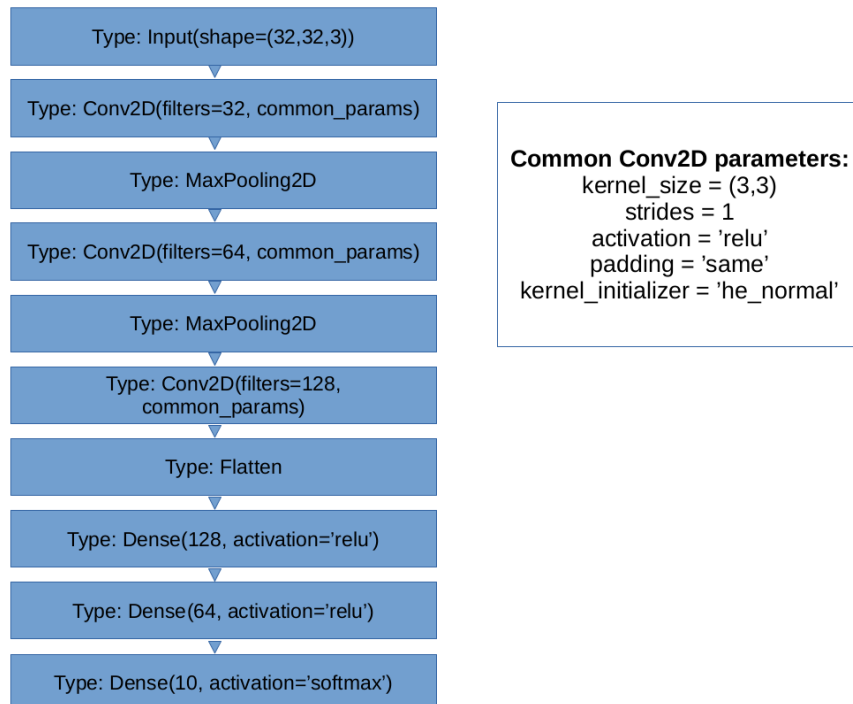


ICAT3260 Machine Learning for Computer Vision
Assignment 1 (23.10.2023) max. 10 points

- 1) Fetch the package CIFAR10-mini.zip from Moodle and inspect its contents. What kinds of files and how many of them do the folders 'training' and 'testing' contain? (0.5p)
- 2) Create a function in Python that reads the images of folder 'training' or 'testing' and returns the images in a numpy `ndarray` of size `<image_count, image_rows, image_cols, image_channels>`. For image reading you can use the PIL library's `Image` class. (2p)
- 3) Create another function in Python that returns a numpy `ndarray` of size `<image_count, 1>` containing the label of each image. The image labels are in the `.csv` file within the `training` and `testing` folders such that the first line in the file indicates the label of `image_0000.png`, the second line the label of `image_0001.png`, etc. (2p)
- 4) Create a function that normalizes the color channel (pixel) values of each image in the training set and test set to the range `[-1, 1[` (0.5p)
- 5) Create a function that takes the test data set and an integer value (such as '3000') and returns the data as two arrays that have been split from the input array at the given point. For example – input: `array1(10000,1)` → output: `array2(3000,1)`, `array3(7000,1)`. Use this function to split the test data (from item 2 above) into *test* and *validation* sets (0.5p)
- 6) Create a function to perform *one-hot* encoding of the training, validation and test labels. Hint: use the `keras.utils` function `to_categorical` (0.5p)
- 7) Using TensorFlow Keras
 - Create a basic neural network model that has the following structure: (2p)



- Compile the model with *categorical cross-entropy* loss, *accuracy* metric, and *Adam* optimizer with learning rate $3e-4$ (0.5p)
- 8) Fit the model (`model.fit`) on the training data using one-hot training data labels for 12 epochs. Also provide the validation data and one-hot validation data labels to the fitter.

- If everything is correct, the validation accuracy should be around 60% after 12 epochs (0.5p)
- 9) Use `model.evaluate` to test the trained model on the test data set and one-hot test labels. What is the achieved test accuracy and how does it compare to the model's validation accuracy? (1p)