



North South University

Department of Electrical & Computer Engineering

Topic: Lips Movement Detection

Submitted By:

Name: Tasnim Shahrin. (202 2025 042)

Abu Mukaddim Rahi. (202 2027 042)

Section: 09

Faculty: MUO

Course Code: CSE299

Spring 2023.

Submitted To:

Muhammad Shafayat Oshman

Submission Date: 20.06.2023

| Table of Contents | Page no. |
|---|-----------------|
| Abstract | 03 |
| Introduction | 04 |
| Problem Statement | 05 |
| Project Description | 06 |
| Project Flowchart | 09 |
| Ghant Chart | 10 |
| Technology used in project | 11 |
| Cost Analysis | 12 |
| Conclusion | 14 |
| Source & References | 16 |

Abstract

Lip movement detection is a computer vision technique that finds and tracks the movement of the lips in a video or image. This technology has a wide range of applications, including speech recognition, lip reading, and facial expression recognition.

The goals of this project are to: Develop a LIPS-MOVEMENT-DETECTION-model-app that can detect and track lip movement in real time. Train a deep learning model for lip movement detection. Evaluate the performance of the app on a variety of datasets.

The results of this project will be a LIPS-MOVEMENT-DETECTION-model-app that can be used to detect and track lip movement in real time. The app will be a valuable tool for researchers and developers working on speech recognition, lip reading, and facial expression recognition.

Introduction

The field of computer vision and artificial intelligence has seen remarkable advancements in recent years, opening new avenues for innovation and research. One such area of interest is the development of lip movement detection models, which can have many applications in fields like speech recognition, emotion analysis, and even human-computer interaction. In this junior project report, we present the design and implementation of the LIPS-MOVEMENT-DETECTION Model App, an innovative application that uses computer vision techniques to detect and analyze lip movements in real-time.

The LIPS-MOVEMENT-DETECTION Model App supplies a user-friendly interface for real-time lip movement analysis. The app processes the video frames, isolates the lip region, and tracks its movements throughout the video. Additionally, the app offers various analysis features, such as estimating speech patterns, showing emotional expressions, and even lip-reading capabilities, thus displaying the versatility of the implemented model.

This project's outcomes are significant, as they pave the way for further research and development in lip movement detection. By harnessing the power of computer vision and deep learning, our LIPS-MOVEMENT-DETECTION Model App offers a practical tool for diverse applications, ranging from enhancing communication systems to helping individuals with speech impairments.

Problem Statement

The need for an efficient and exact system that can detect and interpret lip movements in real-time is the problem we hope to address with the development of the LIPS-MOVEMENT-DETECTION model app. This project aims to bridge the gap between speech recognition and visual cues by detecting and analyzing lip movements for various applications using computer vision techniques.

Individuals with hearing impairments face difficulties communicating effectively with others who do not understand sign language or are not skilled at reading lip movements in terms of accessibility and inclusivity. Conventional methods of communication, such as relying solely on sign language interpreters or text-based communication, may not always be available or practical. This project aims to empower people with hearing loss by providing them with a tool that can accurately interpret lip movements in real time, allowing them to communicate more effectively with the rest of the community.

While speech recognition technology has advanced, it still has limitations in noisy environments or situations where audio input is not available or clear. The LIPS-MOVEMENT-DETECTION model app aims to augment existing speech recognition systems by using computer vision algorithms to track and interpret lip movements. The app aims to improve the accuracy and reliability of speech recognition, particularly in difficult conditions, by combining visual and audio input, thereby improving the overall user experience.

By addressing the social problem of communication barriers faced by individuals with hearing impairments and the technical problem of enhancing speech recognition accuracy, the LIPS-MOVEMENT-DETECTION model app strives to provide an innovative solution that promotes inclusivity and helps effective communication for people with hearing impairments.

Project Description

In this project, we designed a machine learning model and application which can predict our lips movement detection.

