Source: CanStockPhoto.com,

What would you do if you can't communicate?

We have experienced it

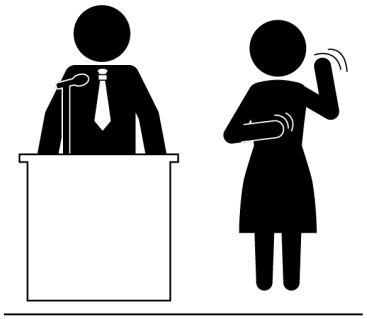
Traveling to another country

Source: Toolshero image

Somebody with an emergency

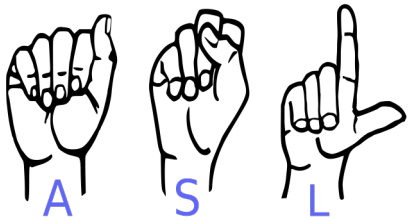
Source: Boston children's hospital

Job as a translator



Source: pngegg.com

Somebody who can't communicate verbally

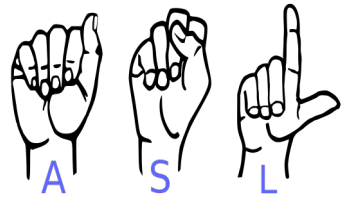


Source: Wikipedia by Psiĥedelisto

Arturo Bravo

ASL Translator

The goal



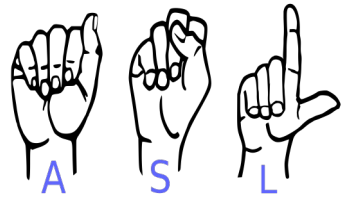
Source: Wikipedia by Psiĥedelisto

To create a tool, to help people with a hearing loss disability, to communicate with another person by translating a message emitted in the American Sign Language (ASL) into text.

What is ASL?

- By definition: American Sign Language is a complete and organized visual language that is expressed by both manual and nonmanual features
- ASL has its own syntax and grammar
- Natural language for many people in the world

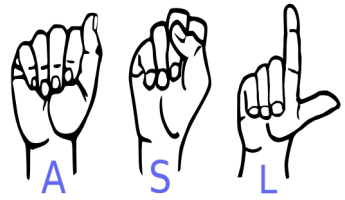
Context on the situation



Source: Wikipedia by Psiĥedelisto

- ASL as natural language for ~500,000 people in the US and Canada
- 28 million Americans have some degree of hearing loss
- About 2 million of these 28 million people are classified as deaf
- 200,000 of these 2 million were born deaf
- More than 70 million deaf people worldwide (80% in developing countries)

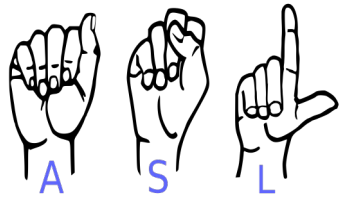
But, who cares?



Source: Wikipedia by Psiĥedelisto

- Service companies
- Companies who take care of their employees
- Health and emergency systems institutions
- Companies who develop video conference tools
- Language schools
- Governments

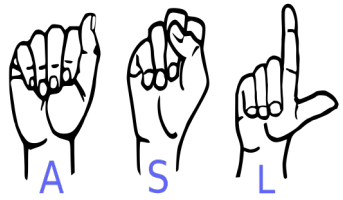
How to do it



Source: Wikipedia by Psiĥedelisto

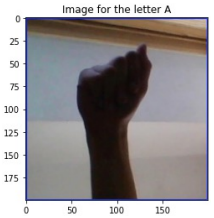
- Gather a lot of information (images)
- Ensure data is reliable
- Use image recognition
- Rely on CNN
- Trial and error
- Go beyond!

