

## Selected topics in computer science -2

Team : 22

Name	ID
Salah Eldin kamel Mahmoud	202000467
Diaa hany Abdel monem	202000475
Abdelrahman mohamed zahran	202000475
Mariiam amr fawzy mohamed	202000881
Mariiam mohamed mowad	202000885
Hisham mohamed ahmed hefny	202001035

## 1-Paper details

### A. Citation

Author name	I Khandokar <sup>1</sup> , M Hasan Md <sup>1</sup> , F Ernawan <sup>1</sup> , S Islam Md <sup>1</sup> and M N Kabir <sup>1</sup>
Paper name	Handwritten character recognition using cnn
Publisher name	Published under licence by IOP Publishing Ltd
Year of publication	2021

### B. Dataset used

To test the HCR system

**NIST** database has been used. To assess the performance of CNN algorithm, we experimented with the dataset NIST and found the accuracy of handwritten characters. The handwritten characters in NIST are given as images. The images are split into training and testing sets. Training is carried out with various number of images; then testing is conducted to find the accuracy of the CNN.

### C. The implemented algorithm

The implemented algorithm is the the CNN architecture algorithm ,Three types of layers used the conv layer , pooling and finally the fully connected and the training worked on 4 stages [pre-processing -segmentation -Feature extraction -Training and prediction]

### D. The result

The result of this algorithm is a text formed from the input image which contains the characters

## 2-Project description

### A.Information on the selected dataset :

DatasetName	Handwriting Recognition Handwritten characters
Link of the dataset	<a href="https://www.kaggle.com/datasets/landlord/handwriting-recognition?datasetId=818027">https://www.kaggle.com/datasets/landlord/handwriting-recognition?datasetId=818027</a> <a href="https://www.kaggle.com/datasets/vaibhao/handwritten-characters">https://www.kaggle.com/datasets/vaibhao/handwritten-characters</a>
Total number of samples	<ol style="list-style-type: none"> <li>1. Train samples are about 600k image</li> <li>2. Validation samples = 270k images</li> <li>3. Test samples = 380k image</li> </ol>
Dimension of image	(32*32) grey scale
Classification	25 class from (a~z) both lowercase and uppercse

### B.Implementation details

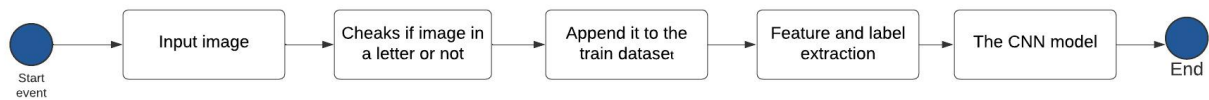
- a. The ratio user for training is : **133218 90%**
- b. The ratio of validation is : **6651 10 %**
- c. Testing on names dataset : **388k images**

