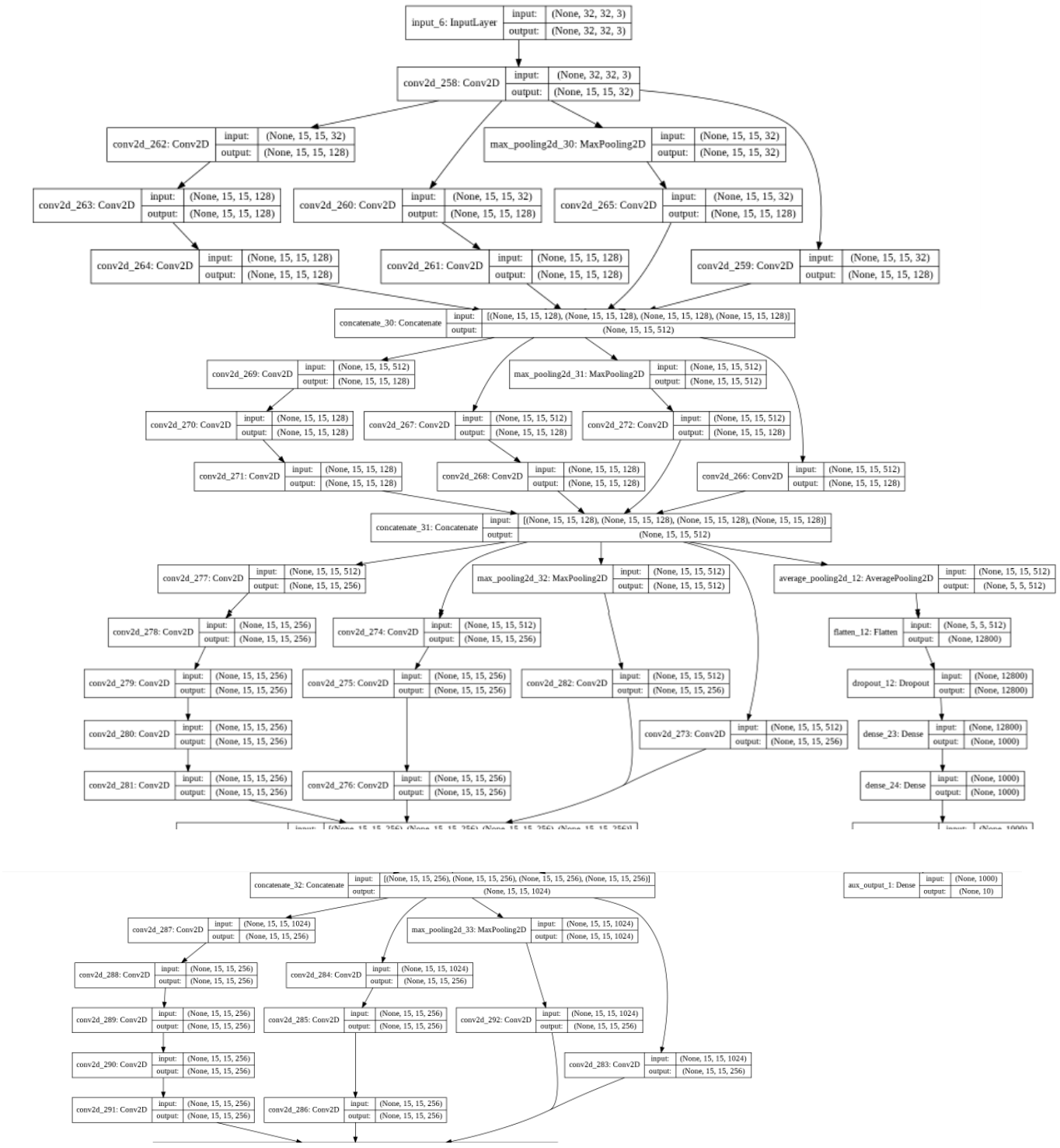
- conv2d_270 (Conv2D) (None, 15, 15, 128) 147584 conv2d_269[0][0]
- _____
- max_pooling2d_31 (MaxPooling2D) (None, 15, 15, 512) 0 concatenate_30[0][0]
- _____
- conv2d_266 (Conv2D) (None, 15, 15, 128) 65664 concatenate_30[0][0]
- _____
- conv2d_268 (Conv2D) (None, 15, 15, 128) 147584 conv2d_267[0][0]
- _____
- conv2d_271 (Conv2D) (None, 15, 15, 128) 147584 conv2d_270[0][0]
- _____
- conv2d_272 (Conv2D) (None, 15, 15, 128) 65664 max_pooling2d_31[0][0]
- _____
- concatenate_31 (Concatenate) (None, 15, 15, 512) 0 conv2d_266[0][0]
- conv2d_268[0][0]
- conv2d_271[0][0]
- conv2d_272[0][0]
- _____
- conv2d_277 (Conv2D) (None, 15, 15, 256) 131328 concatenate_31[0][0]
- _____
- conv2d_278 (Conv2D) (None, 15, 15, 256) 196864 conv2d_277[0][0]
- _____
- conv2d_274 (Conv2D) (None, 15, 15, 256) 131328 concatenate_31[0][0]
- _____
- conv2d_279 (Conv2D) (None, 15, 15, 256) 196864 conv2d_278[0][0]
- _____
- conv2d_275 (Conv2D) (None, 15, 15, 256) 196864 conv2d_274[0][0]
- _____
- conv2d_280 (Conv2D) (None, 15, 15, 256) 196864 conv2d_279[0][0]
- _____
- max_pooling2d_32 (MaxPooling2D) (None, 15, 15, 512) 0 concatenate_31[0][0]
- _____
- conv2d_273 (Conv2D) (None, 15, 15, 256) 131328 concatenate_31[0][0]
- _____
- conv2d_276 (Conv2D) (None, 15, 15, 256) 196864 conv2d_275[0][0]
- _____
- conv2d_281 (Conv2D) (None, 15, 15, 256) 196864 conv2d_280[0][0]

