

SignLanguage Detection With CNN

Mohamed Mahmoud
Abdelrhman Ayman
Abdelrhman Hany

Computers and Information
Assiut University

April, 2024



① Task Description

② Contribution

③ Dataset

④ Live Demo

⑤ Methods

⑥ Results

1 Task Description

2 Contribution

3 Dataset

4 Live Demo

5 Methods

6 Results

SignLanguage Detection with CNN is a machine learning model trained with a convolutional neural network using a custom dataset to recognize some words in the American sign language such as ..help,movie,game,me,water,what

.

1 Task Description

2 Contribution

3 Dataset

4 Live Demo

5 Methods

6 Results

Abdelrhman Hany

- capturing the dataset for words
- preparing the labelmap
- creating the neural network and training the model
- Writing the presentation

Abdelrhman Ayman

- labeling the images to be trained
- creating the neural network and training the model
- preparing the landmarks

Mohamed Mahmoud

- Function to calculate angle between three points
- creating the neural network and training the model
- implementing the function of the realtime detection with openCV
- Writing the readme

1 Task Description

2 Contribution

3 Dataset

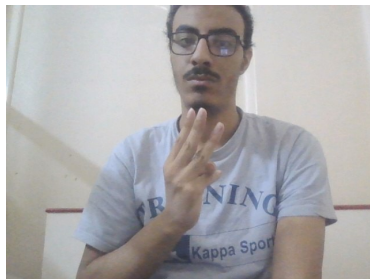
4 Live Demo

5 Methods

6 Results

Dataset

we capturing the images using opencv with the laptop camera



Dataset



1 Task Description

2 Contribution

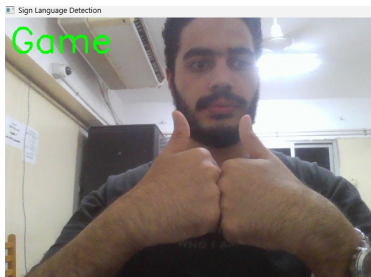
3 Dataset

4 Live Demo

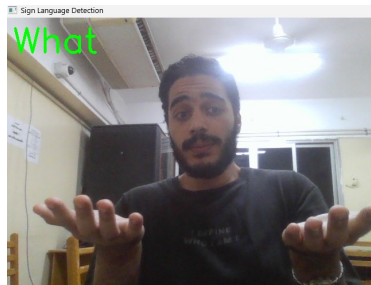
5 Methods

6 Results

Live Demo



Live Demo



1 Task Description

2 Contribution

3 Dataset

4 Live Demo

5 Methods

6 Results

Methods

in this model we used a sequential model from tensorflow.keras and we prepared 5 convolutional neural network layers then we start to train the network with our dataset we started with 25 epochs and we get a 83 accuracy and after some changes on the number of epochs and layers being used we get a 89.8 accuracy with 50 epoch and 5 layers and this is the description of layers

- the first layer is the input layer and it's a Dense Layer with 128 units and Relu activation function and it's job is Processing the input and applying ReLU activation.

- the second layer is a flatten layer and its job is to Convert multi-dimensional input into a 1D array.
- the third layer is a Dense Layer with 64 units and a Relu activation function and its job is Further processes the flattened input.
- the Fourth layer is a Dense Layer with 32 units and a Relu activation function and its job is Further processes the output from the previous dense layer.

- the last layer is the output layer and it's a Dense Layer with 6 units and Softmax activation function and it's job is Produces the final output probabilities for each class. This network structure is typical for a classification task where the input data is transformed through several layers of processing to eventually output a probability distribution over 6 classes.

1 Task Description

2 Contribution

3 Dataset

4 Live Demo

5 Methods

6 Results

Result

we got a 89.8 accuracy after training on approximately 148 training photo and 51 testing photo

```
5/5 [=====] - 0s 8ms/step - loss: 0.0493 - accuracy: 0.9793 - val_loss: 0.5350 - val_accuracy: 0.7755
Epoch 41/50
5/5 [=====] - 0s 7ms/step - loss: 0.0481 - accuracy: 0.9862 - val_loss: 0.5328 - val_accuracy: 0.7959
2/2 - 0s - loss: 0.3458 - accuracy: 0.8980 - 21ms/epoch - 10ms/step
Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...
[0.34576860070228577, 0.8979591727256775]
```