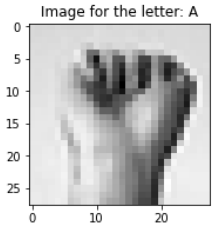
The data

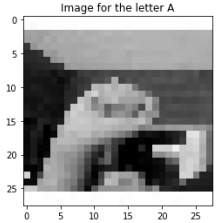


Source: Wikipedia by Psiĥedelisto

There are 146,627 images divided into 3 datasets.







- Data set No. 1 - Name: **ASL,** source: Kaggle

  a) Number of images: 84,000
  b) Number of classes: 28 (Alphabet and delete, space)
  c) Format: Color images in size 200 x 200

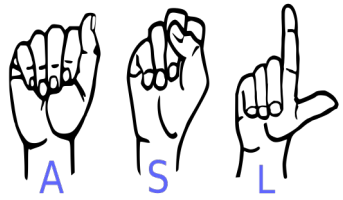
- Data set No. 2 - Name: **MNIST,** source: Kaggle

  a) Number of images: 34,627 (train: 27455, test: 7172)
  b) Number of classes: 24 (Alphabet except 'J' and 'Z')
  c) Format: A csv file of grayscale images of size 28 x 28

- Data set No. 2 - Name: **APPLE_CAM,** source: Self made

  a) Number of images: 28,000
  b) Number of classes: 28 (Alphabet and delete, space)
  c) Format: Grayscale images in size 28 x 28

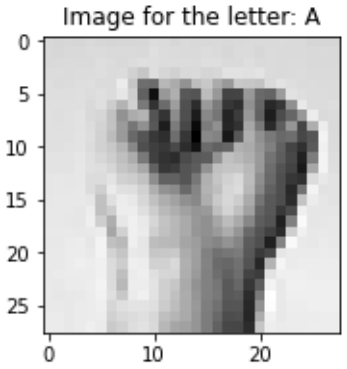
Analyzing the data



Source: Wikipedia by Psiĥedelisto

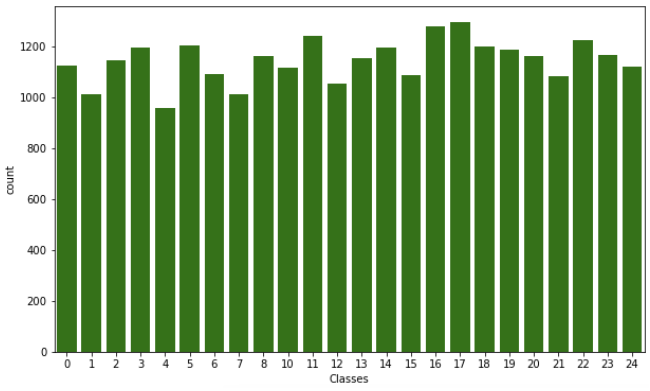
The **MNIST** set is a csv file with 784 (28 x 28) pixels per observation (image).

- Missing values: 0
- Duplicated values: 0
- Values out of range (0-255): 0
- Data balance: some imbalance