So, at the beginning we designed and trained our lips-reading detect model. We trained our model in Jupiter Notebook; first we installed and imported all dependencies what we need to train our model properly. Then we build our data loading function functions. We are going to train a model to be able to decode this from purely a video with no audio so still means silent so. The function called load_video is going to take a data path and then it is going to output a list of floats that are going to stand for our video. CV2 instance a video capture instance which takes in going to loop through each one of these frames and store it inside of an array called frames.

Then we created data pipelining in our model. Data pipelining is essential for organizations to streamline data integration, transformation, synchronization, scalability, efficiency, data governance, and automation. It enables organizations to make better-informed decisions based on reliable, prompt, and well-processed data. For designed the Deep Neural Network we need to import necessary libraries and functions from the TensorFlow Keras API. Then we need to defines a neural network model using the Sequential API of TensorFlow Keras.

We use Adam optimizer (the Adam optimizer is an adaptive learning rate optimizer that computes individual adaptive learning rates for different parameters from estimates of first and second moments of the gradients). Train the model using the training set train and evaluate the model on the validation set test. The model will be trained for 100 epochs. The training data is being processed in batches of size 450.

For make a prediction in our model we download a file from Google Drive using the gdown library, extracts its contents, and saves them to a directory called models. The zip file holds pre-trained modelcheckpoints that can be loaded and used to make predictions. By downloading and extracting the checkpoints, the code can quickly load the pre-trained model and use it for inference without having to train the model from scratch. Sets up the model for training using a legacy optimizer, compiles the model with the binary cross-entropy loss function, and

loads the weights from a checkpoint file for further training or evaluation. By generating predictions using the trained neural network, the accuracy, and performance of the model can be evaluated on the test data. This can help identify any issues with the model or training process and can guide future improvements to the model.

Then we debug and evaluate the performance of the neural network on the test data. By printing the original text for each test sample, it can be easier to identify any errors or discrepancies in the output generated by the neural network. The accuracy of the neural network can also be evaluated by comparing the predicted text to the original text for the test data.

We built up a machine learning app using a deep learning model we had made. We create "modelutil.py" which could contain functions that assist with tasks such as loading and saving machine learning models, preprocessing input data, or evaluating model performance. We created a "streamlitapp.py" refers to a Python script or file that is used to create a web application using the Streamlit framework. Streamlit is especially well-suited for data science and machine learning applications, as it provides a number of built-in tools for working with data, visualizing results, and running machine learning models.

(Streamlit web application can be updated by modifying the code in the "streamlitapp.py".Streamlit allows developers to create interactive web applications using Python scripts). We set up a Streamlit app with a sidebar, video selection, and displays the input video, model output, and converted prediction.

streamlitapp - Streamlit

localhost:8501

App


Choose video

bbal8p.mpg

The video below displays the converted video in mp4 format

0:00 / 0:03

This is all the machine learning model sees when making a prediction




This is the output of the machine learning model as tokens

```
[[ 2  9 14 39  2 12 21  5 39  1 20 39 12 39
 19  5  0  0  0  0  0  0  0  0  0  0  0  0
 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -
 -1 -1 -1]]
```

