ML Final Project report
Arian Jalalian, Mahdieh Moghieseh

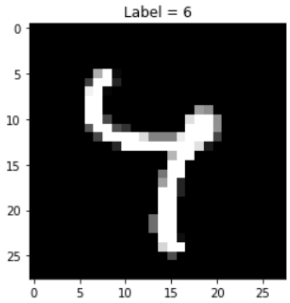
Step 1 : Dataset

The purpose of the project is to classify persian handwritten digits. We wrote 20 of each digit on a white and without line piece of paper, and took a picture of each 50 one, then cropped each image to extract digits. First attempt was not successful because the size of the cropped image resulted in low quality 28x28 images. So we did another round of this step, taking pictures carefully and got good results.

Before processing

Step 2 : Preprocessing

We gathered data in a folder with correct order but with not meaningfull file names, then we implemented a .py code that renames each image with its corresponding digit represnitng. Then based on the file name, the correct labeling happens.
With each image, a sequence of actions is taken, like thresholding, NOTing pixels and resizing to achieve MNIST like images.



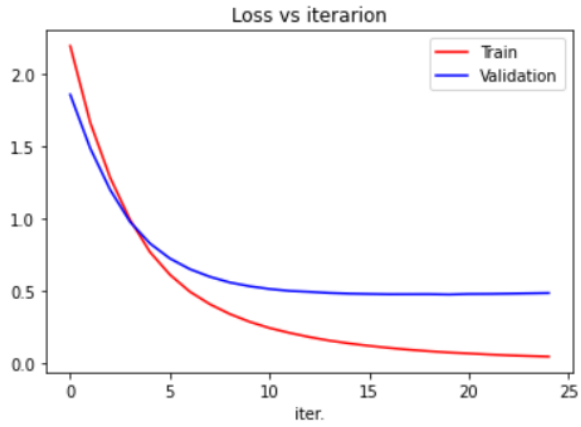
After that, splitting is going to happen, 75% for train, 15% for test and 10% for validation. It is crucial to note that due to shortage of number of instances, a stratified split should happen to assure that each set has equal number of each digit within it.

Step 3 : ANN

A simple ANN architecture is employed with the following details :

Layer (type)	Output Shape	Param #
flatten_2 (Flatten)	(None, 784)	0
dense_4 (Dense)	(None, 128)	100480
dense_5 (Dense)	(None, 10)	1290

Model is trained on training set with optimizer set to adam for adjusting learning rate, and loss to sparse_categorical_crossentropy which doesn't need labels in a one-hot encoding manner.
We reached good accuracy after 25 epochs (90%)



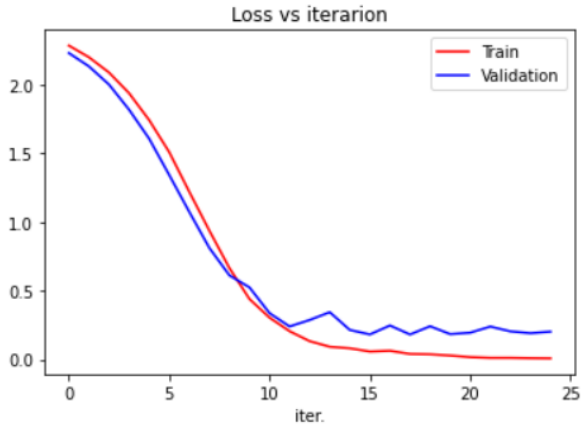
Step 4 : CNN

A simple CNN architecture is employed with the following details :

Layer (type)	Output Shape	Param #
conv2d_21 (Conv2D)	(None, 26, 26, 32)	320
max_pooling2d_21 (MaxPooling2D)	(None, 13, 13, 32)	0
conv2d_22 (Conv2D)	(None, 11, 11, 64)	18496
max_pooling2d_22 (MaxPooling2D)	(None, 5, 5, 64)	0

conv2d_23 (Conv2D)	(None, 3, 3, 64)	36928
max_pooling2d_23 (MaxPooling2D)	(None, 1, 1, 64)	0
flatten_16 (Flatten)	(None, 64)	0
dense_32 (Dense)	(None, 32)	2080
dense_33 (Dense)	(None, 10)	330

Model is trained on training set with optimizer set to adam for adjusting learning rate, and loss to sparse_categorical_crossentropy which doesn't need labels in a one-hot encoding manner.
We reached good accuracy after 25 epochs (93%)

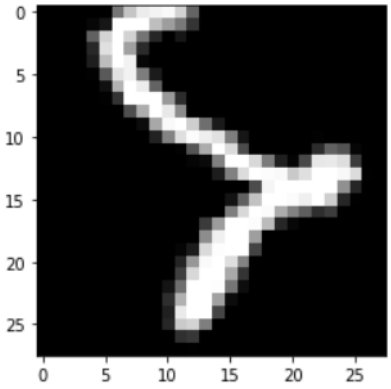


Step 4 : Data Augmentation

Based on the performance of the models on bigger unseen datasets, and the fact that our dataset is really small, data augmentation is needed.
The primary goal of data augmentation is to expose the model to a wider range of variations and scenarios that it might encounter during the testing phase. By generating new instances of the data through transformations, the model becomes more adept at handling different perspectives, scales, rotations, and other variations in the input data. In our example only rotation and horizontal flip is used, vertical is not an option because it turns 7 to 8 and vice versa.

```
augmenter = ImageDataGenerator(
    rotation_range=30,
    horizontal_flip=True,
    vertical_flip=False
   )
```

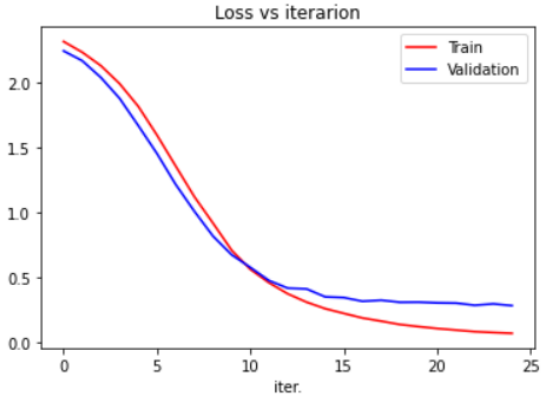
After the augmentation, our dataset got bigger by a factor of 5.



Also testing our new big datasets on previous models, shows that those didn't do well enough on this bigger dataset comparing to our small dataset. (73% accuracy)

Step 5 : Transfer learning

The main idea of transfer learning is to start learning in a better condition. Like a person that knows how to play violin has an easier struggle on learning violonsel compared to a completely unfamiliar guy.
So we trained a CNN on MNIST dataset and took early stages of training that generally learns basic things like edges, and put it into a new network with new layers on top of it to learn persian handwritten digits specifically. We reached 93% accuracy.



Step 6 : Multidigit

If we come up with an idea that extracts digits in a picture in the correct order, then with the help of our model it would be easy to predict multi digits numbers. We did image processing on an image containing a multi digit number like 4951 to get 4, 9, 5 and 1.
Contouring is a computer vision and image processing technique used to detect and represent the boundaries of objects or regions within an image. It involves identifying continuous curves that form the outline of objects, enabling further analysis, object recognition, or manipulation.

Contouring is fundamental to object recognition tasks where identifying the boundaries of objects in an image is crucial. This is commonly used in fields like computer vision for image classification and segmentation. After extracting each digit, we then repeat step 1 and feed the results into our model