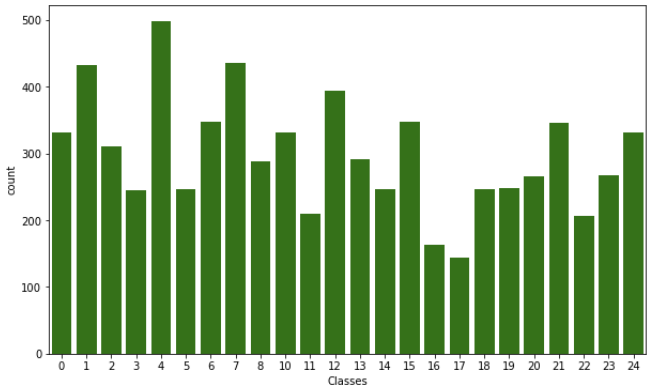


Arturo Bravo

Classes on the train set

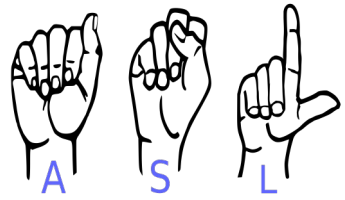


Classes on the test set



ASL Translator

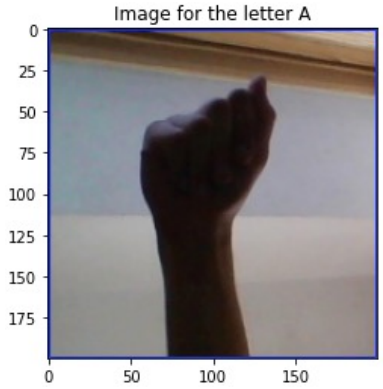
Analyzing the data

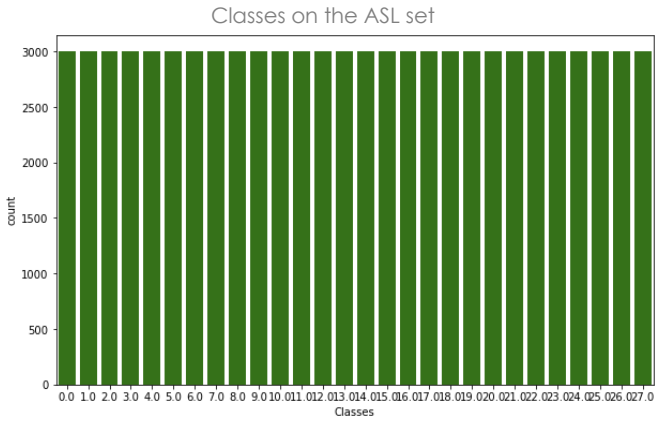


Source: Wikipedia by Psiĥedelisto

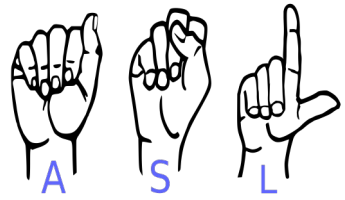
The **ASL** set is a group of 84,000 color images in size 200 x 200 pixels.

- Extract images from directory and resize to 28 x 28 pixels
- Save as a csv file to standardize
- Missing values or values out of range: 0
- Data balance: Perfect balance (3,000 images per class)





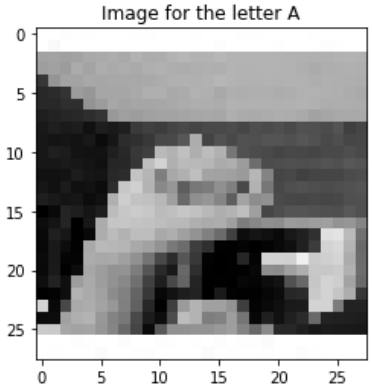
Analyzing the data



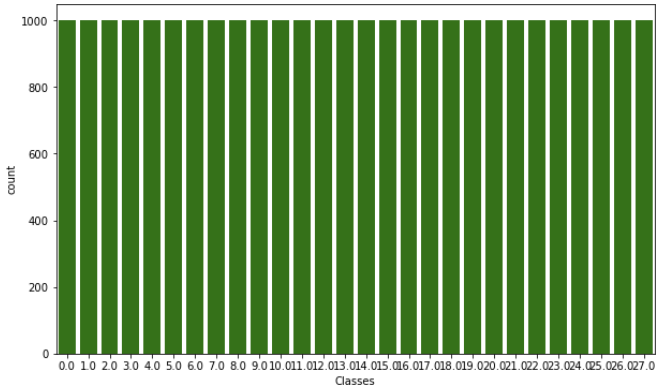
Source: Wikipedia by Psiĥedelisto

The **APPLE_CAM** set is a group of 28,000 grayscale images in size 28 x 28 pixels.