- ____
- conv2d_282 (Conv2D) (None, 15, 15, 256) 131328 max_pooling2d_32[0][0]
- ____
- concatenate_32 (Concatenate) (None, 15, 15, 1024) 0 conv2d_273[0][0]
- conv2d_276[0][0]
- conv2d_281[0][0]
- conv2d_282[0][0]
- ____
- conv2d_287 (Conv2D) (None, 15, 15, 256) 262400 concatenate_32[0][0]
- ____
- conv2d_288 (Conv2D) (None, 15, 15, 256) 196864 conv2d_287[0][0]
- ____
- conv2d_284 (Conv2D) (None, 15, 15, 256) 262400 concatenate_32[0][0]
- ____
- conv2d_289 (Conv2D) (None, 15, 15, 256) 196864 conv2d_288[0][0]
- ____
- conv2d_285 (Conv2D) (None, 15, 15, 256) 196864 conv2d_284[0][0]
- ____
- conv2d_290 (Conv2D) (None, 15, 15, 256) 196864 conv2d_289[0][0]
- ____
- max_pooling2d_33 (MaxPooling2D) (None, 15, 15, 1024) 0 concatenate_32[0][0]
- ____
- conv2d_283 (Conv2D) (None, 15, 15, 256) 262400 concatenate_32[0][0]
- ____
- conv2d_286 (Conv2D) (None, 15, 15, 256) 196864 conv2d_285[0][0]
- ____
- conv2d_291 (Conv2D) (None, 15, 15, 256) 196864 conv2d_290[0][0]
- ____
- conv2d_292 (Conv2D) (None, 15, 15, 256) 262400 max_pooling2d_33[0][0]
- ____
- concatenate_33 (Concatenate) (None, 15, 15, 1024) 0 conv2d_283[0][0]
- conv2d_286[0][0]
- conv2d_291[0][0]
- conv2d_292[0][0]
- ____
- conv2d_294 (Conv2D) (None, 15, 15, 512) 524800 concatenate_33[0][0]
- ____

- conv2d_295 (Conv2D) (None, 15, 15, 512) 2359808 conv2d_294[0][0]
- ____
- conv2d_298 (Conv2D) (None, 15, 15, 512) 524800 concatenate_33[0][0]
- ____
- max_pooling2d_34 (MaxPooling2D) (None, 15, 15, 1024) 0 concatenate_33[0][0]
- ____
- conv2d_293 (Conv2D) (None, 15, 15, 512) 524800 concatenate_33[0][0]
- ____
- conv2d_296 (Conv2D) (None, 15, 15, 512) 786944 conv2d_295[0][0]
- ____
- conv2d_297 (Conv2D) (None, 15, 15, 512) 786944 conv2d_295[0][0]
- ____
- conv2d_299 (Conv2D) (None, 15, 15, 512) 786944 conv2d_298[0][0]
- ____
- conv2d_300 (Conv2D) (None, 15, 15, 512) 786944 conv2d_298[0][0]
- ____
- conv2d_301 (Conv2D) (None, 15, 15, 512) 524800 max_pooling2d_34[0][0]
- ____
- concatenate_34 (Concatenate) (None, 15, 15, 3072) 0 conv2d_293[0][0]
- conv2d_296[0][0]
- conv2d_297[0][0]
- conv2d_299[0][0]
- conv2d_300[0][0]
- conv2d_301[0][0]
- ____
- conv2d_303 (Conv2D) (None, 15, 15, 512) 1573376 concatenate_34[0][0]
- ____
- conv2d_304 (Conv2D) (None, 15, 15, 512) 2359808 conv2d_303[0][0]
- ____
- conv2d_307 (Conv2D) (None, 15, 15, 512) 1573376 concatenate_34[0][0]
- ____
- max_pooling2d_35 (MaxPooling2D) (None, 15, 15, 3072) 0 concatenate_34[0][0]
- ____
- conv2d_302 (Conv2D) (None, 15, 15, 512) 1573376 concatenate_34[0][0]
- ____
- conv2d_305 (Conv2D) (None, 15, 15, 512) 786944 conv2d_304[0][0]
- ____

