1)what are the advantage of a CNN over a fully connected DNN for image classification ?

Ans : CNN will outperform a fully-connected network if they have same number of hidden layers with same structure.Convolutional Neural Networks perform better than fully-connected networks on binary image classification, with a lot less parameters, because of their shared-weights architecture and translation invariance characteristics.

2) consider a CNN composed of three convolution layer each with 3x3 kernels a stride of 2 and same padding.the lowest layer outputs 100 features maps, the middle one outputs 200 and the top one output 400 the input image are RGB images 200x300 pixels.

Ans :

parameters:

first convolutional layer kernel-size and RGB channels, plus bias: 3 \* 3 \* 3 + 1 = 28 output feature maps is 100: 28 \* 100 = 2800

second convolutional layer kernel-size and last feature maps, plus bias: 3 \* 3 \* 100 + 1 = 901 output feature maps is 200: 901 \* 200 = 180200

third convolutional layers kernel-size and last feautre maps, plus bias: 3 \* 3 \* 200 + 1 =1801 output feautre maps is 400: 1801 \* 400 = 720400

Total parameters is 2800 + 180200 + 720400 = 903400

memories since 32-bit is 4 bytes

first convolutional layer one feature map size: 100 \* 150 = 15000 total output: 15000 \* 100 = 1,500,000

second convolutional layer one feature map size: 50 \* 75 = 3,750 total output: 3750 \* 200 = 750,000

third convolutional layer one feature map size: 25 \* 38 = 950 total ouput: 950 \* 400 = 380, 000

(1,500,000 + 750,000 + 380,000) \* 4 / 1024 /1024 = 10.032 (MB) 903400 \* 4 / 1024 / 1024 = 3.44 (MB) 10.032+ 3.44=13.47(MB)

3)if your GPU runs out of memory while training a CNN what are five things you could try to solve the problems ?

Ans : Reduce the mini-batch size.

Reduce dimensionality using a larger stride in one or more layers.

Remove one or more layers.

Use 16-bit floats instead of 32-bit floats.

Distribute the CNN across multiple devices.

4) why would you want to add a max pooling layer rather than a Convolution layer with the same stride ?

Ans : A max pooling layer has no parameters at all, whereas a convolutional layer has a lot.

5) why would you want to add a local response normalization layer ?

Ans : This form of normalization makes the neurons that most strongly activate inhibit neurons at the same location but in neighboring feature.This encourages different feature maps to specialize, pushing them apart and forcing them to explore a wider range of features, ultimately improving generalization.

It is typically used in the lower layers to have a larger pool of low-level features that the upper layers can build upon.

6) can you name the main innovation in Alexnet,compared to LeNet5 ? What about the main innovation in googLeNet,ResNet,SEnet and xception ?

Ans :

The main innovations in AlexNet compared to LeNet-5 are (1) it is much larger and deeper, and (2) it stacks convolutional layers directly on top of each other, instead of stacking a pooling layer on top of each convolutional layer.

The main innovation in GoogLeNet is the introduction of inception modules, which make it possible to have a much deeper net than previous CNN architectures, with fewer parameters.

Finally, ResNet's main innovation is the introduction of skip connections, which make it possible to go well beyond 100 layers. Arguably, its simplicity and consistency are also rather innovative.

7) what is a fully Convolutional network ? How can you convert a dense layer into a convolutional layer?

Ans :

FCN is a network that does not contain any “Dense” layers (as in traditional CNNs) instead it contains 1x1 convolutions that perform the task of fully connected layers (Dense layers).

A fully convolution network can be built by simply replacing the FC layers with there equivalent Conv layers. In the example of VGG16 we can do so by first removing the last four layers. One way to do so is to pop layers from the model. In the model stack, each popping will remove the last layer.

8) what is the main Technical Difficulty of semantic segmentation ?

Ans : Semantic Segmentation follows three steps:

Classifying: Classifying a certain object in the image.

Localizing: Finding the object and drawing a bounding box around it.