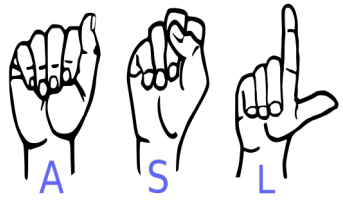
- Extract images from directory
- Save as a csv file to standardize
- Missing values or values out of range: 0
- Data balance: Perfect balance (1,000 images per class)



Classes on the APPLE_CAM set



Analyzing the data



Source: Wikipedia by Psiĥedelisto

Put everything together.

- Integrated set: 139,455
- Train set: 109,455 (reduce the integrated set by 30,000 images)
- Train_validation split:
  - Train set: 98509
  - Validation set: 10946
- Test set: 37,172 (30,000 images from integrated set + MNIST test set)

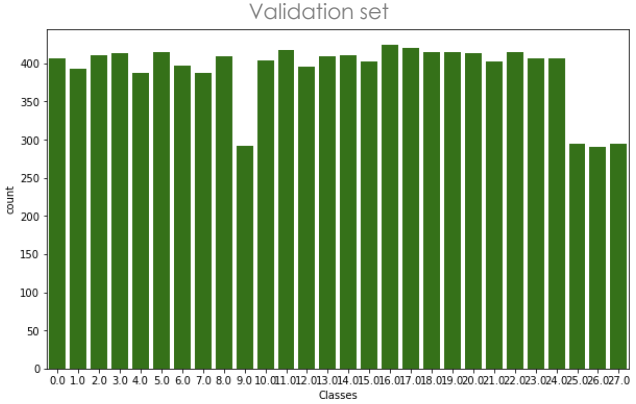
Integrated set

```
2 train = pd.concat([train_MNIST, train_APPLE_CAM])
3 train = pd.concat([train, train_ASL])
4 train.shape
```

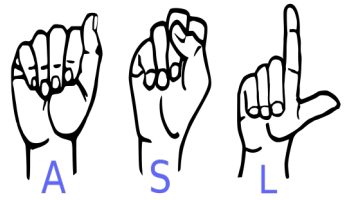
```
(139455, 785)
```

Train set

Validation set



Preprocessing the data



Source: Wikipedia by Psiĥedelisto

Getting the information ready for modeling.

- One-hot encoded labels
- Normalize pixel values to [0,1]
- Missing values or values out of range: 0
- Reshape the observations (images) to 28 x 28 x 1 arrays

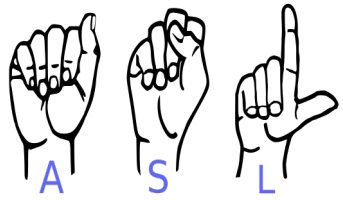
One-hot encoded labels

```
encoder = LabelBinarizer()
y_train = encoder.fit_transform(y_train)
y_val = encoder.transform(y_val)
y_test_enc = encoder.transform(y_test)
```

Reshaped arrays

```
X_train = X_train.to_numpy().reshape(-1,28,28,1)
X_val = X_val.to_numpy().reshape(-1,28,28,1)
X_test = X_test.to_numpy().reshape(-1,28,28,1)
```