Decode the raw tokens into words

bin blue at l eight please

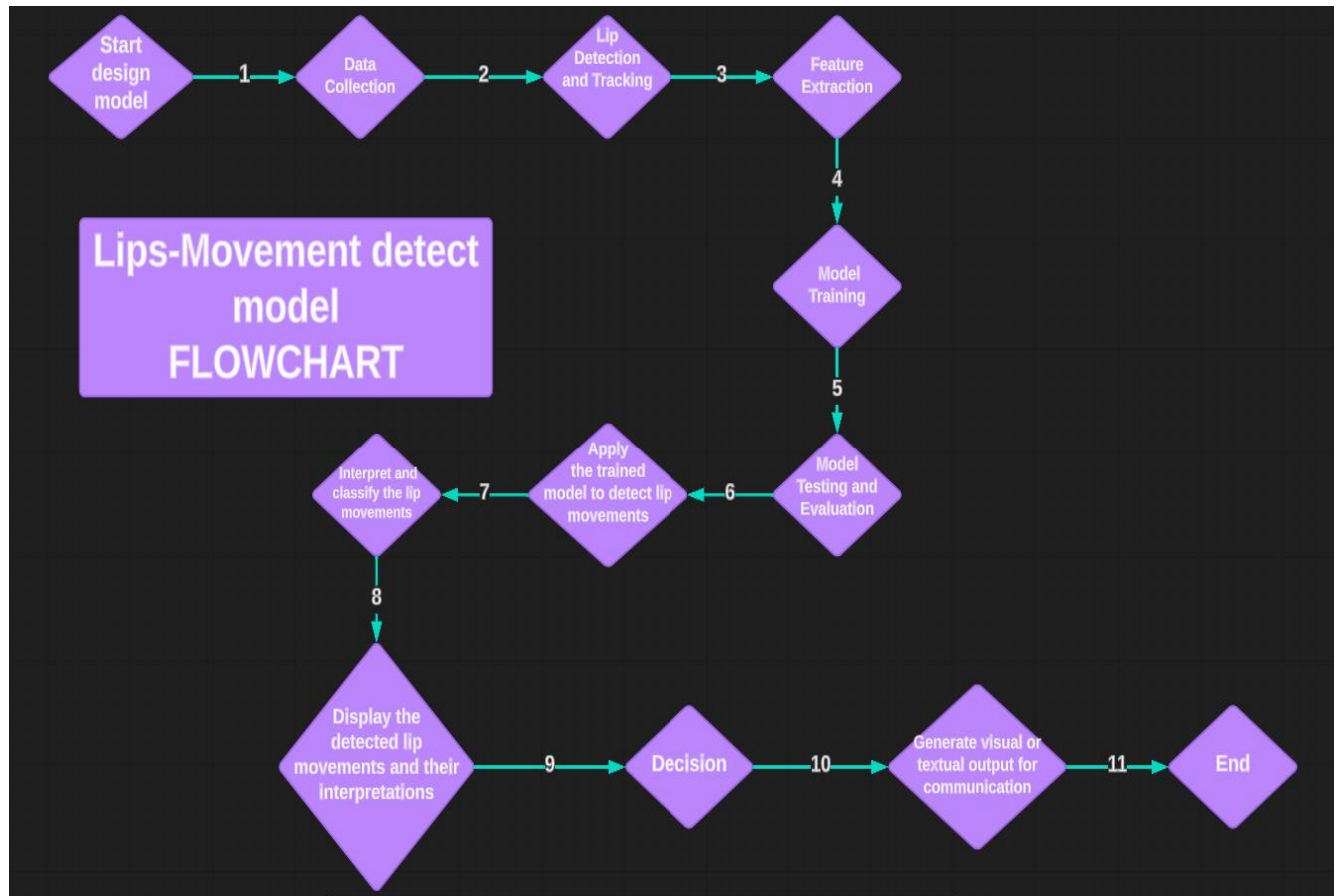
CSE299.9 Junior Design Project Spring-2023



