Keras has the method fit\_generator() in its models. It accepts a python generator or a keras Sequence as input.

You can create a simple generator like this:

fileList = listOfFiles

def imageLoader(files, batch\_size):

L = len(files)

#this line is just to make the generator infinite, keras needs that

while True:

batch\_start = 0

batch\_end = batch\_size

while batch\_start < L:

limit = min(batch\_end, L)

X = someMethodToLoadImages(files[batch\_start:limit])

Y = someMethodToLoadTargets(files[batch\_start:limit])

yield (X,Y) #a tuple with two numpy arrays with batch\_size samples

batch\_start += batch\_size

batch\_end += batch\_size

And fit like this:

model.fit\_generator(imageLoader(fileList,batch\_size),steps\_per\_epoch=..., epochs=..., ...)

Normally, you pass to steps\_per\_epoch the number of batches you will take from the generator.

You can also implement your own [Keras Sequence](https://keras.io/utils/#sequence). It's a little more work, but they recommend using this if you're going to make multi-thread processing.