- conv2d_306 (Conv2D) (None, 15, 15, 512) 786944 conv2d_304[0][0]
- conv2d_308 (Conv2D) (None, 15, 15, 512) 786944 conv2d_307[0][0]
- conv2d_309 (Conv2D) (None, 15, 15, 512) 786944 conv2d_307[0][0]
- conv2d_310 (Conv2D) (None, 15, 15, 512) 1573376 max_pooling2d_35[0][0]
- concatenate_35 (Concatenate) (None, 15, 15, 3072) 0 conv2d_302[0][0]
  - conv2d_305[0][0]
  - conv2d_306[0][0]
  - conv2d_308[0][0]
  - conv2d_309[0][0]
  - conv2d_310[0][0]
- average_pooling2d_12 (AveragePo (None, 5, 5, 512) 0 concatenate_31[0][0]
- average_pooling2d_13 (AveragePo (None, 5, 5, 1024) 0 concatenate_33[0][0]
- average_pooling2d_14 (AveragePo (None, 5, 5, 3072) 0 concatenate_35[0][0]
- flatten_12 (Flatten) (None, 12800) 0 average_pooling2d_12[0][0]
- flatten_13 (Flatten) (None, 25600) 0 average_pooling2d_13[0][0]
- flatten_14 (Flatten) (None, 76800) 0 average_pooling2d_14[0][0]
- dropout_12 (Dropout) (None, 12800) 0 flatten_12[0][0]
- dropout_13 (Dropout) (None, 25600) 0 flatten_13[0][0]
- dropout_14 (Dropout) (None, 76800) 0 flatten_14[0][0]
- dense_23 (Dense) (None, 1000) 12801000 dropout_12[0][0]
- dense_25 (Dense) (None, 1000) 25601000 dropout_13[0][0]

- dense_27 (Dense) (None, 1000) 76801000 dropout_14[0][0]
- _____
  _____
- dense_24 (Dense) (None, 1000) 1001000 dense_23[0][0]
- _____
  _____
- dense_26 (Dense) (None, 1000) 1001000 dense_25[0][0]
- _____
  _____
- output (Dense) (None, 10) 10010 dense_27[0][0]
- _____
  _____
- aux_output_1 (Dense) (None, 10) 10010 dense_24[0][0]
- _____
  _____
- aux_output_2 (Dense) (None, 10) 10010 dense_26[0][0]
- ==========
  ==========
- Total params: 141,746,134
- Trainable params: 141,746,134
- Non-trainable params: 0
- _____
  _____