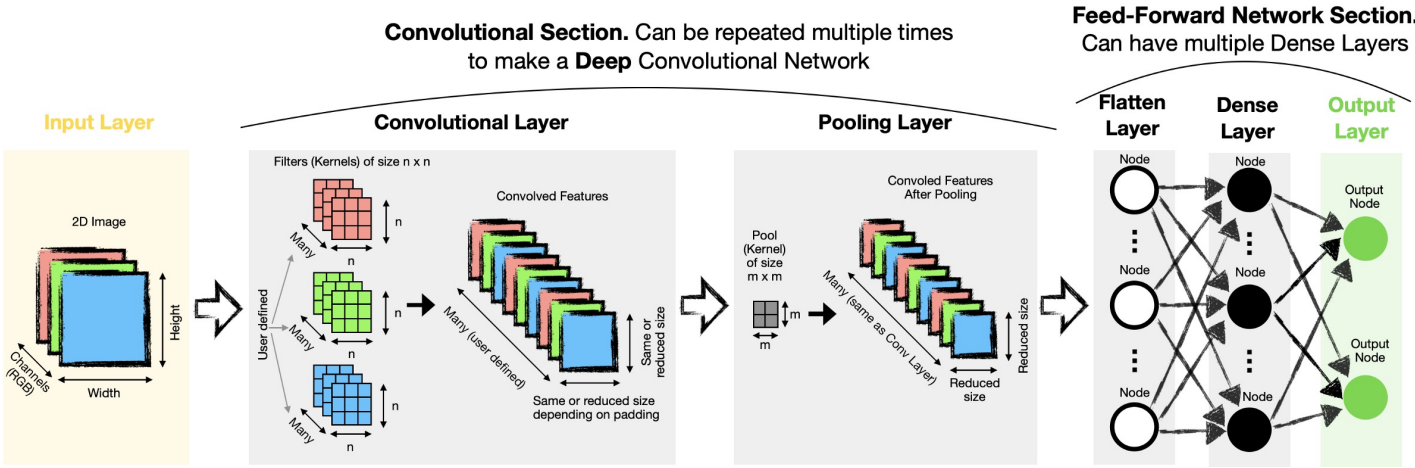
Overview of CNNs



Source: Wikipedia by Psiĥedelisto

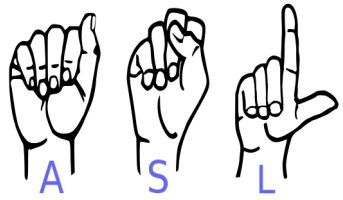
A quick reference on Convolutional Neural Networks (CNN).

- Three main components: Input, hidden, output
- Different types of hidden layers: Convolutional, pooling, dropout, etc.
- Number of hidden layers is self decided



Reference TDS: By Saul Dobilas

Overview of CNNs

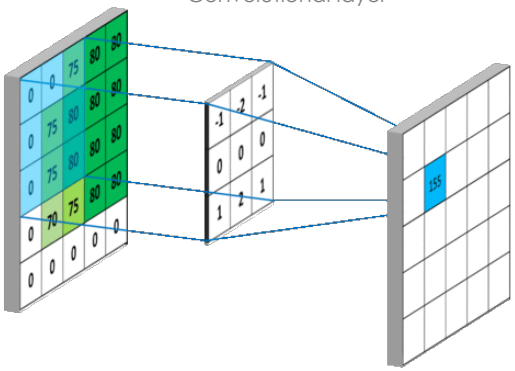


Source: Wikipedia by Psiĥedelisto

A quick reference on Convolutional Neural Networks (CNN).

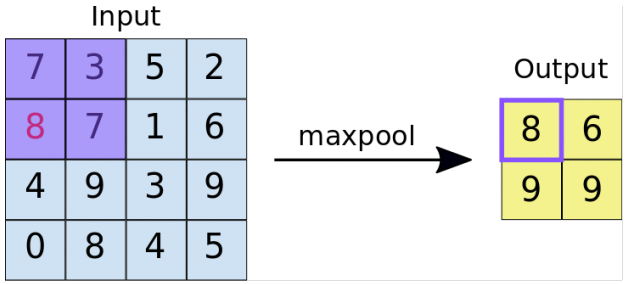
- Convolutional layer: Apply filters to images
- Pooling: Max or Avg to reduce feature map
- Dropout: Avoid overfitting
- Flatten and Dense layers
- Output layer: Classify the images

