AI Course

Team Project Final Report

For students (instructor review required)

ⓒ2019 SAMSUNG. All rights reserved.

Samsung Electronics Corporate Citizenship Office holds the copyright of this document.

This document is a literary property protected by copyright law so reprint and reproduction without permission are prohibited.

To use this document other than the curriculum of Samsung innovation Campus, you must receive written consent from copyright holder.

| Sign Language Classification Using Deep Learning |
| --- |

May 11 2022

Team: Vision of Sauron

Abel Asfaw

Hilarion Reyes

Mosunmola Oyeleye

Mya Thanegi Soe

Giovan Panzanella

Content

1. Introduction

1.1. Background Information

1.2. Motivation and Objective

1.3. Members and Role Assignments

1.4. Schedule and Milestones

2. Project Execution

2.1. IoT Service Model

2.2. Data Processing

2.3. Service Implementation

2.4. System Diagram

3. Results

3.1. Data Acquisition (Sensor, Actuator, Controller)

3.2. Network & Communication

3.3. Hardware Implementation

3.4. Data Visualization

3.5. Testing and Improvements

4. Projected Impact

4.1. Accomplishments and Benefits

4.2. Future Improvements

5. Team Member Review and Comment

6. Instructor Review and Comment

1. Introduction

1.1. Background Information

* computer vision project to detect sign language letters
* Group name: Vision of Sauron

1.2 Motivation and Objective

* The deaf or people with hearing problems don’t have efficient mobile or web applications that can be used to communicate
* Current visual recognition algorithms perform poorly with real world applications
* American Sign Language is a complex language and primary language for deaf people.

1.3 Members and Role Assignments

* Mosunmola Oyeleye: load files, implement “get train + test + validation” from table data, load model for openCV in google colab
* Mya Thanegi Soe: load model for opencv locally, design action plan
* Giovan Panzanella: image preprocessing, openCV image post processing (for webcam demo)
* Record model demo video while using openCV:
* Abel Asfaw: Data analyst, train resnet + custom cnn model (28,28,1 images)
* Hilarion Reyes: train custom cnn model (SGD, 128,128,1 images)

1.4 Schedule and Milestones

* ~~4/212022: Finished Action Plan~~
* ~~4/25/2022: Finalized Work Breakdown Structure~~
* ~~5/10/2022: Finalized Google Colab code file~~
* ~~5/10/2022: Finalized model demo video~~
* ~~5/12/2022: Finalized Capstone Final Report~~
* ~~5/12/2022: Finalized Pitch Presentation~~

2. Project Execution

2.1 Data Acquisition

* Sign MNIST Kaggle Dataset
* Training and test set contain a label (0-25) and letters (A-Z)
* Match the patterns of the MNIST dataset
* Each pixel is of size 28\*28
* Each sample has 784 pixels
* Letters J and Z excluded (required motion)

2.2 Training Methodology

* customized RMSprop cnn model
* custom SGD cnn model
* transfer learning: RESNET

2.3 Workflow

* Load files on colab
* Data Analysis
* SGD model training and results
* RESNET model training and results
* Custom CNN model training and results
* open cv testing locally on pc
* open cv testing on Google Collab
* analyze losses, accuracy and classification report

2.4 System Diagram



3. Results

3.1. Data Preprocessing

* Upload Train and Test files on google collab (faster data-loading)
* obtain validation set (from Train)
* resize image to speed up processing
* normalize images (divide by 255)

3.2 Exploratory Data Analysis (EDA)

* Highest count is Q
* 0 – 25 is mapped to A-Z
* ***J and Z*** require motion
* There is no count for 9 = J and 25 = Z

3.3 Modeling

* sign\_train.csv file used for training
* sign\_test.csv file used for testing
* split some of the training data to build the training set and validation set.
* Built a custom SGD model using Convolution2D Layers and Dense Layers with a total of 1,526,425 trainable params.
  + Used Max-pooling, Dropout, and Decay for Regularization..
* Used a Pre trained model Resnet by applying transfer learning
  + Used Imagenet as a database for visual recognition to train the resnet model
* Built a custom cnn model to train data
  + used RMSprop(learning\_rate=1e-4), MaxPooling and Dropout

3.4 User Interface (Interface).

* openCV:
  + get webcam data from live computer webcam
  + outline bounding box
  + outline model’s top 3 results + model prediction confidence.

3.5. Testing and Improvements.

* Expand diversity of dataset by getting different individuals to model letter types (excluding J and Z which require motion).
  + potential factors: size of hand, colour of hand
* Implement Data Augmentation (rotations, flipping, etc..) to improve model robustness .
* Increase image sizes to improve training generalization and improve live-testing accuracy.
* Continue fine-tuning the Hyperparameters and the complexity of the hidden layers in order to optimize models’ performance.



4. Projected Impact

4.1. Accomplishments and Benefits

* successfully predicted some letters like B, K , U with ~97% prediction

4.2 Future Improvements

* collect more data (augment data, diversify data)
* utilize more algorithms like inception or vgg to get better results
* Try different scaling measurements to make capturing signs more easier

5. Team Member Review and Comment 

| N/A |
| --- |

| NAME | REVIEW and COMMENT |
| --- | --- |
| Abel Asfaw |  |
| Hilarion Reyes |  |
| Mosunmola Oyeleye |  |
| Mya Thanegi Soe |  |
| Giovan Panzanella |  |

6. Instructor Review and Comment

| CATEGORY | SCORE | REVIEW and COMMENT |
| --- | --- | --- |
| IDEA | \_\_/20 |  |
| CODING | \_\_/20 |  |
| PROJECT MANAGEMENT | \_\_/30 |  |
| PRESENTATION & REPORT | \_\_/30 |  |
| TOTAL | \_\_/100 |  |