Segmentation: Grouping the pixels in a localized image by creating a segmentation mask.

the task of Semantic Segmentation can be referred to as classifying a certain class of image and separating it from the rest of the image classes by overlaying it with a segmentation mask.It can also be thought of as the classification of images at a pixel level.The goal is simply to take an image and generate an output such that it contains a segmentation map where the pixel value (from 0 to 255) of the input image is transformed into a class label value (0, 1, 2, … n).

9) build your own CNN from scratch and try to achieve the highest possible accuracy of MNIST.

Ans : Build your own CNN and try to achieve the highest accuracy (PYTHON PROGRAMMING)

Build a CNN for for image classification task on MNIST dataset. In your work, please set below random seed as the last two digits of your student id:

tf.random.set\_seed(84)

np.random.seed(84)

Task 1 (1%) Load MNIST dataset with Keras. Split it into training/validation/test set (Training set size: 55000 images, validation set size: 5000 images, test set size:10000 images).

Task 2 (2%) Build a CNN model with below structure:

Layer (type)

conv2d (Conv2D)

conv2d 1 (Conv2D)

max\_pooling2d (MaxPooling2D

)

flatten (Flatten)

dropout (Dropout)

dense (Dense

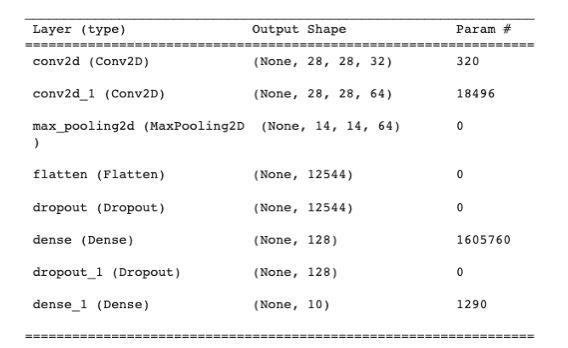
Train your model and evaluate it on test set with model.evaluate(). What's the accuracy on test set? Please visualize several examples of correctly classified and mis-classified and explain what you can observe from these results.

Task 3 (3%) improve your work via below strategies:

