**Alphabet Mnist Dataset : (24 alphabets: J and Z deleted as they include gesture movements)**

The training data (27,455 cases) and test data (7172 cases) : each 24x24 pixels

Dataset: <https://www.kaggle.com/datamunge/sign-language-mnist>

**Notebook: accuracy: 95.3**

<https://colab.research.google.com/drive/1-Se6nlPuBEhSkLDbjB-hJ60ucCw668wd?usp=sharing>

**Alphabet Dataset :**

The training data set contains 87,000 images which are 200x200 pixels. There are 29 classes, of which 26 are for the letters A-Z and 3 classes for *SPACE*, *DELETE* and *NOTHING*.

Dataset: <https://www.kaggle.com/grassknoted/asl-alphabet>

**Notebook-1: accuracy: 98.9**

<https://drive.google.com/file/d/1RQNDuLsfUuiO4CzocJ9bd7GbKrDg4saP/view?usp=sharing>

|  |
| --- |
| model = Sequential()  model.add(Conv2D(64, (3, 3), padding='same', input\_shape=(32, 32, 3), activation='relu'))  model.add(MaxPooling2D(pool\_size=(2, 2)))  model.add(Conv2D(128, (3, 3), padding='same', input\_shape=(32, 32, 3), activation='relu'))  model.add(MaxPooling2D(pool\_size=(2, 2)))  model.add(Conv2D(256, (3, 3), padding='same', input\_shape=(32, 32, 3), activation='relu'))  model.add(MaxPooling2D(pool\_size=(2, 2)))  model.add(BatchNormalization())  model.add(Flatten())  model.add(Dropout(0.5))  model.add(Dense(1024, activation='sigmoid'))  model.add(Dense(classes, activation='softmax')) |

**Notebook-2: accuracy 99.8**

<https://drive.google.com/file/d/1BBo1EfjuST8iDIVZkIqhtjYTfU_-xvxX/view?usp=sharing>

|  |
| --- |
| model = Sequential()  model.add(Conv2D(filters = 64, kernel\_size = 5, padding = 'same', activation = 'relu', input\_shape = (64, 64, 1)))  model.add(Conv2D(filters = 64, kernel\_size = 5, padding = 'same', activation = 'relu'))  model.add(MaxPooling2D(pool\_size = (4, 4)))  model.add(Dropout(0.5))  model.add(Conv2D(filters = 128 , kernel\_size = 5, padding = 'same', activation = 'relu'))  model.add(Conv2D(filters = 128 , kernel\_size = 5, padding = 'same', activation = 'relu'))  model.add(MaxPooling2D(pool\_size = (4, 4)))  model.add(Dropout(0.5))  model.add(Conv2D(filters = 256 , kernel\_size = 5, padding = 'same', activation = 'relu'))  model.add(Dropout(0.5))  model.add(Flatten())  model.add(Dense(29, activation='softmax')) |

**Alphabet + Digits:**

36 classes - 25327 images

<https://www.kaggle.com/ayuraj/american-sign-language-dataset>

Notebook-1: <https://www.kaggle.com/joshithareddy/americal-sign-language-training>

**Accuracy - 95**

3 layers of CNN are implemented.

|  |
| --- |
| model = Sequential()    model.add(Conv2D(64, kernel\_size = [3,3], padding = 'same', activation = 'relu', input\_shape = (200,200,3)))  model.add(MaxPool2D(pool\_size = [3,3]))    model.add(Conv2D(128, kernel\_size = [5,5], padding = 'same', activation = 'relu'))  model.add(MaxPool2D(pool\_size = [3,3]))  model.add(Conv2D(256, kernel\_size = [3,3], padding = 'same', activation = 'relu'))  model.add(MaxPool2D(pool\_size = [3,3]))    model.add(BatchNormalization())  model.add(Flatten())  model.add(Dropout(0.5))    model.add(Dense(1024, activation='sigmoid'))  model.add(Dense(36, activation = 'softmax')) |

[**Notebook-2:**](https://drive.google.com/file/d/1QtRCZrTtE2YPMdxWhAJczsDF0ABgrAfv/view?usp=sharing) **accuracy: 98**

<https://colab.research.google.com/drive/1nCrdD9OA0NaQYAZ8BsqttMdg2ofcZpSj?usp=sharing>

4 layers of CNN

|  |
| --- |
| model = Sequential()    model.add(Conv2D(64, kernel\_size = [3,3], padding = 'same', activation = 'relu', input\_shape = (200,200,3)))  model.add(Conv2D(64, kernel\_size = [3,3], padding = 'same', activation = 'relu'))  model.add(MaxPool2D(pool\_size = [3,3]))    model.add(Conv2D(128, kernel\_size = [5,5], padding = 'same', activation = 'relu'))  model.add(Conv2D(128, kernel\_size = [5,5], padding = 'same', activation = 'relu'))  model.add(MaxPool2D(pool\_size = [3,3]))    model.add(Conv2D(256, kernel\_size = [3,3], padding = 'same', activation = 'relu'))  model.add(Conv2D(256, kernel\_size = [3,3], padding = 'same', activation = 'relu'))  model.add(Conv2D(256, kernel\_size = [3,3], padding = 'same', activation = 'relu'))  model.add(MaxPool2D(pool\_size = [3,3]))    model.add(Conv2D(512, kernel\_size = [3,3], padding = 'same', activation = 'relu'))  model.add(Conv2D(512, kernel\_size = [3,3], padding = 'same', activation = 'relu'))  model.add(MaxPool2D(pool\_size = [3,3]))    model.add(Conv2D(512, kernel\_size = [3,3], padding = 'same', activation = 'relu'))  model.add(MaxPool2D(pool\_size = [2,2]))    model.add(BatchNormalization())  model.add(Flatten())  model.add(Dropout(0.5))    model.add(Dense(1024, activation = 'relu', kernel\_regularizer = regularizers.l2(0.001)))  model.add(Dense(512, activation = 'relu', kernel\_regularizer = regularizers.l2(0.001)))  model.add(Dense(36, activation = 'softmax')) |