Convolutional layer



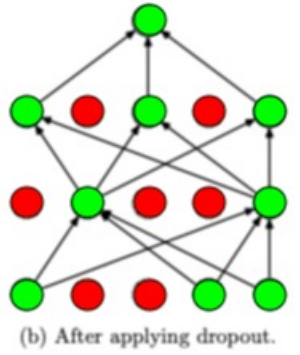
Reference: mlnotebook.github.io

Pooling layer



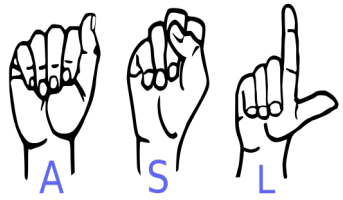
Reference: NumPyNet

Dropout layer



Reference: Adapted from Srivastava, Nitish

Modeling



Source: Wikipedia by Psiĥedelisto

6 CNNs were trained.

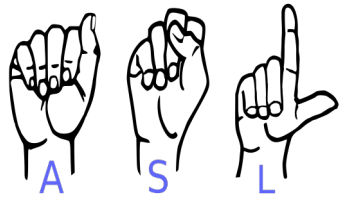
- Data augmentation used in 4 CNNs
- Learning rate and epochs increased accuracy
- Best 2 models with accuracy greater than 97%
- Chosen model with no augmented data: 99.8% accuracy

Best CNN with 99.8%

Layer (type)	Output Shape	Param #
conv2d_14 (Conv2D)	(None, 28, 28, 32)	320
batch_normalization_10 (BatchNormalization)	(None, 28, 28, 32)	128
max_pooling2d_9 (MaxPooling2D)	(None, 14, 14, 32)	0
conv2d_15 (Conv2D)	(None, 14, 14, 64)	18496
dropout_16 (Dropout)	(None, 14, 14, 64)	0
batch_normalization_11 (BatchNormalization)	(None, 14, 14, 64)	256
conv2d_16 (Conv2D)	(None, 14, 14, 128)	73856
dropout_17 (Dropout)	(None, 14, 14, 128)	0
batch_normalization_12 (BatchNormalization)	(None, 14, 14, 128)	512
max_pooling2d_10 (MaxPooling2D)	(None, 7, 7, 128)	0
conv2d_17 (Conv2D)	(None, 7, 7, 128)	147584
dropout_18 (Dropout)	(None, 7, 7, 128)	0
batch_normalization_13 (BatchNormalization)	(None, 7, 7, 128)	512
max_pooling2d_11 (MaxPooling2D)	(None, 4, 4, 128)	0
flatten_5 (Flatten)	(None, 2048)	0
dense_15 (Dense)	(None, 512)	1049088
dropout_19 (Dropout)	(None, 512)	0
dense_16 (Dense)	(None, 512)	262656
dropout_20 (Dropout)	(None, 512)	0
dense_17 (Dense)	(None, 512)	262656
dropout_21 (Dropout)	(None, 512)	0
dense_18 (Dense)	(None, 28)	14364
dropout_27 (Dropout)	(None, 28)	0
dropout_33 (Dropout)	(None, 28)	0

Total params: 1,830,428
Trainable params: 1,829,724
Non-trainable params: 704

Modeling

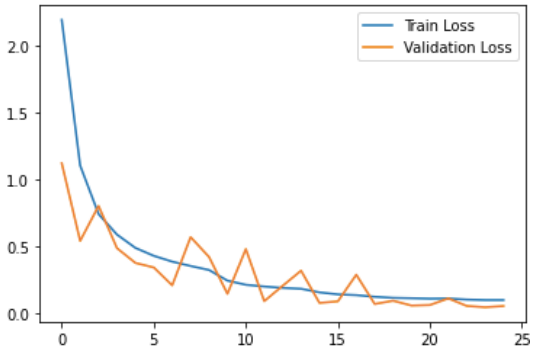


Source: Wikipedia by Psiĥedelisto

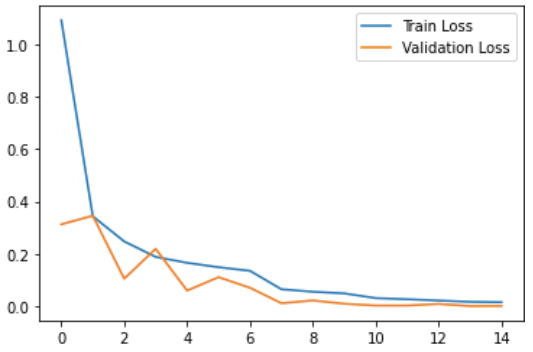
6 CNNs were trained.