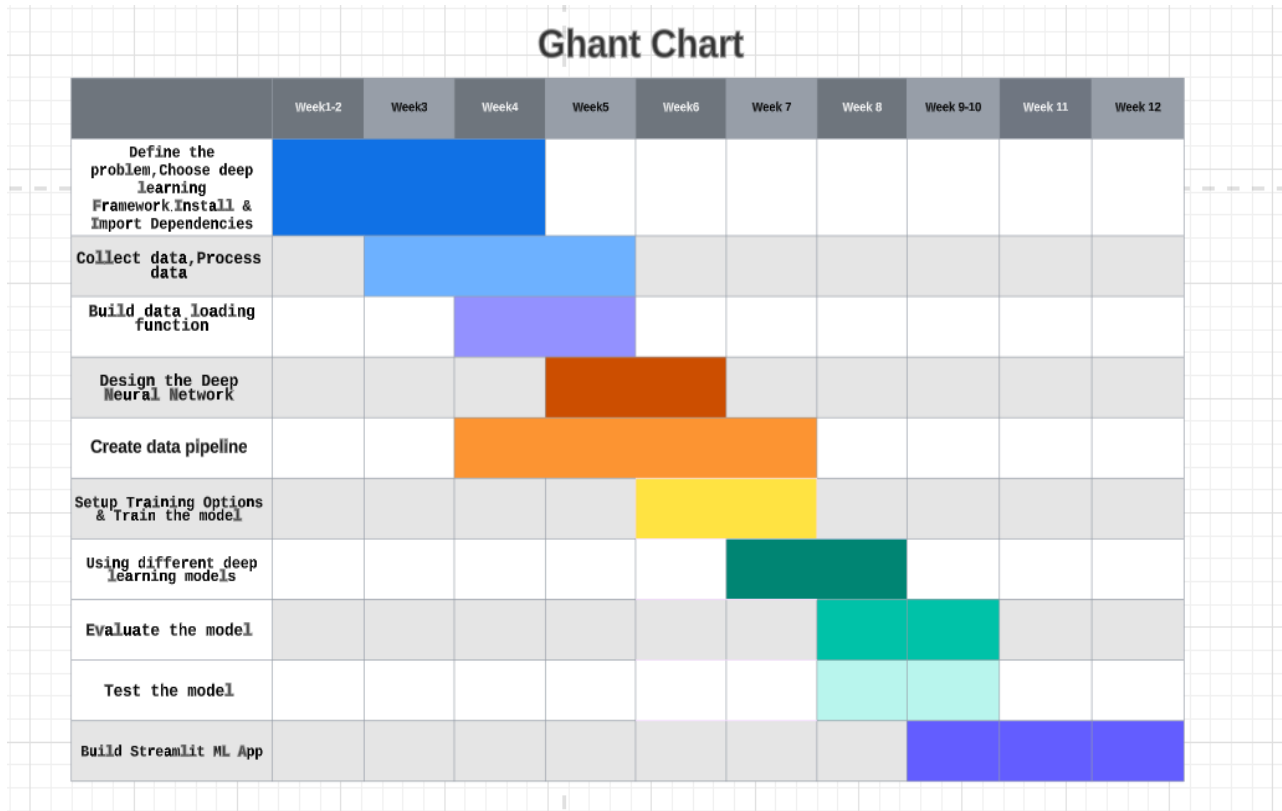
Lips Movement Detection

Our junior design project CSE299 Lip_Movement_Detection uses computer vision and deep learning to understand speech or non-verbal communication.

Project Flowchart



Ghant Chart



Technology Used in this project

This project is totally based on Software (Python, Jupyter Notebook, Tensorflow 2.0, Keras, Visual Studio). In a project focused on deep learning-based lip movement detection, several technologies and techniques can be used. Here are some key components commonly employed in such projects:

1. Convolutional Neural Networks (CNNs)
2. Deep Learning Frameworks: Frameworks like TensorFlow, PyTorch, or Keras provide high-level abstractions and tools to implement and train deep learning models efficiently.
3. Data Augmentation: To enhance the generalization and robustness of the lip movement detection model, techniques such as image rotation, scaling, cropping, and flipping can be used to create additional training examples.
4. Transfer Learning: Pretrained models, such as those trained on large-scale image datasets like ImageNet, can be used as a starting point. By fine-tuning these models on lip movement data, it is possible to leverage their learned features and accelerate the training process.
5. Data Preprocessing
6. Loss Functions: The choice of loss functions depends on the specific task. For example, binary cross-entropy loss can be used for binary classification (e.g., open/closed mouth), while categorical cross-entropy loss may be suitable for multi-class classification tasks (e.g., phoneme recognition).
7. GPU Acceleration
8. Deployment (Using Streamlit to design the ML App)

Cost Analysis

Constructive Cost Model:

Project Type: Organic

Coefficient<Effect Factor>: 2.4 [P=1.05; T=0.38]