1. add image augmentation

2. batch norm

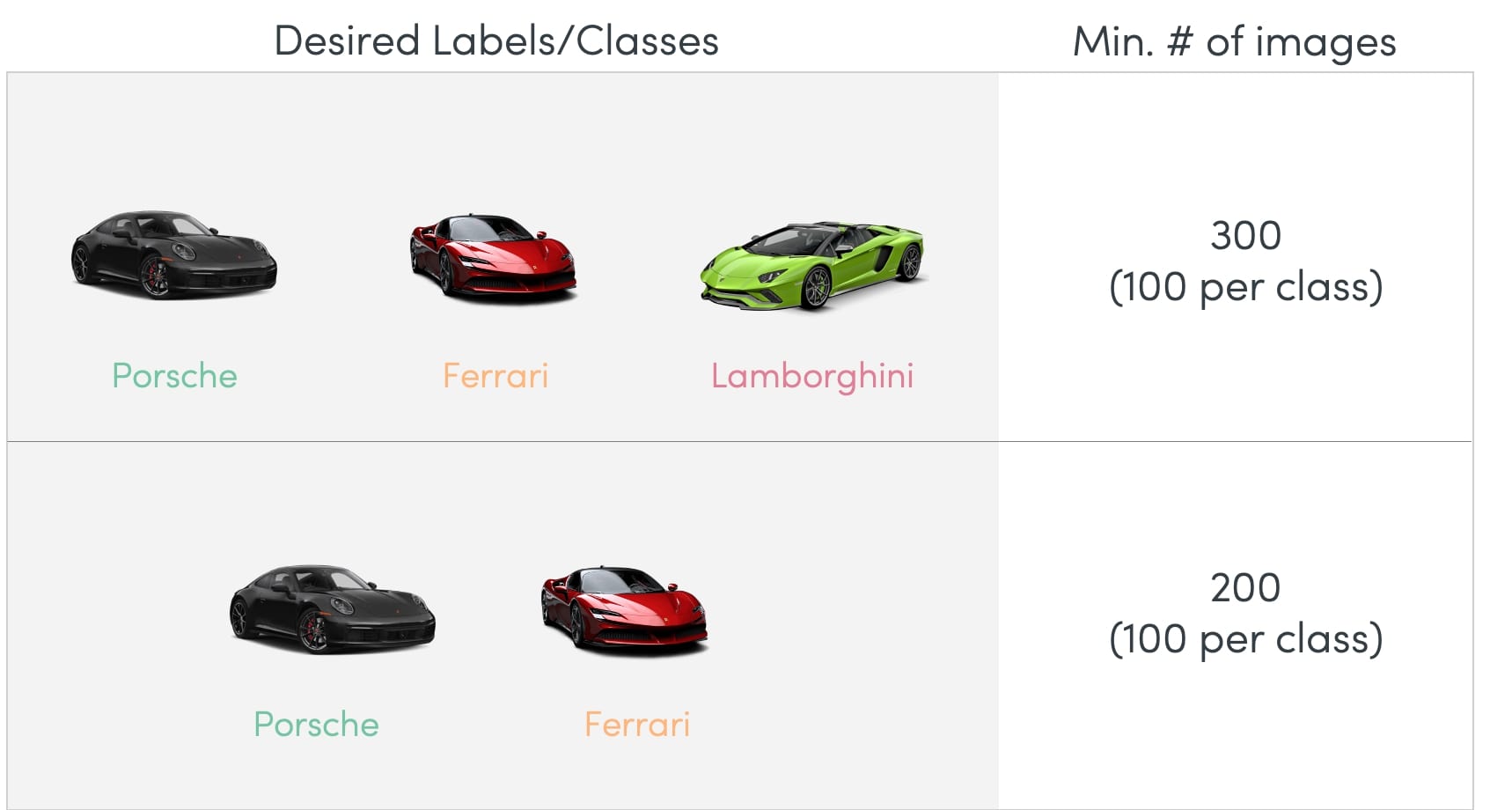
3. 2-fold cross-validation to select model.