- Improved train and validation accuracy
- Learning rate and epochs increased accuracy
- Best DA model: 948/37,172 misclassified
- Best no DA model: 71/37,172 misclassified

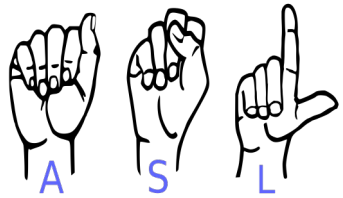
Best CNN with data augmented 97.5%



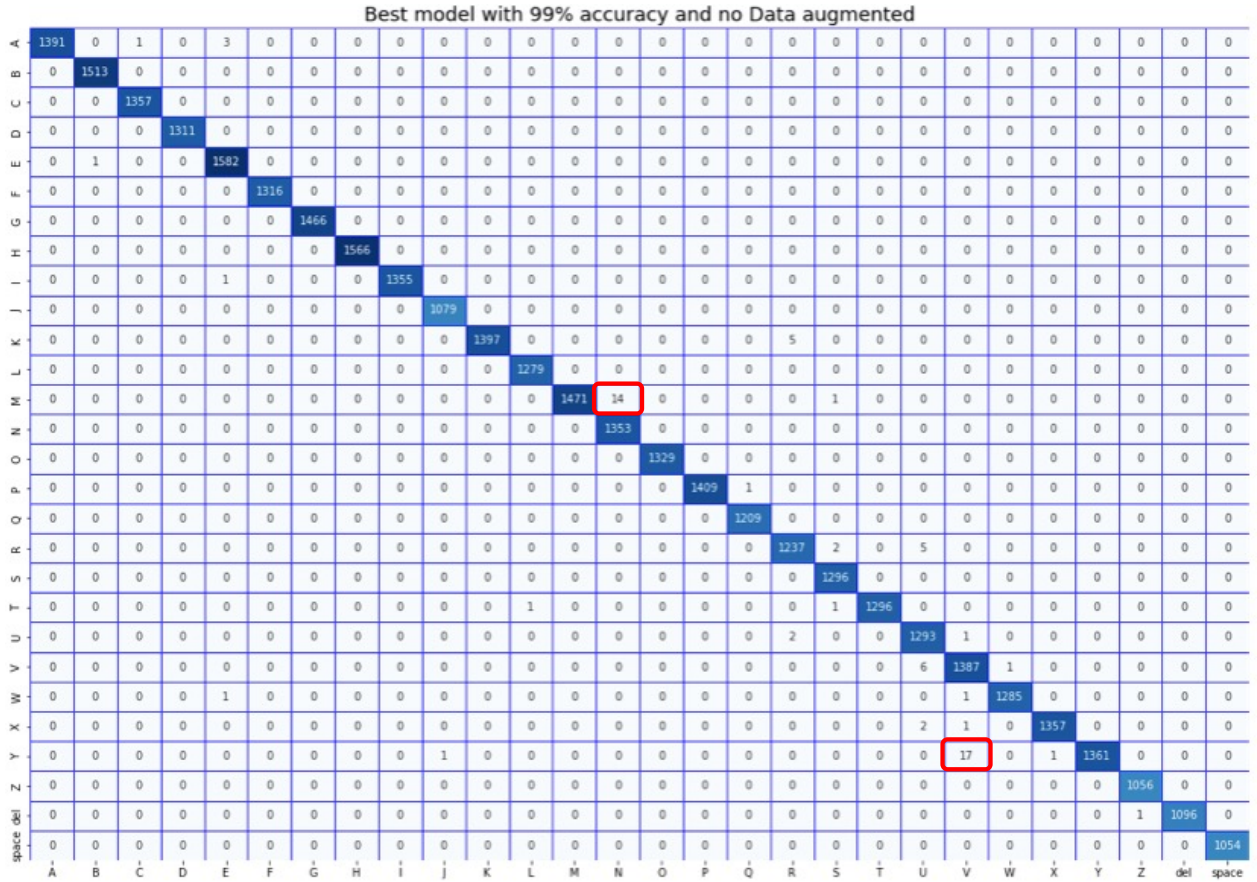
Best CNN with no data augmented 99.8%



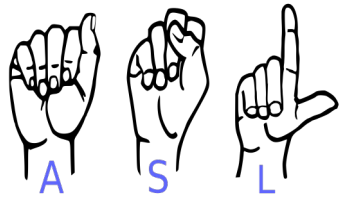
Modeling



Source: Wikipedia by Psiĥedelisto



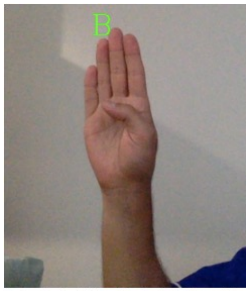
Modeling



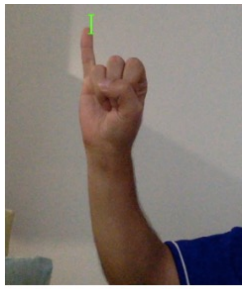
Source: Wikipedia by Psiĥedelisto

Correctly classified

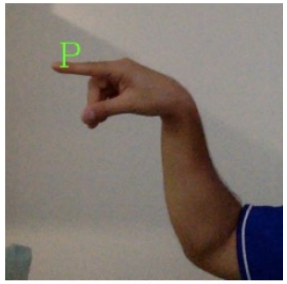
Predicted class: A, Actual class: A



Predicted class: B, Actual class: B

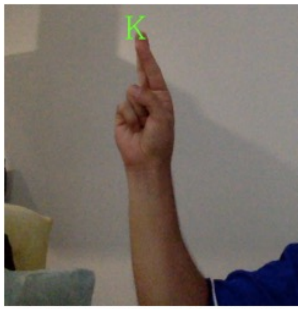


Predicted class: I, Actual class: I



Predicted class: P, Actual class: P

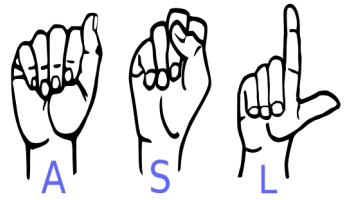
Incorrectly classified



Predicted class: K, Actual class: R

Predicted class: E, Actual class: X

Modeling