## The implemented model :

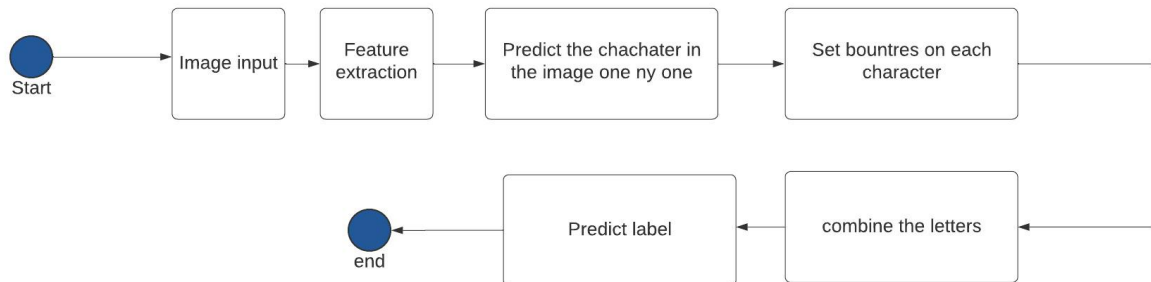
### object detection

Mariam Amr | May 4, 2023

#### 1-The train model



#### 2-Prediction model

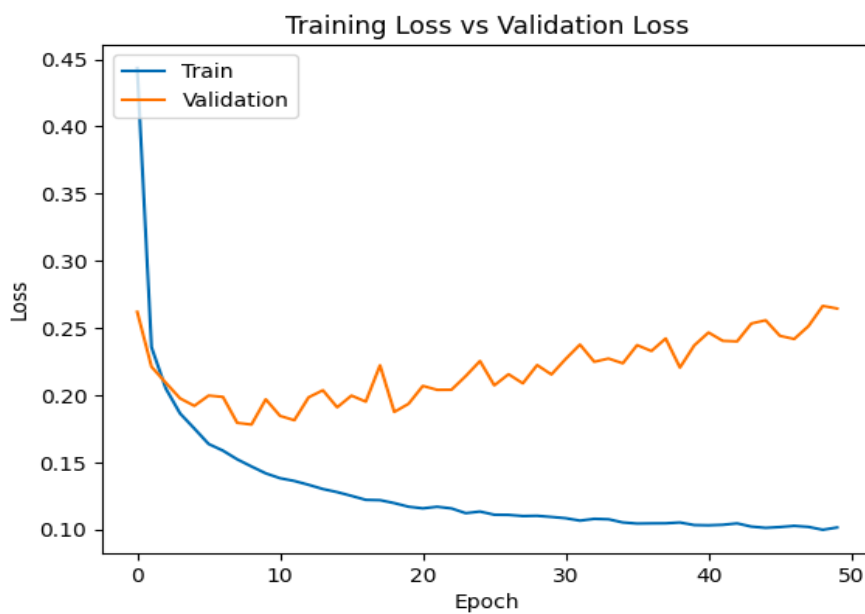


- \*Hyperparameters :**
- 1-Number of kernels is 32
  - 2-Size of kernel is (3\*3)
  - 3-size of kernel in pooling is (2\*2)

## C.Results

### \*Loss [Training and validation curve]

```
[95] plt.plot(history.history['loss'])  
     plt.plot(history.history['val_loss'])  
     plt.title('Training Loss vs Validation Loss')  
     plt.ylabel('Loss')  
     plt.xlabel('Epoch')  
     plt.legend(['Train', 'Validation'], loc='upper left')  
     plt.show()
```

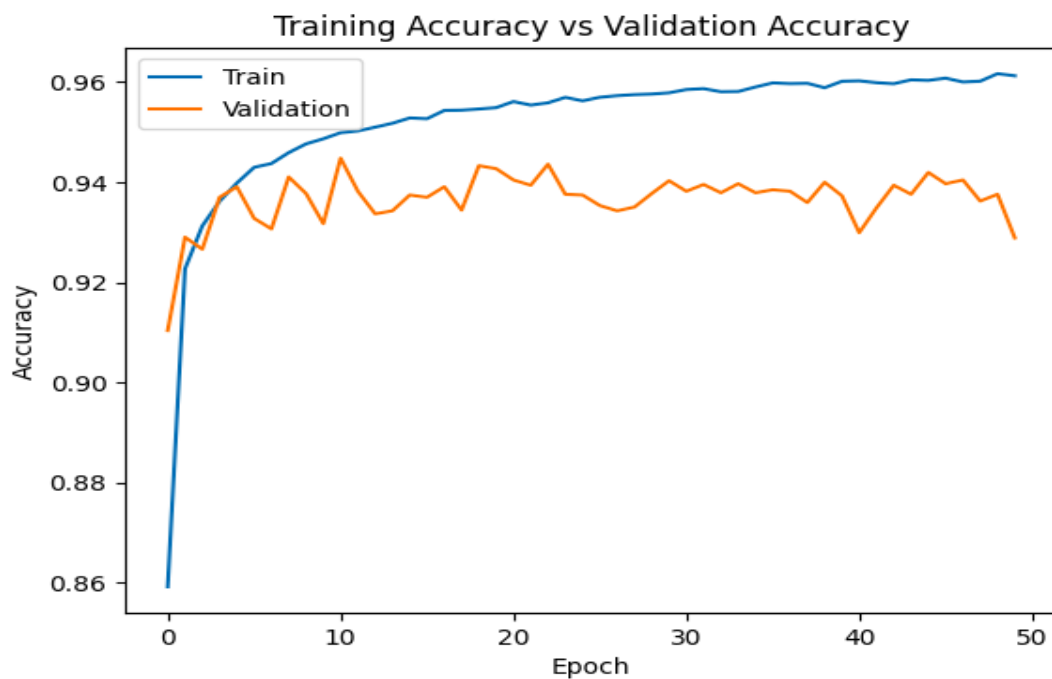


## \*Accuracy [Training and validation ]

```

1s ✓ ▶ plt.plot(history.history['accuracy'])
plt.plot(history.history['val_accuracy'])
plt.title('Training Accuracy vs Validation Accuracy')
plt.ylabel('Accuracy')
plt.xlabel('Epoch')
plt.legend(['Train', 'Validation'], loc='upper left')
plt.show()

```



## \*Result sample

```

1s ✓ [102] letter,image = get_letters("/content/train_v2/train/TRAIN_00030.jpg")
word = get_word(letter)
print(word)
plt.imshow(image)

```

```

1/1 [=====] - 0s 22ms/step
1/1 [=====] - 0s 19ms/step
1/1 [=====] - 0s 21ms/step
1/1 [=====] - 0s 20ms/step
1/1 [=====] - 0s 21ms/step
1/1 [=====] - 0s 20ms/step
1/1 [=====] - 0s 21ms/step

```

JAFFEUX

<matplotlib.image.AxesImage at 0x7fc0c257dc60>