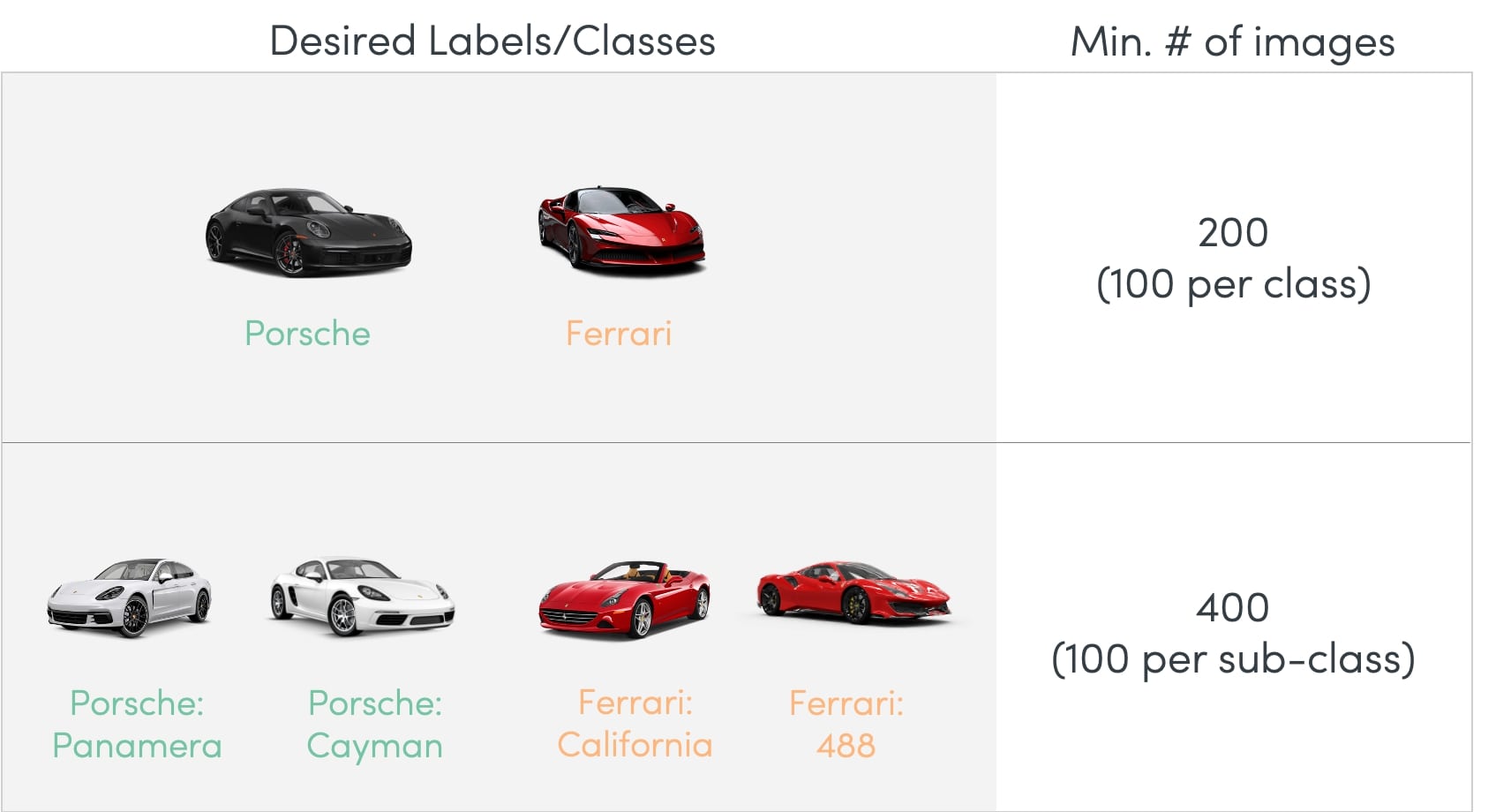
10) use transfer learning for large image classification going through these step.