SLOC = 6000 Lines

Person Months, PM = $(2.4 * 6^{1.05}) = 15.74$

Duration Time, DM = $(2.5 * 15.74^{0.38}) = 7.12 = 7$ months = 8,400 working hours
=1050 working days

Required People, ST = PM/DM = 2.24 = 2 people

Budgeting:

Developer salary in 7 Months: Per developer salary per working days = 300 taka

Total developer salary = $300 * 1050 * 4 = 1260000$ taka

Requirement Analysis:

Time needed: 1 month (22 working days = 154 working hours)

Hourly wage for requirement analysis = 37.5 tk per hour and per day 300tk

Total Requirement Analysis Expense = $300 * 154 = 46200$ tk

Transportation cost estimation: 10,000tk

Training & hardware Expenses Estimation: 100,000tk

Rent Expenses:

Room per month = 10,000 taka

Total in 7 Months = 70,000 taka

Total utilities in 7 Months: 20,000 taka

Maintenance (Till 6 months after delivery):

Expense per hour = 1000 taka

Total Estimated Time needed for maintenance = 60 Hours

Total Estimated Maintenance Expense = $60 * 1000 = 60,000$ taka

Total Estimated Expense: $1260000 + 46200 + 10,000 + 100,000 + 70,000 + 20,000 + 60,000 = 1,566,200$ tk

Profit:

25% of total Estimated Expense = $1,566,200 * 25\% = 39,1550$ taka

Project Budget: $1,566,200 + 39,1550 = 19,57,750$ taka

Conclusion

This project focused on deep learning-based lip movement detection, a challenging and important task in computer vision and human-computer interaction. The objective was to develop a model capable of accurately detecting and analyzing lip movements from video or image sequences.

Throughout the project, several technologies and techniques were employed to achieve this goal. Convolutional Neural Networks (CNNs) were used to extract relevant visual features from the lip region.

Data augmentation techniques, such as image rotation, scaling, cropping, and flipping, were applied to enhance the model's generalization and robustness. Additionally, transfer learning was utilized by leveraging pretrained models to expedite the training process and benefit from their learned features.

The project also required diligent data preprocessing, including frame extraction, resizing, normalization, and potentially applying filters to optimize the input data for training the deep learning model. Various loss functions were explored, depending on the specific task, such as binary cross-entropy for binary classification or categorical cross-entropy for multi-class classification.

Furthermore, the utilization of GPU acceleration significantly accelerated the training and inference processes, allowing for efficient experimentation and model development.

Overall, the developed deep learning-based lip movement detection model shows promise in accurately analyzing and classifying lip movements. However, it is important to note the model's performance is contingent on the quality and diversity of the training data and the availability of labeled datasets tailored for lip movement detection.

Future work could involve expanding the dataset to incorporate more variations in lip movements, exploring advanced deep learning architectures or novel techniques, and further optimizing the model for real-time lip movement detection applications and try to do this model in pure Bangla dataset. Additionally, integrating the developed model into practical applications, such as human-

computer interfaces, speech recognition systems, or emotion recognition systems, could lead to exciting possibilities in enhancing human-machine interaction.

In conclusion, this project contributes to the advancement of deep learning techniques for lip movement detection, providing a foundation for further research and application development in computer vision and human-computer interaction.

Source Code:

<https://github.com/Tas890/LIPS-MOVEMENT-DETECTION-model-app>

References:

1.

S. -j. Lee, J. Park and E. -k. Kim, "Speech Activity Detection with Lip Movement Image Signals," 2007 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing, Victoria, BC, Canada, 2007, pp. 403-406, doi: 10.1109/PACRIM.2007.4313259.

2.

https://www.researchgate.net/publication/352429113_Lip_Movement_Feature_Detection_and_Classification_Methods/citations

3.

<https://www.ijstr.org/final-print/aug2019/Lip-Feature-Extraction-And-Movement-Recognition-Methods-A-Review.pdf>

4. https://en.wikipedia.org/wiki/Lip_reading

THE END

