Implementing Convolutional Neural Network by Keras

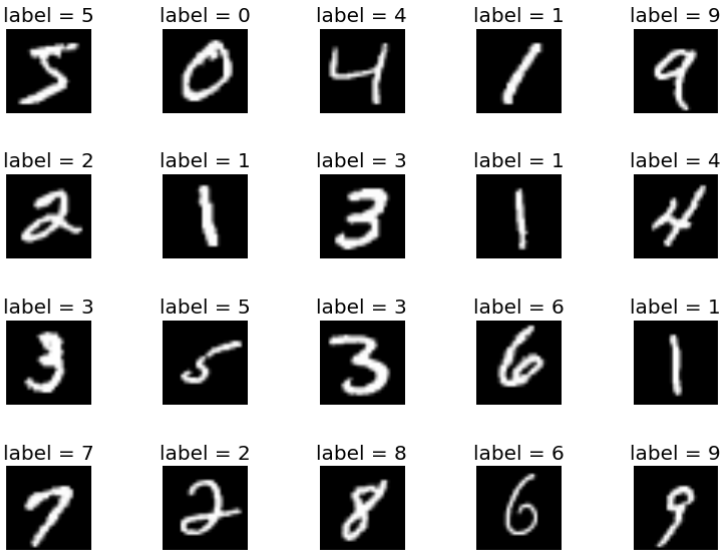
Kietikul Jearanaitanakij

Department of Computer Engineering, KMITL

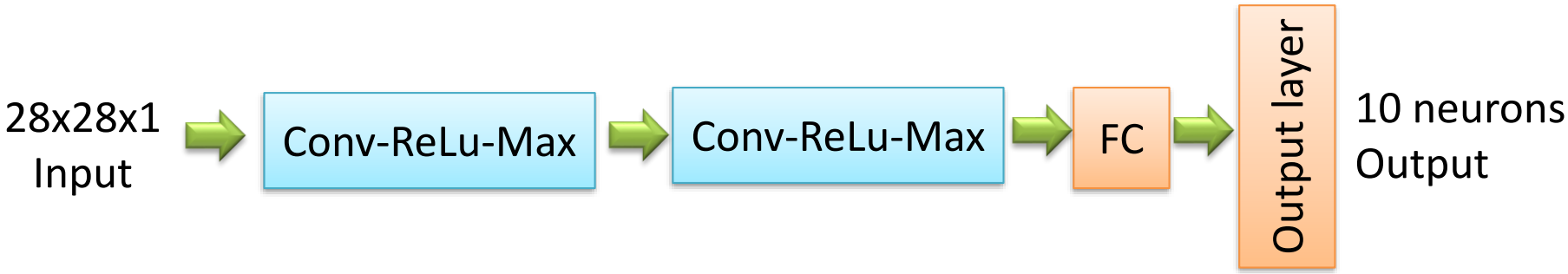
Example1: Predicting the MNIST digits

- MNIST : National Institute of Standards and Technology
- MNIST : a large dataset of handwritten digits that is commonly used for training various image processing systems.

- Image size 28 x 28 pixels
- Image channel = 1
- Digit ranges from 0 to 9
- Training : 55,000 instances
- Validation: 5,000 instances
- Test : 10,000 instances

