1- The files under the names 'train_dataF.bin' and 'train_labelsF.bin' contain the input images and labels of the neural network, respectively. They are binary files and are uint8-coded. Meaning that, for example, the pixels of an input image of size 51X51 (51^2=2601 pixels in total) are serialized in which each 1 Byte of data corresponds to the value of a single pixel. So the aforementioned image would then be a file of 1 ByteX2601 data arranged in a row (serialized). These files can be generated in Matlab, for example, using the following lines of codes.

```
f5=fopen('./ train_dataF.bin','w');
f6=fopen('./ eval_dataF.bin','w');
f7=fopen('./ train_labelsF.bin','w');
f8=fopen('./ eval_labelsF.bin','w');

fwrite(f5,X_train,'uint8');
fwrite(f6,X_test,'uint8');
fwrite(f7,Y_train,'uint8');
fwrite(f8,Y_test,'uint8');

fclose(f5);
fclose(f6);
fclose(f7);
fclose(f8);
```

The important point is that the python scripts expects uint8 coded data as input (hence, range of values between 0 to 255). Anything, apart from that scrambles the input. If higher dynamic range is needed, the script could be modified to support float32, etc.

2- All four files `train_dataF.bin,` `train_labelsF.bin,` `eval_dataF.bin,` and `eval_labelsF.bin` should be placed in the same folder as the python scripts. A set of training and evaluation datasets are provided as an example. The training input images dataset (train_dataF) contains 60000 images of speckle patterns of sizes 51X51. The label images (train_labelsF) dataset contains 60000 images of Latin alphabet characters of size 48X48. The evaluation datasets have the same corresponding sizes except the number of images being 20.

3- The python scripts code be open with a python interpreter, for example by **spyder** (to install spyder use the code *pip install spyder* in a tensorflow environment in command prompt.)

4- There are 4 python files: eye_train.py, eye.py, eye_eval.py and eye_input.py. Use eye_train.py to start training. Few things need to be adjusted based on the input-output images that are used for training the network. In the current format, the input images should have size 51X51 and the output images (labels) should be of size 48X48. If other input-output size is required, the variables indicating these sizes should be changed (defined in the eye_train.py script). Accordingly, the architecture of the network should be modified to accommodate the new image sizes.

5- Currently, the network has the architecture of a Residual network (ResNets) as outlined here: Rahmani, B., Loterie, D., Konstantinou, G., Psaltis, D. & Moser, C. Multimode optical fiber transmission with a deep learning network. *Light: Science & Applications* **7**, 69 (2018).

6- The training parameters should be set in the eye_train.py script.

batch_size=32
IMAGE_SIZE = 51
IMAGE_SIZE_label = 48
learning_rate=0.001

```
step0m=1000 # Number of steps in one epoch.
NUMTRAIN_SAMPLES=60000
NUMEVAL_SAMPLES=20
```

7- The script has been written in a way that after training for step0m steps, a validation of training on evaluation dataset is executed automatically. since all evaluation dataset is used in the evaluation phase (as opposed to the training phase in which one batch of training data is used for each step), the evaluation dataset should not be very large, otherwise an out of memory error will be thrown (please do some experiments to figure out how big of a dataset your machine handles).

8- After training is complete, optional testing a completely new dataset could be carried out. In order to do that, eye_eval.py script should be run autonomously (first set the size of the images in eye_eval.py).

9- The predicted output images both in the validation phase after each epoch of training and in the optional testing is produced in the same directory under the name 'predicted_output.bin'.

Materials and codes are prepared and collected by Babak Rahmani on 22.07.2019, EPFL, Lausanne.