College name : Computer science and artificial intelligence

Course name : Selected CS-2

Team number : 55

Name (Arabic)	ID
سامح وليد محمد الشافعي	201900335
سعيد مليجي سيد مليجي	201900341
سعيد عماد ماهر محمد	201900339
زياد احمد عبدالتواب سعيد	201900322
ستيف عادل فاروق حبيب	201900337
محمد شريف فتحي محمد سالم	201900688

(Paper Details)

- Paper Name : OBJECT DETECTION USING CNN

- Publishers Name : Ms.Gunasundari , Lokesh , Gopirengaraj

-Year Of Publication : 2 April 2018

-The Implemented Algorithm : Convolution Neural Network (CNN) , Deep learning algorithm

- The Results : After implementation , a highest accuracy 98% has been gained using Kaggle dataset

(Project Description Document)

1) General Information on the selected dataset

- Name of the dataset used: American Sign Language

- The link of dataset: https://www.kaggle.com/datasets/kapillondhe/american-sign-language

- The total number of samples in the dataset : 121,608 samples

- Dimension of images (150 , 150)

- Number of classes & their labels : 27 classes with labels (from a to z and space sign)

- The ratio used for training, validation, and testing : Training (90% of the training dataset) = 97,200 images Validation (10 % of the training dataset) = 10,800 images ,

Testing (100% of the testing dataset) = 13608 images

2) Implementation details:

-The hyperparameters used in the model

```
my_model=Sequential()
my_model.add(Conv2D(64, kernel_size=7, strides=1, activation='relu', input_shape=[img_size,img_size,1]))
my_model.add(MaxPooling2D(pool_size=(2, 2)))
my_model.add(Conv2D(64, kernel_size=7, strides=2, activation='relu'))
my_model.add(Dropout(0.5))
my_model.add(Conv2D(256, kernel_size=3, strides=1, activation='relu'))
my_model.add(MaxPooling2D(pool_size=(2, 2)))
my_model.add(Conv2D(256, kernel_size=3, strides=2, activation='relu'))
my_model.add(MaxPooling2D(pool_size=(2, 2)))
my_model.add(Flatten())
my_model.add(Dropout(0.5))
my_model.add(Dense(512, activation='relu'))
my_model.add(Dense(27, activation='softmax'))
my_model.summary()
```

- model summery

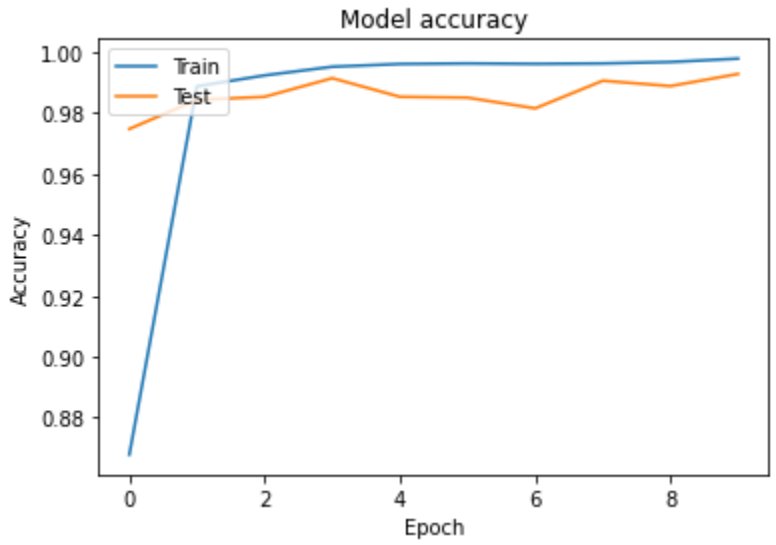
```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_4 (Conv2D)            (None, 144, 144, 64)      3200
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 72, 72, 64)        0
_________________________________________________________________
conv2d_5 (Conv2D)            (None, 33, 33, 64)        200768
_________________________________________________________________
dropout_1 (Dropout)          (None, 33, 33, 64)        0
_________________________________________________________________
conv2d_6 (Conv2D)            (None, 31, 31, 256)       147712
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 15, 15, 256)       0
_________________________________________________________________
conv2d_7 (Conv2D)            (None, 7, 7, 256)         590080
_________________________________________________________________
max_pooling2d_4 (MaxPooling2 (None, 3, 3, 256)         0
_________________________________________________________________
flatten (Flatten)            (None, 2304)              0
_________________________________________________________________
dropout_2 (Dropout)          (None, 2304)              0
_________________________________________________________________
dense (Dense)                (None, 512)               1180160
_________________________________________________________________
dense_1 (Dense)              (None, 27)                13851
=================================================================
Total params: 2,135,771
Trainable params: 2,135,771
Non-trainable params: 0
```