input_6: InputLayer | input: (None, 32, 32, 3) | output: (None, 32, 32, 3)
conv2d_258: Conv2D | input: (None, 32, 32, 3) | output: (None, 15, 15, 32)
conv2d_262: Conv2D | input: (None, 15, 15, 32) | output: (None, 15, 15, 128)
max_pooling2d_30: MaxPooling2D | input: (None, 15, 15, 32) | output: (None, 15, 15, 32)
conv2d_263: Conv2D | input: (None, 15, 15, 128) | output: (None, 15, 15, 128)
conv2d_260: Conv2D | input: (None, 15, 15, 32) | output: (None, 15, 15, 128)
conv2d_265: Conv2D | input: (None, 15, 15, 32) | output: (None, 15, 15, 128)
conv2d_264: Conv2D | input: (None, 15, 15, 128) | output: (None, 15, 15, 128)
conv2d_261: Conv2D | input: (None, 15, 15, 128) | output: (None, 15, 15, 128)
conv2d_259: Conv2D | input: (None, 15, 15, 32) | output: (None, 15, 15, 128)
concatenate_30: Concatenate | input: [(None, 15, 15, 128), (None, 15, 15, 128), (None, 15, 15, 128), (None, 15, 15, 128)] | output: (None, 15, 15, 512)
conv2d_269: Conv2D | input: (None, 15, 15, 512) | output: (None, 15, 15, 128)
max_pooling2d_31: MaxPooling2D | input: (None, 15, 15, 512) | output: (None, 15, 15, 512)
conv2d_270: Conv2D | input: (None, 15, 15, 128) | output: (None, 15, 15, 128)
conv2d_267: Conv2D | input: (None, 15, 15, 512) | output: (None, 15, 15, 128)
conv2d_272: Conv2D | input: (None, 15, 15, 512) | output: (None, 15, 15, 128)
conv2d_271: Conv2D | input: (None, 15, 15, 128) | output: (None, 15, 15, 128)
conv2d_268: Conv2D | input: (None, 15, 15, 128) | output: (None, 15, 15, 128)
conv2d_266: Conv2D | input: (None, 15, 15, 512) | output: (None, 15, 15, 128)
concatenate_31: Concatenate | input: [(None, 15, 15, 128), (None, 15, 15, 128), (None, 15, 15, 128), (None, 15, 15, 128)] | output: (None, 15, 15, 512)
conv2d_277: Conv2D | input: (None, 15, 15, 512) | output: (None, 15, 15, 256)
max_pooling2d_32: MaxPooling2D | input: (None, 15, 15, 512) | output: (None, 15, 15, 512)
average_pooling2d_12: AveragePooling2D | input: (None, 15, 15, 512) | output: (None, 5, 5, 512)
conv2d_278: Conv2D | input: (None, 15, 15, 256) | output: (None, 15, 15, 256)
conv2d_274: Conv2D | input: (None, 15, 15, 512) | output: (None, 15, 15, 256)
flatten_12: Flatten | input: (None, 5, 5, 512) | output: (None, 12800)
conv2d_279: Conv2D | input: (None, 15, 15, 256) | output: (None, 15, 15, 256)
conv2d_275: Conv2D | input: (None, 15, 15, 256) | output: (None, 15, 15, 256)
conv2d_282: Conv2D | input: (None, 15, 15, 512) | output: (None, 15, 15, 256)
dropout_12: Dropout | input: (None, 12800) | output: (None, 12800)
conv2d_280: Conv2D | input: (None, 15, 15, 256) | output: (None, 15, 15, 256)
conv2d_273: Conv2D | input: (None, 15, 15, 512) | output: (None, 15, 15, 256)
dense_23: Dense | input: (None, 12800) | output: (None, 1000)
conv2d_281: Conv2D | input: (None, 15, 15, 256) | output: (None, 15, 15, 256)
conv2d_276: Conv2D | input: (None, 15, 15, 256) | output: (None, 15, 15, 256)
dense_24: Dense | input: (None, 1000) | output: (None, 1000)
concatenate_32: Concatenate | input: [(None, 15, 15, 256), (None, 15, 15, 256), (None, 15, 15, 256), (None, 15, 15, 256)] | output: (None, 15, 15, 1024)
aux_output_1: Dense | input: (None, 1000) | output: (None, 10)
conv2d_287: Conv2D | input: (None, 15, 15, 1024) | output: (None, 15, 15, 256)
max_pooling2d_33: MaxPooling2D | input: (None, 15, 15, 1024) | output: (None, 15, 15, 1024)
conv2d_288: Conv2D | input: (None, 15, 15, 256) | output: (None, 15, 15, 256)
conv2d_284: Conv2D | input: (None, 15, 15, 1024) | output: (None, 15, 15, 256)
conv2d_289: Conv2D | input: (None, 15, 15, 256) | output: (None, 15, 15, 256)
conv2d_285: Conv2D | input: (None, 15, 15, 256) | output: (None, 15, 15, 256)
conv2d_292: Conv2D | input: (None, 15, 15, 1024) | output: (None, 15, 15, 256)
conv2d_290: Conv2D | input: (None, 15, 15, 256) | output: (None, 15, 15, 256)
conv2d_283: Conv2D | input: (None, 15, 15, 1024) | output: (None, 15, 15, 256)
conv2d_291: Conv2D | input: (None, 15, 15, 256) | output: (None, 15, 15, 256)
conv2d_286: Conv2D | input: (None, 15, 15, 256) | output: (None, 15, 15, 256)

➢ Model Results:

```
[ ]  # Evaluate the model on Test set

     scores = inception_V3.evaluate(x_test, {"output":y_test, "aux_output_1":y_test, "aux_output_2":y_test})
     print("\nTest %s: %.2f%%" % (inception_V3.metrics_names[1], scores[1]*100))
```

```
10000/10000 [==============================] - 102s 10ms/step

Test output_loss: 71.70%
```

THANK YOU