Republic of Tunisia
Ministry of Higher Education and Scientific Research
University of Tunis El Manar
National School of Engineers of Tunis

Computer Vision Project

LipReading:
From Visemes to Phonemes:
The Intersection of Vision and Language

Realised by :
Trigui Hatem
Hassouna Malek
Med Rabii Baccari

Supervised by :
Ms. Linda Marrakchi

Class : 3ATEL.DASEC

01 Introduction

02 State of the art

03 Applying Traditional Approaches For Lip Reading

04 Sentence Level Enhancement using LipNet

05 Conclusion

Introduction

Problem Statement

Deafness and Hearing Loss

- Over **5%** of the world's population (**430 million people**)
- By 2050, this number is expected to rise to over **700 million people**, or 1 in every 10 individuals globally.
- Approximately **30%** of people **over 60 years** of age experience hearing loss.

Speech Disorders

Studies indicate that **15.3% of children** may suffer from speech disorders.

Soure : National Institute on Deafness and Other Communication Disorders (NIDCD)

What is Lip Reading ?

Phonemes vs Visemes:

- Phonemes are the smallest units of sound in a language
- Visemes are the visual counterparts of phonemes

Phoneme Overlap and Coarticulation

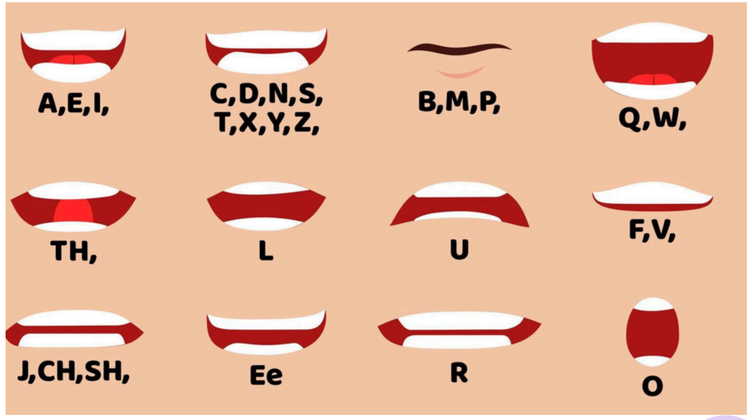
- Syllable Superposition
- Temporal and Spatial Overlaps
- Variability Across Speakers

s sat	t tap	p pan	n nose	m mat	a ant	e egg	i ink	o otter
g goat	d dog	c k click	r run	h hat	u up	ai rain	ee knee	igh light
b bus	f farm	l lolly	j jam	v van	oa boat	oo cook	oo boot	ar star
w wish	x axe	y yell	z zap	qu quill	or fork	ur burn	ow now	oi boil
ch chin	sh ship	th think	th the	ng sing	ear near	air stair	ure sure	er writer

Decoding speech from the movement of a speaker's mouth.

Viseme Ambiguity

- Ambiguity in Speech Recognition
- Coarticulation Effects
- Contextual Dependencies



Approaches of Lip Reading Tasks

Traditional Approaches: Two-Stage Process

Visual Features

Designing or learning representations from lip movements.

Pediction

Mapping these features to text or speech.

Modern Deep Learning Approaches

Evolution of LipReading Approaches

2000 - Traditional approaches and starting to integrate machine learning

2010 - Transition Towards Deep Learning

2015 -

2016 - End-to-End Deep Learning Revolution

now -

State of the art

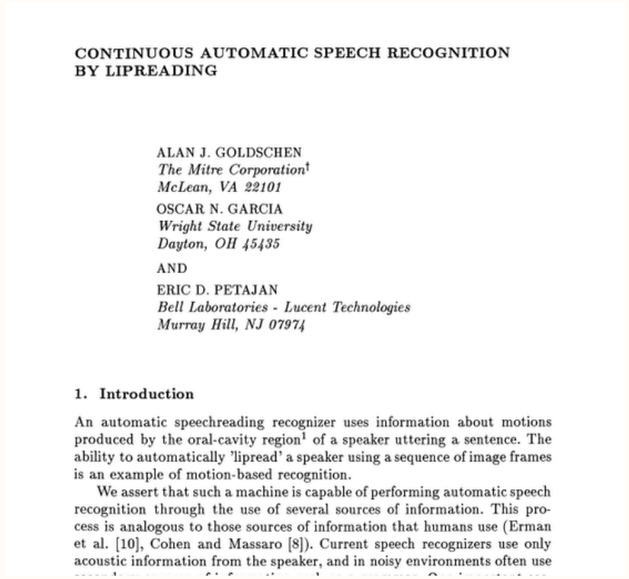
Automated lipreading

1997 — 2000 — .... — 2016

Continuous Automatic Speech Recognition by Lipreading

Goldschen et al.

visual-only sentence-level by using **Hidden Markov Models (HMMs)** on a limited dataset with hand-segmented phones.