- running epochs details

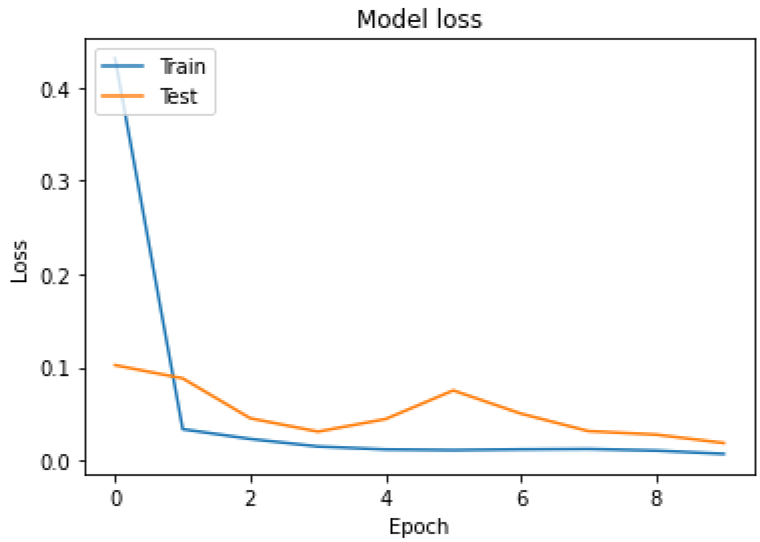
```
Epoch 1/10
380/380 [==============================] - 303s 765ms/step - loss: 0.4311 - accuracy: 0.8679 - val_loss: 0.1023 - val_accur
acy: 0.9748
Epoch 2/10
380/380 [==============================] - 225s 592ms/step - loss: 0.0336 - accuracy: 0.9888 - val_loss: 0.0881 - val_accur
acy: 0.9844
Epoch 3/10
380/380 [==============================] - 204s 536ms/step - loss: 0.0231 - accuracy: 0.9924 - val_loss: 0.0453 - val_accur
acy: 0.9854
Epoch 4/10
380/380 [==============================] - 212s 558ms/step - loss: 0.0150 - accuracy: 0.9952 - val_loss: 0.0310 - val_accur
acy: 0.9915
Epoch 5/10
380/380 [==============================] - 217s 570ms/step - loss: 0.0118 - accuracy: 0.9961 - val_loss: 0.0445 - val_accur
acy: 0.9854
Epoch 6/10
380/380 [==============================] - 217s 570ms/step - loss: 0.0111 - accuracy: 0.9963 - val_loss: 0.0752 - val_accur
acy: 0.9851
Epoch 7/10
380/380 [==============================] - 211s 555ms/step - loss: 0.0119 - accuracy: 0.9962 - val_loss: 0.0503 - val_accur
acy: 0.9816
Epoch 8/10
380/380 [==============================] - 213s 560ms/step - loss: 0.0123 - accuracy: 0.9963 - val_loss: 0.0315 - val_accur
acy: 0.9906
Epoch 9/10
380/380 [==============================] - 211s 556ms/step - loss: 0.0106 - accuracy: 0.9968 - val_loss: 0.0278 - val_accur
acy: 0.9889
Epoch 10/10
380/380 [==============================] - 214s 563ms/step - loss: 0.0071 - accuracy: 0.9979 - val_loss: 0.0188 - val_accur
acy: 0.9929
```

3) Results details:

- The Accuracy : 99.29%



- The loss : 1.88%



- The Confusion Matrix