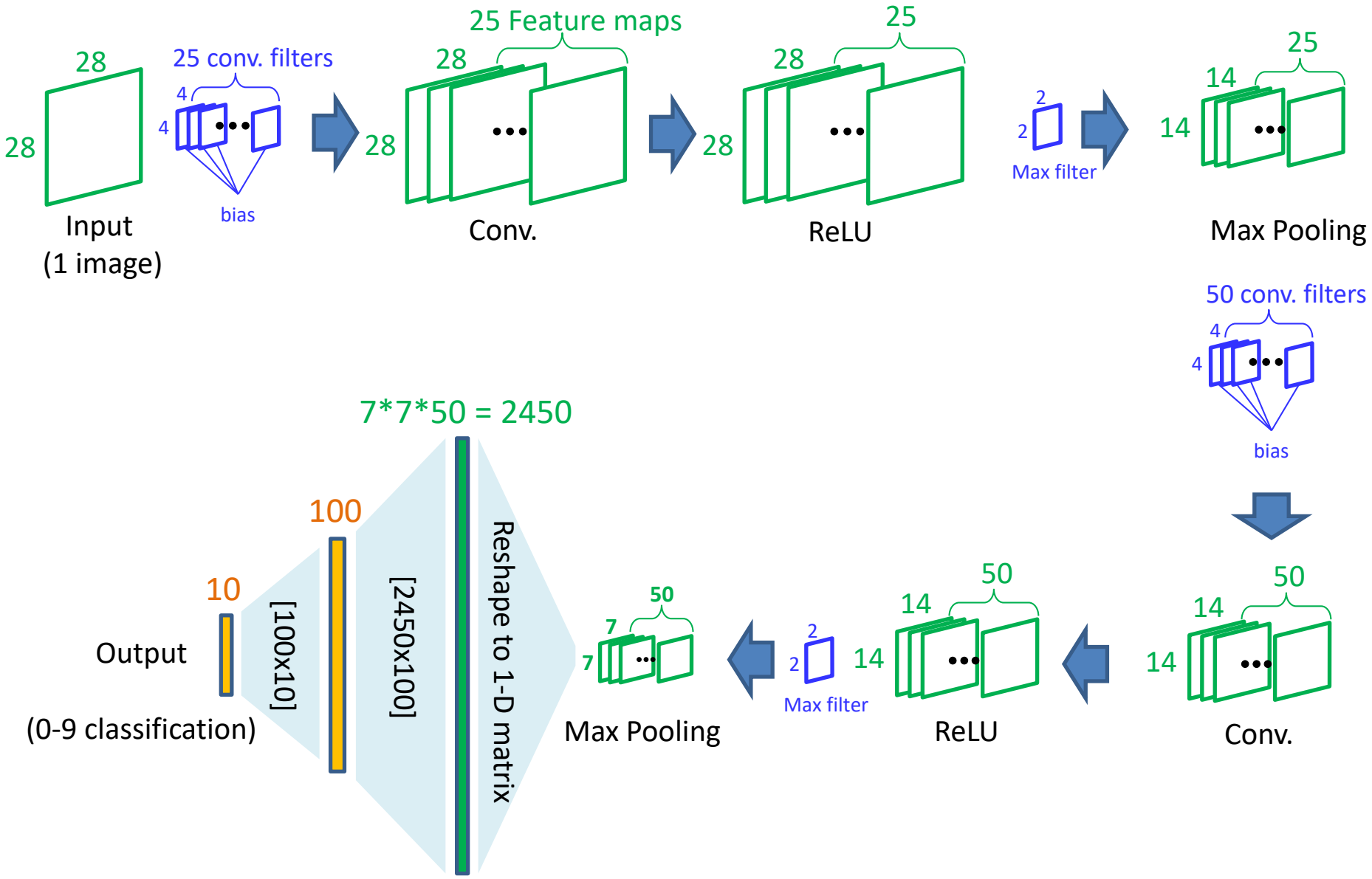


- We will develop a conv net to predict the MNIST digits.



- After we load the data, we will
  - Setup our model variables,
  - Create the model,
  - Train the model in batches,
  - Visualize loss, accuracy, and some sample digits.

