You are provided with 2 Jupyter notebook created by Francois Chollet: 5.2-using-convnets-with-small-dateset.ipynb and 5.3-using-a-pretrained-convnet.ipynb. Please save an unmodified copy of each of those Jupyter notebooks so that you could compare the original results with your own.

Before you start running your Jupyter notebooks, please install Python packages: pillow and h5py.

Please note that this time you are given “original” notebooks. You might have to tweak them here and there to make accommodation to the latest version of TensorFlow (Keras) you are running.

**Problem 1.** Examine the CNN for the basic binary classification on Kaggle.com dog\_vs\_cats dataset, as provided in the Jupyter Notebook 5.2-using-convnets-with-small-dateset.ipynb. That CNN has 4 convolutional layers, 4 max\_pooling layers, 1 flatten layer and 2 dense layers. Keep original cells with code and results unmodified so that you could use them for comparison. Add new cells for modified definition and training of your model. Add an L1 regularizer on next to the last Dense layer. That is the Dense layer with (512,1) output tensor. As the regularization lambda, use the value of 0.0001. Run your model for 15 epochs and compare it with the original result on the model without regularization. Present the plots of training and validation accuracies and let us know whether regularization alleviate the overfitting and to what extent. Save your model as an H5 file. Keep H5 file for future reference. Do not submit H5 file as part of your solution.

**(25%)**

**Problem 2.** Demonstrate image augmentation for an image of your choice. Produce one modified image for every one of these options separately: rotation\_range, width\_shift, shear\_range, zoom\_range, vertical\_flip, and horizontal\_fit. Reference notebooks generate images which have random transformations along all of those option axes. You are asked to perform one transformation at a time.

**(25%)**

**Problem 3.** In the first portion of Jupyter notebook 5.3-using-a-pretrained-convnet.ipynb the author applied Transfer Learning or Feature Extraction technique by collecting the output of the convolutional base and then training simple Dense classifier. A reasonable question is whether we could add a convolutional layer at the bottom of the dense classifier and whether that would improve the accuracy of the model. Please keep the original code and original results unmodified (cells 6 and 7) for comparison with your new results. Add new cells to the notebook with a trainable layer which besides the dense top has at least one convolutional layer. New convolutional layer is added before the Flatten layer. You are asked to increase the number of convolutional layers. Please add proper padding to that new convolutional layer to avoid shrinking its output. Report on the number of trainable parameters in your model. Report on the results of your model as compared with that of Chollet.

**(25%)**

**Problem 4**. In the last portion of Jupyter notebook 5.3-using-a-pretrained-convnet.ipynb we have results for a fine-tuned VGG16 network. Chollet fine-tuned the last 3 conv layers: block5\_conv1, block5\_conv2 and block5\_conv3. Examine whether fine-tuning layerblock4\_conv3 only and freezing layers in block 5: block5\_conv1, block5\_conv2 and block5\_conv3 will result in significant change for better or worse in the training accuracy. Report the number of trainable parameters when Chollet was fine tuning the last 3 convolutional layers in block5 vs. when you are tuning only one layer block4\_conv3. Please leave the original code and results in the notebook unmodified so that you can easily make the comparison. You might have to search through Keras documentation on how to freeze and unfreeze particular convolutional layers.

**(25%)**

Please keep your saved models for your future use. Please do not upload them as parts of your solution.

Please add your name to the top of your notebooks.

We expect you to submit at least two notebooks (\*.ipynb files) along with their HTML images.

Some problems require GPU processing. Unless you have a decent GPU card yourself, do those problems in Google Colab.

If your notebook(s) contain(s) excessively long outputs please copy a meaningful and illustrative number of initial and/or final lines and paste those in a markdown (comment) cell. Then, delete the long output(s).