a.create a training set containing at least 100 image per class for ex.you could classify your own pictures based on the locations or Alternatively you can use an Existing dataset.

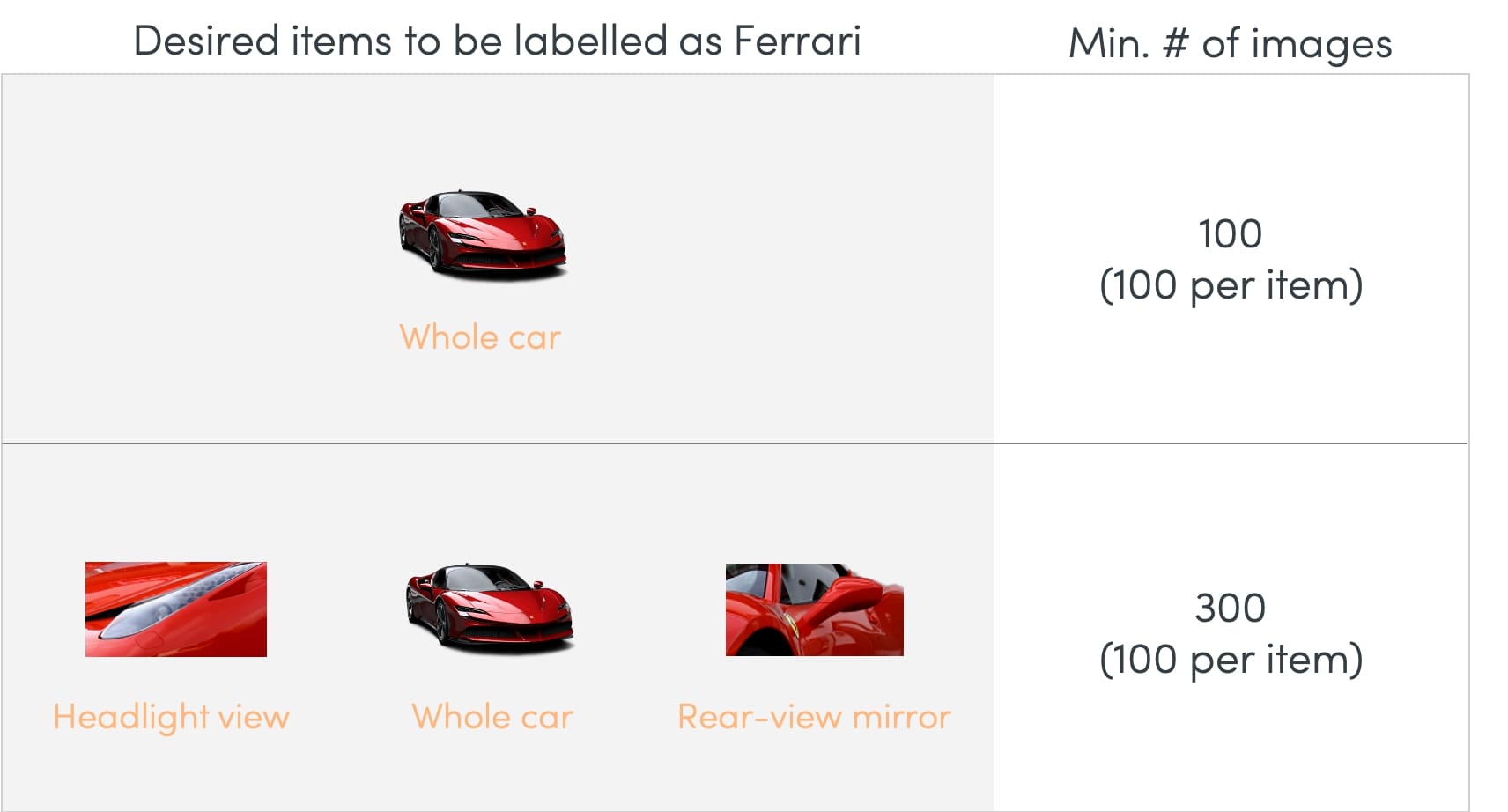
Ans : A rule of thumb on our platform is to have a minimum number of 100 images per each class you want to detect. In many cases, however, more data per class is required to achieve high-performing systems. If you seek to classify a higher number of labels, then you must adjust your image dataset accordingly

.

If you’re aiming for greater granularity within a class, then you need a higher number of pictures. You need to ensure meeting the threshold of at least 100 images for each added sub-label.



The more items (e.g. headlight view, the whole car, rearview, ...) you want to fit into a class, the higher the number of images you need to ensure your model performs optimally. Again, a healthy benchmark would be a minimum of 100 images per each item that you intend to fit into a label.



it's important to remember that 100 images per class are just a rule of thumb that suggests a minimum amount of images for your dataset. Depending on your use case, you might need more.

b.split it into a training set a validation set and a test set.

Ans : use the train\_test\_split to first make the split on the original dataset. Then, to get the validation set, we can apply the same function to the train set to get the validation set. In the function below, the test set size is the ratio of the original data we want to use as the test set

The main idea of splitting the dataset into a validation set is to prevent our model from overfitting , the model becomes really good at classifying the samples in the training set but cannot generalize and make accurate classifications on the data it has not seen before.

C.build the input pipeline Including the appropriate Preprocessing operation and optionally add data augmentation.

Ans : The input pipeline is a quick and easy utility provided in tf. data api to make complex input pipelines from simple and reusable codes and all in few lines of code.Data augmentation is a process of artificially increasing the amount of data by generating new data points from existing data.

D.fine tune a pretrained model on this dataset.

Ans : model as a feature extraction mechanism.

Use the Architecture of the pre-trained model – What we can do is that we use architecture of the model while we initialize all the weights randomly and train the model according to our dataset again.

Step 1: Understand what tuning machine learning model is.

Step 2: Cover The Basics.

Step 3: Find Your Score Metric.

Obtain Accurate Forecasting Score.

Step 5: Diagnose Best Parameter Value Using Validation Curves.

Step 6: Use Grid Search To Optimize Hyperparameter Combination.