CNN architecture for MNIST

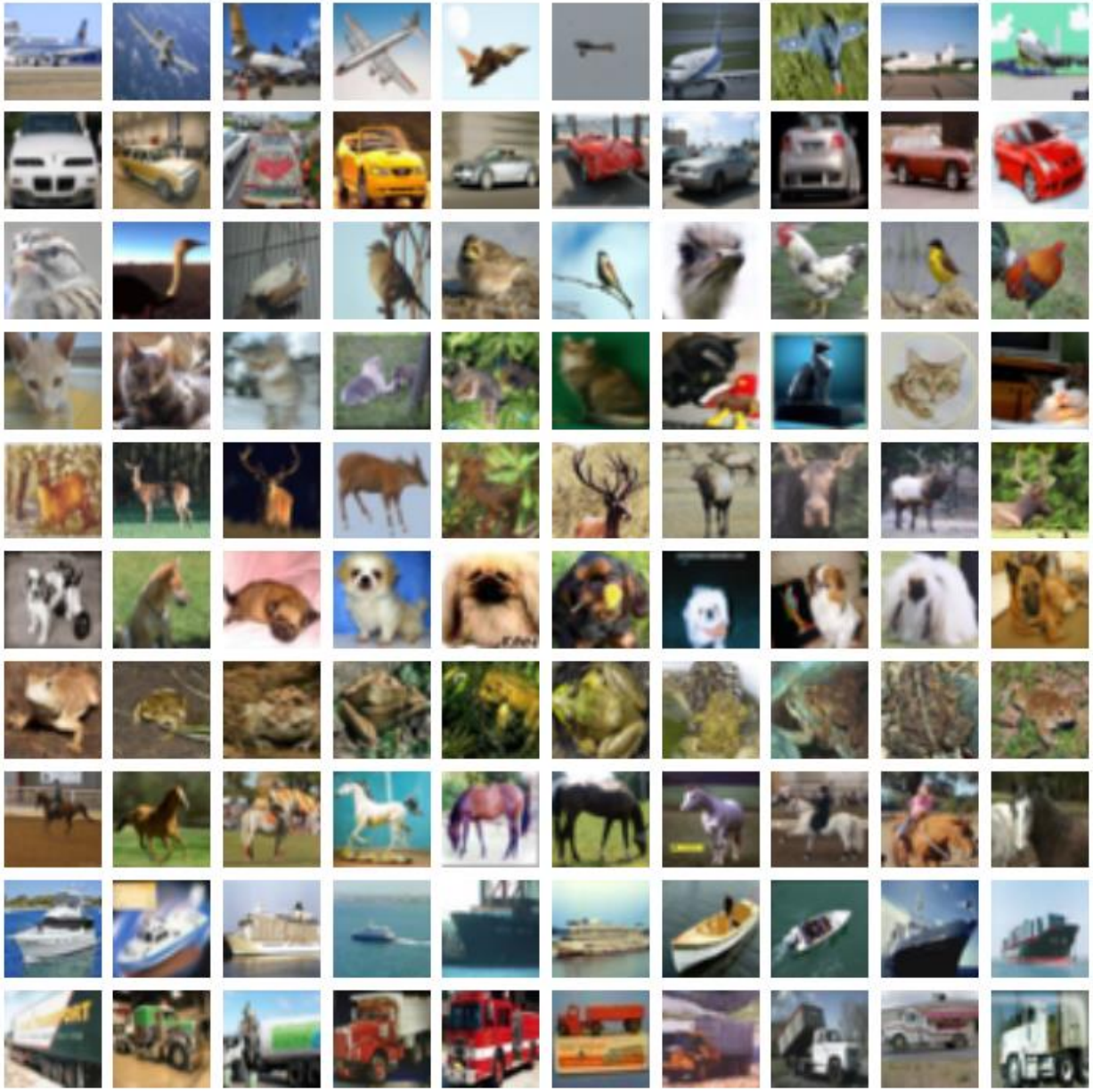


https://colab.research.google.com/drive/10DqpPl5Ag9UBtxpLMBdiBSz0149XlfFo

Example 2: CIFAR-10 Dataset

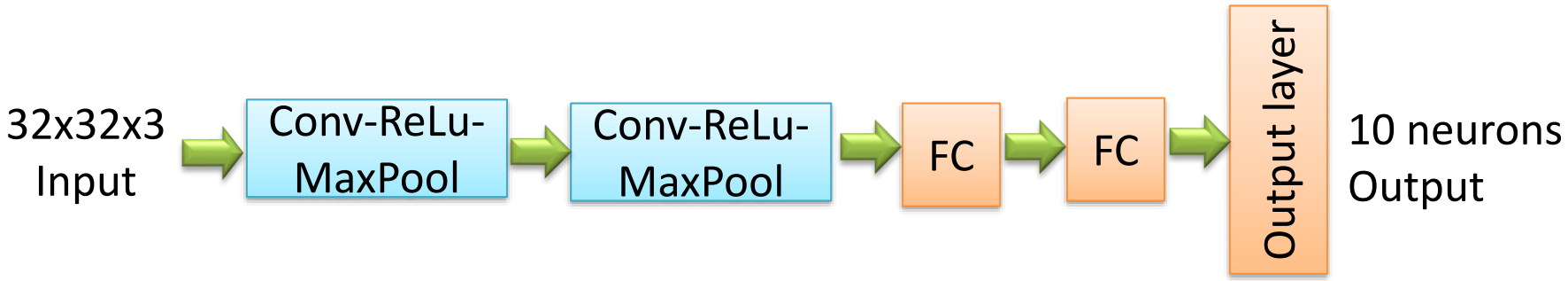
- We will implement a more advanced method of reading image data and use a larger CNN to do image recognition on the CIFAR10 dataset (https://www.cs.toronto.edu/~kriz/cifar.html).
- This dataset has 60,000 of 32x32 color images that fall into exactly one of ten possible classes.
- 50,000 training images and 10,000 test images.
- The potential classes for the images are airplane, automobile, bird, cat, deer, dog, frog, horse, ship, and truck.