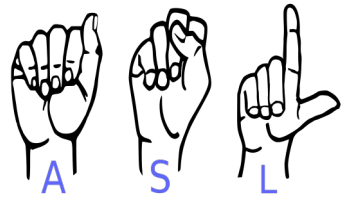


Source: Wikipedia by Psiĥedelisto

Comparison table of 6 CNNs trained

CNN	Data augmented	Number of parameters	Reduce learning rate	Epochs	Train Accuracy	Validation Accuracy	Test accuracy	Number misclassified
1	No	210,204	No	5	90.5%	91.1%	80.3%	-
2	Yes	210,204	No	5	33.9%	42.1%	35.6%	-
3	Yes	6,719,260	No	5	42.5%	55.0%	46.2%	-
4	Yes	1,830,428	Yes	5	71.5%	80.4%	75.6%	-
5	Yes	1,830,428	Yes	25	96.9%	98.4%	97.5%	948/37,172
6	**No**	**1,830,428**	**Yes**	**15**	**99.5%**	**99.9%**	**99.8%**	**71/37,172**

Conclusions



Source: Wikipedia by Psiĥedelisto

- Powerful CNNs for image recognition
- Understand every component's role in the CNN
- Is training with augmented data always better?
- Not just adding more layers or parameters
- The importance of image size

Next steps:

- Gather more data
- Explore transfer learning
- Train a model on a wider range of classes
- Test results on real time video
- Create application
- Distribute