

# Neural Network for Sign Language Digits Classification

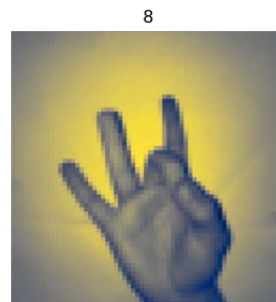
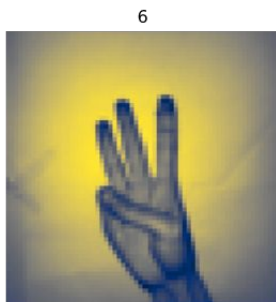
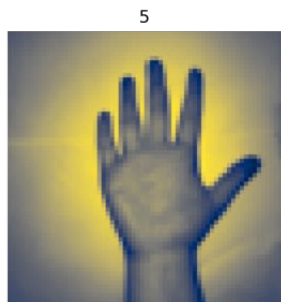
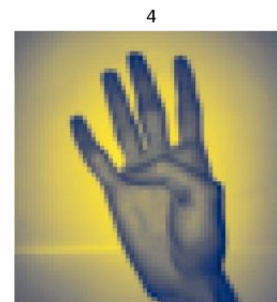
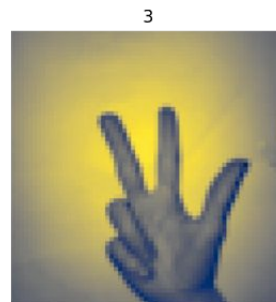
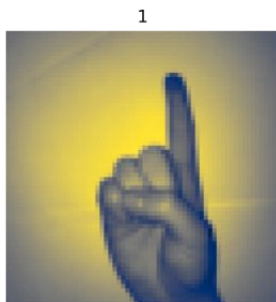
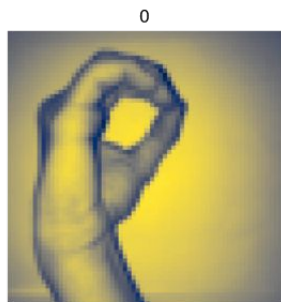
Jan Maděra

# Table of content

1. Problem
2. Dataset
3. Preprocessing
4. Networks
5. Experiments
6. Results
7. Possible improvements

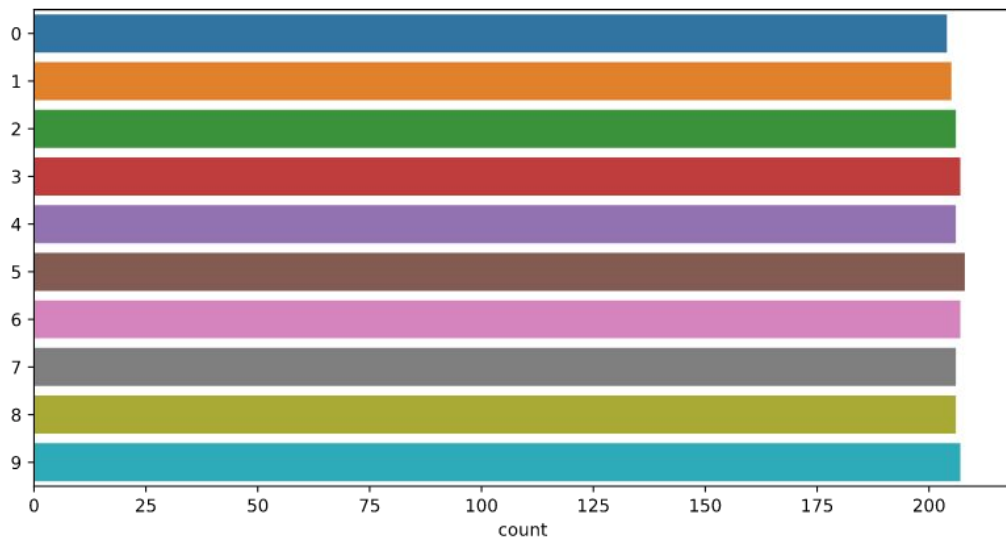
# Problem

- Classification of digits 0-9



# Dataset

- Source: <https://www.kaggle.com/ardamavi/sign-language-digits-dataset>
- Size: 2062 images
- Balanced dataset