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



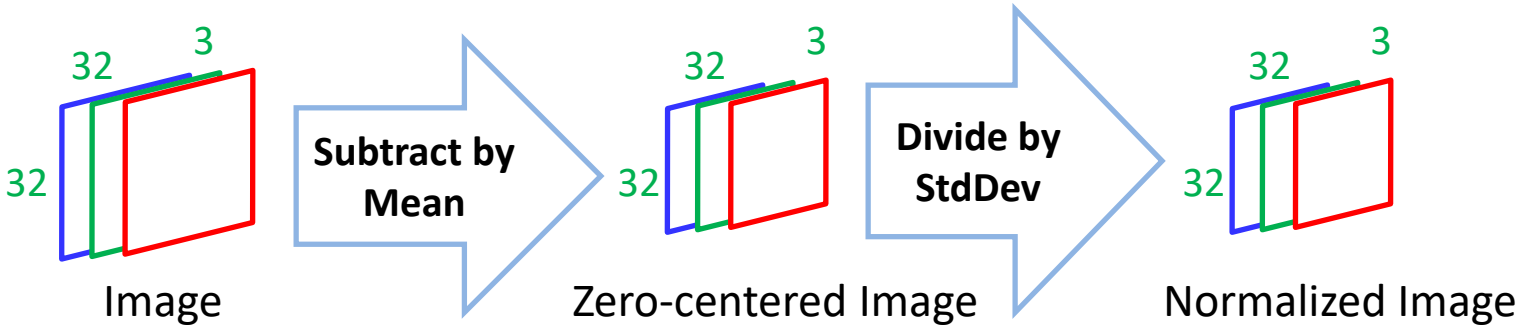
airplane	Label 0
automobile	Label 1
bird	Label 2
cat	Label 3
deer	Label 4
dog	Label 5
frog	Label 6
horse	Label 7
ship	Label 8
truck	Label 9

CNN architecture for CIFAR-10



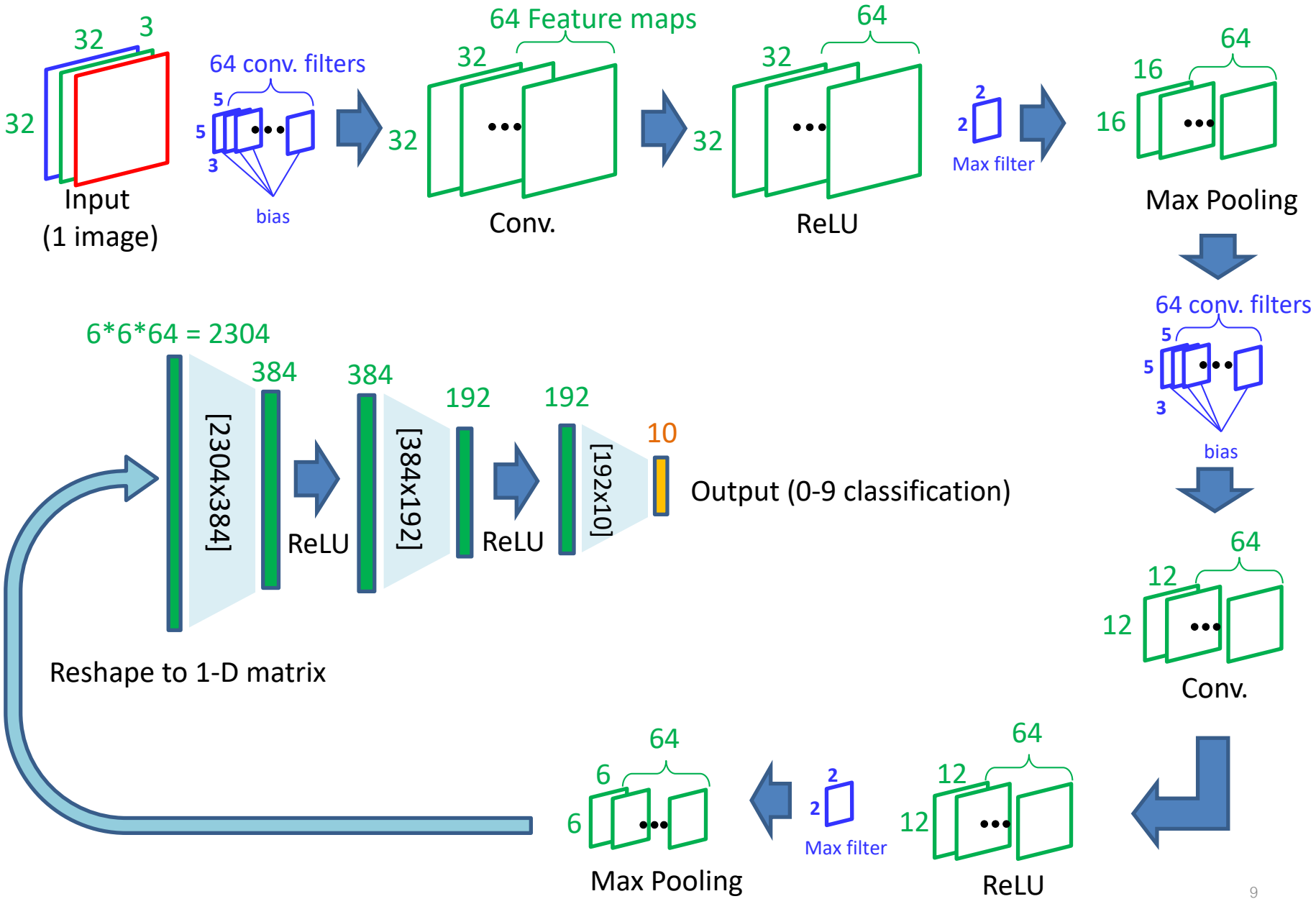
https://colab.research.google.com/drive/10zhiw6MeZrCVDKpkRFiWuQI1UWcnqQN-