CONTINUOUS AUTOMATIC SPEECH RECOGNITION
BY LIPREADING

ALAN J. GOLDSCHEN
The Mitre Corporation[1]
McLean, VA 22101
OSCAR N. GARCIA
Wright State University
Dayton, OH 45435
AND
ERIC D. PETAJAN
Bell Laboratories - Lucent Technologies
Murray Hill, NJ 07974

1. Introduction

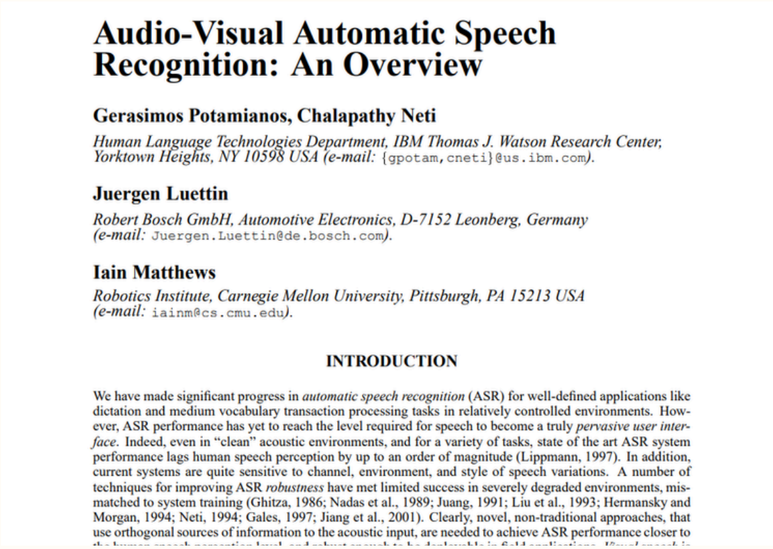
An automatic speechreading recognizer uses information about motions produced by the oral-cavity region[1] of a speaker uttering a sentence. The ability to automatically 'lipread' a speaker using a sequence of image frames is an example of motion-based recognition.

We assert that such a machine is capable of performing automatic speech recognition through the use of several sources of information. This process is analogous to those sources of information that humans use (Erman et al. [10], Cohen and Massaro [8]). Current speech recognizers use only acoustic information from the speaker, and in noisy environments often use

Audio-Visual Automatic Speech Recognition

Neti et al.

first **sentence-level audiovisual speech recognition** using an HMM combined with hand-engineered features, on the IBM ViaVoice dataset.



Audio-Visual Automatic Speech Recognition: An Overview

Gerasimos Potamianos, Chalapathy Neti

Human Language Technologies Department, IBM Thomas J. Watson Research Center, Yorktown Heights, NY 10598 USA (e-mail: {gpotam,cneti}@us.ibm.com*).*

Juergen Luettin

Robert Bosch GmbH, Automotive Electronics, D-7152 Leonberg, Germany (e-mail: Juergen.Luettin@de.bosch.com*).*

Iain Matthews

Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213 USA (e-mail: iainm@cs.cmu.edu*).*