# Preprocessing

- Wrong sample-target ordering → Reorganize data
- Train test split:
  - 1649 training samples (80%)
  - 413 testing samples (20%)
- Additional data generation (augmentation)
  - Rotation ( $-20^\circ$ ,  $+20^\circ$ )
  - Zoom (0.9, 1.2)
  - Brightness (needed only in real-life cases with bad lighting conditions)
  - Reduce overfitting

# Network architecture

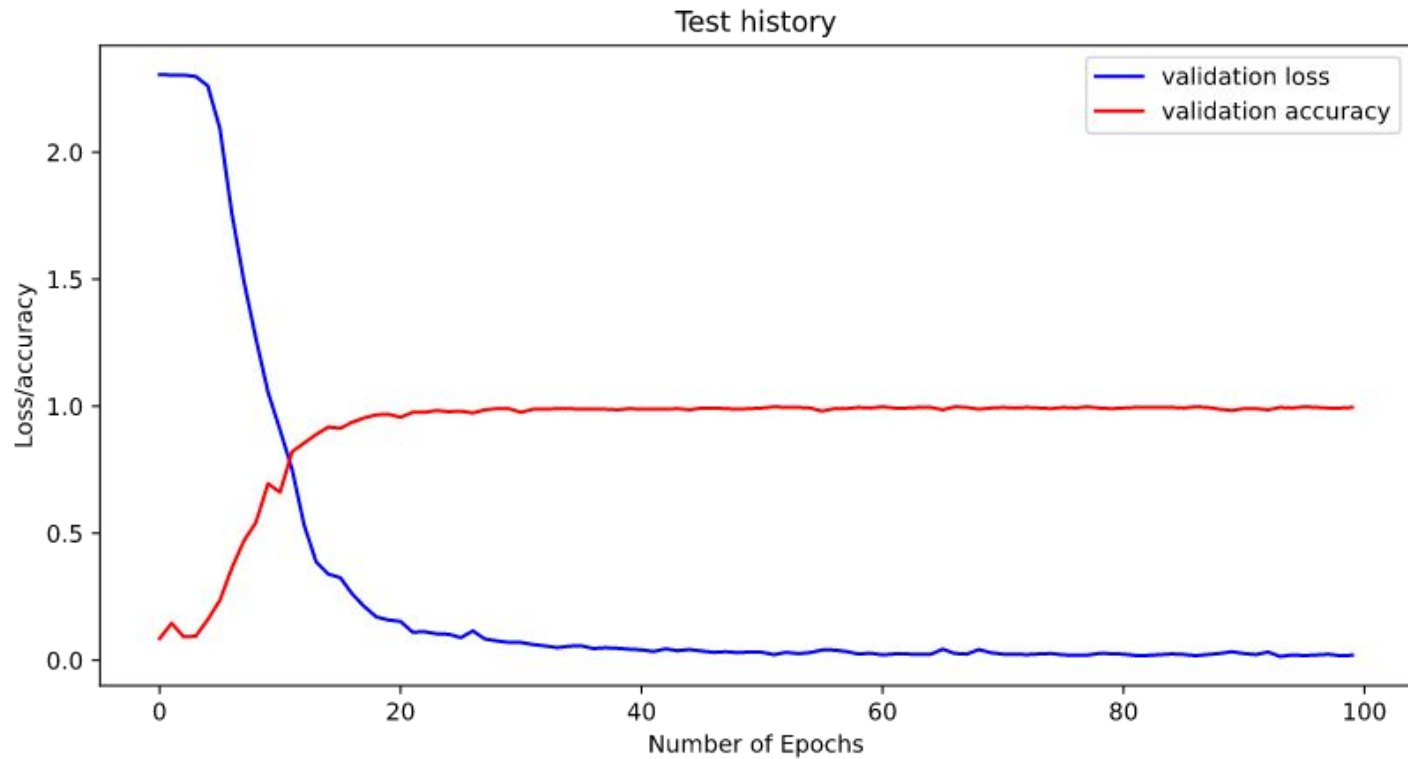
- Input layer
- Dense layers
- Convolutional layers
- Pool layers
- Dropout layers (reduce overfitting)