Image preprocessing



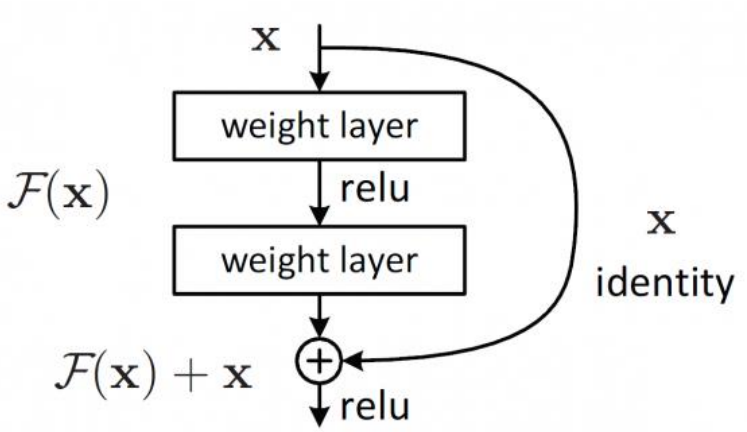
- There are still a lot more image preprocessing techniques, e.g., vertical flip, horizontal flip, etc.
- Further information: https://keras.io/preprocessing/image/

CNN architecture for CIFAR-10



Example 3: CIFAR-10 with ResNet

- We will train the CIFAR10 dataset with ResNet-50 architecture.
- ResNet-50 was trained on more than a million images from the ImageNet database.
- The network is 50 layers deep and can classify images into 1000 object categories, such as keyboard, mouse, pencil, and many animals.



Residual block

https://colab.research.google.com/drive/11H1UMrcZSG88uzLvGZD_HS0pbnbMpBff

```
history = model.fit(x_train, y_train, epochs=5, batch_size=20, validation_data=(x_test, y_test))
```