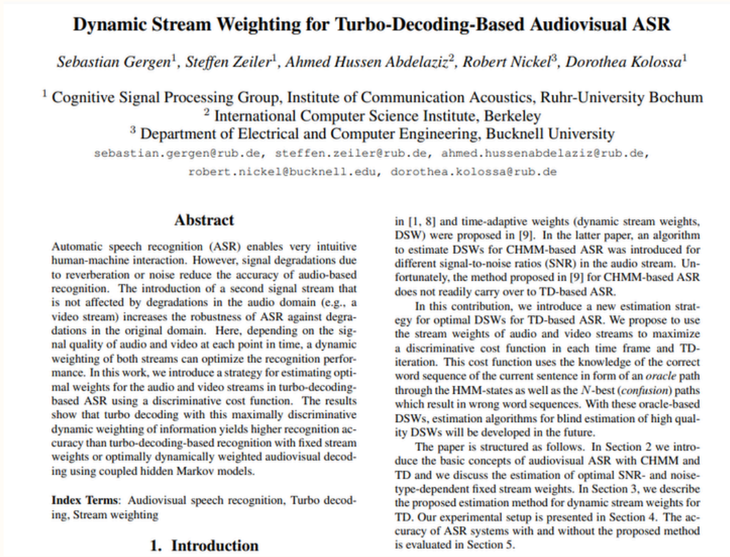
INTRODUCTION

We have made significant progress in *automatic speech recognition* (ASR) for well-defined applications like dictation and medium vocabulary transaction processing tasks in relatively controlled environments. However, ASR performance has yet to reach the level required for speech to become a truly *pervasive user interface*. Indeed, even in "clean" acoustic environments, and for a variety of tasks, state of the art ASR system performance lags human speech perception by up to an order of magnitude (Lippmann, 1997). In addition, current systems are quite sensitive to channel, environment, and style of speech variations. A number of techniques for improving ASR *robustness* have met limited success in severely degraded environments, mismatched to system training (Ghitza, 1986; Nadas et al., 1989; Juang, 1991; Liu et al., 1993; Hermansky and Morgan, 1994; Neti, 1994; Gales, 1997; Jiang et al., 2001). Clearly, novel, non-traditional approaches, that use orthogonal sources of information to the acoustic input, are needed to achieve ASR performance closer to

Dynamic Stream Weighting for Turbo-Decoding-Based Audiovisual ASR

Gergen et al.

use speaker-dependent training on an LDA-transformed version of the Discrete Cosine Transforms of the mouth regions in an HMM/GMM system.



Dynamic Stream Weighting for Turbo-Decoding-Based Audiovisual ASR

Sebastian Gergen[1], *Steffen Zeiler*[1], *Ahmed Hussen Abdelaziz*[2], *Robert Nickel*[3], *Dorothea Kolossa*[1]

[1] Cognitive Signal Processing Group, Institute of Communication Acoustics, Ruhr-University Bochum
[2] International Computer Science Institute, Berkeley
[3] Department of Electrical and Computer Engineering, Bucknell University
sebastian.gergen@rub.de, steffen.zeiler@rub.de, ahmed.hussenabdelaziz@rub.de, robert.nickel@bucknell.edu, dorothea.kolossa@rub.de

Abstract

Automatic speech recognition (ASR) enables very intuitive human-machine interaction. However, signal degradations due to reverberation or noise reduce the accuracy of audio-based recognition. The introduction of a second signal stream that is not affected by degradations in the audio domain (e.g., a video stream) increases the robustness of ASR against degradations in the original domain. Here, depending on the signal quality of audio and video at each point in time, a dynamic weighting of both streams can optimize the recognition performance. In this work, we introduce a strategy for estimating optimal weights for the audio and video streams in turbo-decoding-based ASR using a discriminative cost function. The results show that turbo decoding with this maximally discriminative dynamic weighting of information yields higher recognition accuracy than turbo-decoding-based recognition with fixed stream weights or optimally dynamically weighted audiovisual decoding using coupled hidden Markov models.

Index Terms: Audiovisual speech recognition, Turbo decoding, Stream weighting

in [1, 8] and time-adaptive weights (dynamic stream weights, DSW) were proposed in [9]. In the latter paper, an algorithm to estimate DSWs for CHMM-based ASR was introduced for different signal-to-noise ratios (SNR) in the audio stream. Unfortunately, the method proposed in [9] for CHMM-based ASR does not readily carry over to TD-based ASR.

In this contribution, we introduce a new estimation strategy for optimal DSWs for TD-based ASR. We propose to use the stream weights of audio and video streams to maximize a discriminative cost function in each time frame and TD-iteration. This cost function uses the knowledge of the correct word sequence of the current sentence in form of an *oracle* path through the HMM-states as well as the N-best (*confusion*) paths which result in wrong word sequences. With these oracle-based DSWs, estimation algorithms for blind estimation of high quality DSWs will be developed in the future.

The paper is structured as follows. In Section 2 we introduce the basic concepts of audiovisual ASR with CHMM and TD and we discuss the estimation of optimal SNR- and noise-type-dependent fixed stream weights. In Section 3, we describe the proposed estimation method for dynamic stream weights for TD. Our experimental setup is presented in Section 4. The accuracy of ASR systems with and without the proposed method is evaluated in Section 5.

1. Introduction

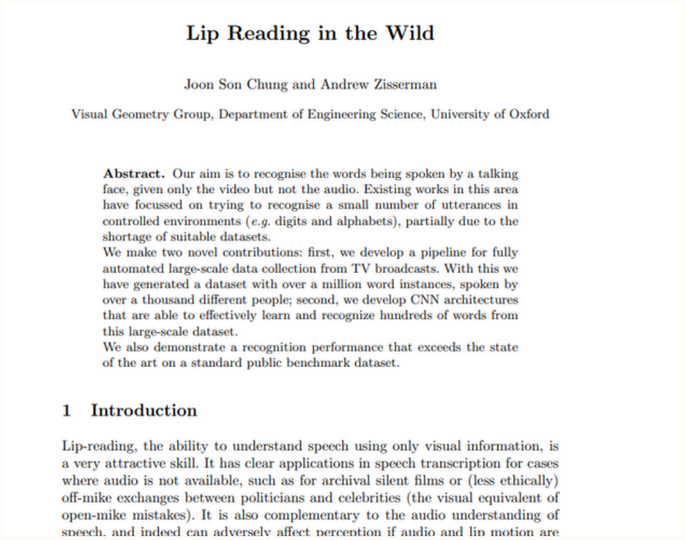
Classification with deep learning

2016

Lip Reading in the Wild

Chung & Zisserman

Propose spatial and spatiotemporal **convolutional neural networks,** based on **VGG**, for word classification.