# Experiments

- 100 epochs, batch size = 250

Layers	Loss (categorical crossentropy)	Accuracy (testing data) /Without data augmentation
2 dense	0.7458	78.2% (75.7%)
1 conv, 1 dense	0.1008	97.8% (93.2%)
3 conv, 1 dense	0.0288	99.1% (97.8%)
3 conv, 2 dense	0.0216	99.5% (98%)

# Results





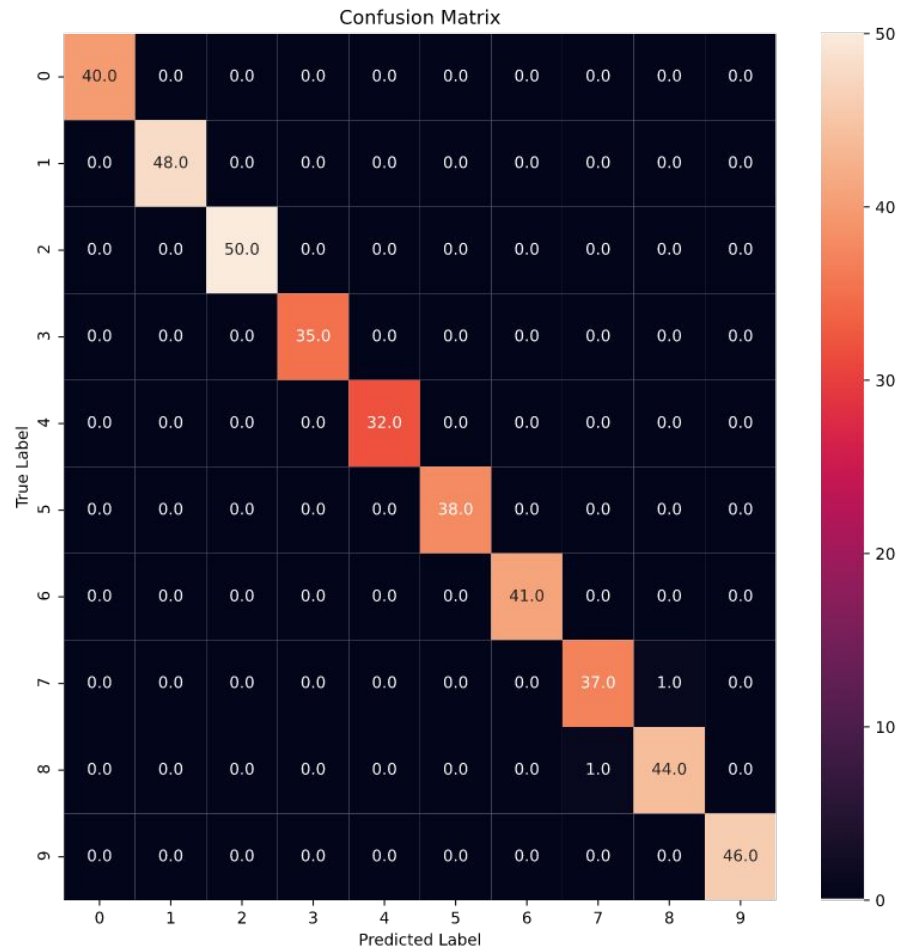
# Results



**7** → **8**



**8** → **7**



# Possible improvements

- More convolutional layers
- Optimize other parameters:
  - number of neurons in dense layer, dropout layers,...
- For real life use:
  - Better data augmentation (brightness, horizontal flip, wider zoom and rotation range,...)
  - Larger dataset with variable environmental conditions

Thank you for your attention