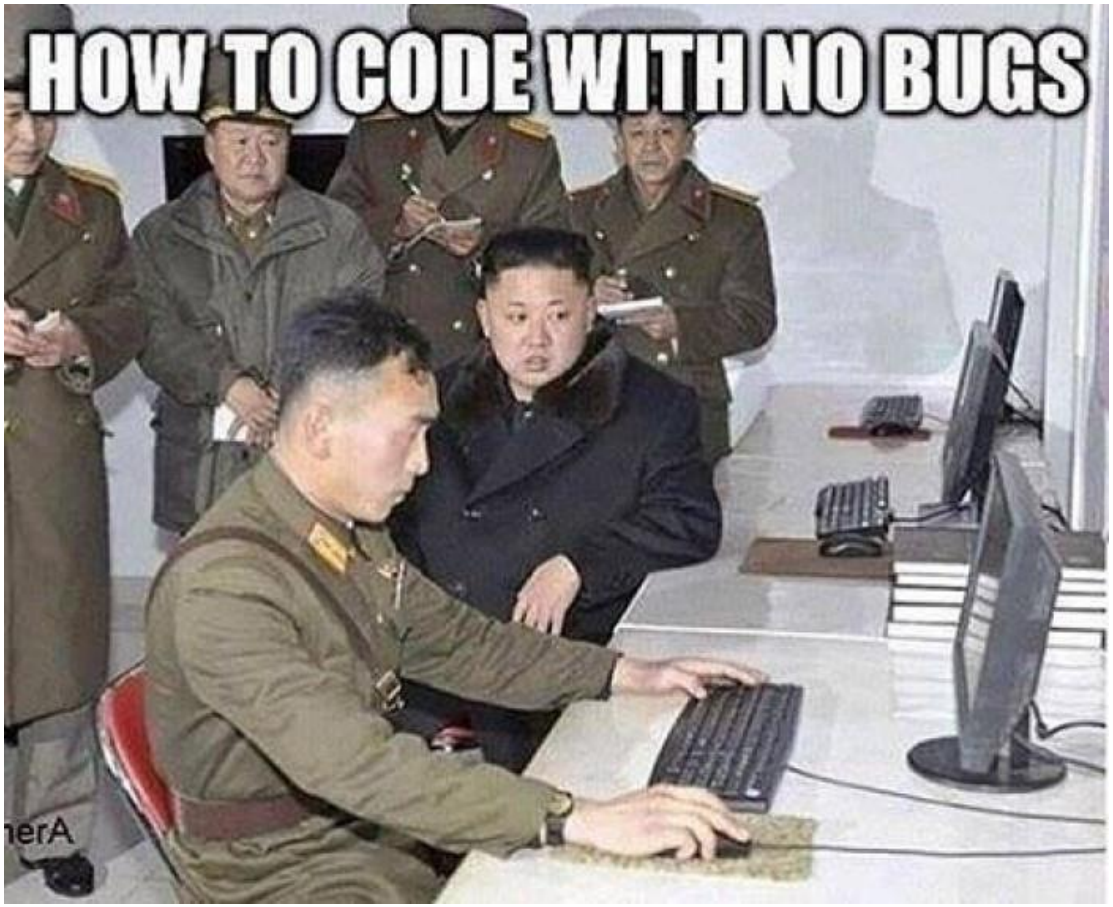
```
Train on 50000 samples, validate on 10000 samples
Epoch 1/5
50000/50000 [==============================] - 360s 7ms/sample - loss: 0.2126 - acc: 0.9244 - val_loss: 0.1757 - val_acc: 0.9385
Epoch 2/5
50000/50000 [==============================] - 353s 7ms/sample - loss: 0.1630 - acc: 0.9392 - val_loss: 0.1561 - val_acc: 0.9465
Epoch 3/5
50000/50000 [==============================] - 352s 7ms/sample - loss: 0.1384 - acc: 0.9481 - val_loss: 0.1326 - val_acc: 0.9527
Epoch 4/5
50000/50000 [==============================] - 351s 7ms/sample - loss: 0.1214 - acc: 0.9546 - val_loss: 0.1316 - val_acc: 0.9552
Epoch 5/5
50000/50000 [==============================] - 351s 7ms/sample - loss: 0.1064 - acc: 0.9602 - val_loss: 0.1365 - val_acc: 0.9563
```

```
model.evaluate(x_test, y_test)
```

```
10000/10000 [==============================] - 10s 967us/sample - loss: 0.1365 - acc: 0.9563
[0.13648285893201828, 0.95625997]
```

Assignment 2

- From Example 2, design your own CNN to improve the performance of the current convolutional neural network on CIFAR-10.
- For examples, insert more convolutional layers, use different value of stride, use different filter size, etc. But DO NOT modify training set and test set. Do not use pretrained models, e.g., ResNet, VGG.
- The number of training epochs is limited to 100 epochs.
- Due : two weeks. Early start your work. It may consume much more training time than you expected.
- Things to turn in :
  - Demo your code. All members must show up and declare their responsibilities, including answer the questions.
  - Sketch your model, including necessary parameter settings.
  - Training loss and accuracy graphs.
  - A screen capture of the latest training epochs.
  - Final test accuracy.
  - Your score depends on the test accuracy and the network simplicity.
- Warning: Cheating will result in getting zero score.


HOW TO CODE WITH NO BUGS

- Next class
  - Literature review of CNN architectures.