Lip Reading in the Wild

Joon Son Chung and Andrew Zisserman

Visual Geometry Group, Department of Engineering Science, University of Oxford

Abstract. Our aim is to recognise the words being spoken by a talking face, given only the video but not the audio. Existing works in this area have focussed on trying to recognise a small number of utterances in controlled environments (*e.g.* digits and alphabets), partially due to the shortage of suitable datasets.
We make two novel contributions: first, we develop a pipeline for fully automated large-scale data collection from TV broadcasts. With this we have generated a dataset with over a million word instances, spoken by over a thousand different people; second, we develop CNN architectures that are able to effectively learn and recognize hundreds of words from this large-scale dataset.
We also demonstrate a recognition performance that exceeds the state of the art on a standard public benchmark dataset.

1 Introduction

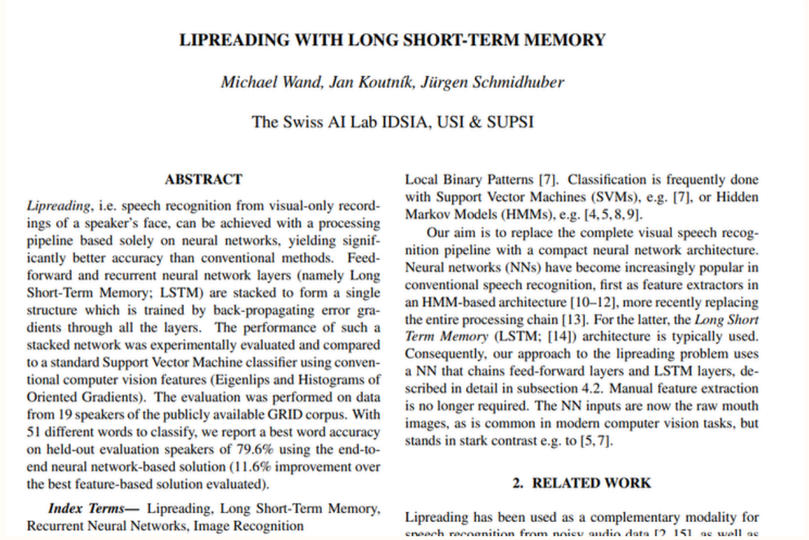
Lip-reading, the ability to understand speech using only visual information, is a very attractive skill. It has clear applications in speech transcription for cases where audio is not available, such as for archival silent films or (less ethically) off-mike exchanges between politicians and celebrities (the visual equivalent of open-mike mistakes). It is also complementary to the audio understanding of speech, and indeed can adversely affect perception if audio and lip motion are

2016

LIPREADING WITH LONG SHORT-TERM MEMORY

Wand et al.

Introduce **LSTM** recurrent neural networks for lipreading but address neither sentence-level sequence prediction nor speaker independence.



LIPREADING WITH LONG SHORT-TERM MEMORY

Michael Wand, Jan Koutník, Jürgen Schmidhuber

The Swiss AI Lab IDSIA, USI & SUPSI

ABSTRACT

Lipreading, i.e. speech recognition from visual-only recordings of a speaker's face, can be achieved with a processing pipeline based solely on neural networks, yielding significantly better accuracy than conventional methods. Feed-forward and recurrent neural network layers (namely Long Short-Term Memory; LSTM) are stacked to form a single structure which is trained by back-propagating error gradients through all the layers. The performance of such a stacked network was experimentally evaluated and compared to a standard Support Vector Machine classifier using conventional computer vision features (Eigenlips and Histograms of Oriented Gradients). The evaluation was performed on data from 19 speakers of the publicly available GRID corpus. With 51 different words to classify, we report a best word accuracy on held-out evaluation speakers of 79.6% using the end-to-end neural network-based solution (11.6% improvement over the best feature-based solution evaluated).

Index Terms— Lipreading, Long Short-Term Memory, Recurrent Neural Networks, Image Recognition

Local Binary Patterns [7]. Classification is frequently done with Support Vector Machines (SVMs), e.g. [7], or Hidden Markov Models (HMMs), e.g. [4, 5, 8, 9].

Our aim is to replace the complete visual speech recognition pipeline with a compact neural network architecture. Neural networks (NNs) have become increasingly popular in conventional speech recognition, first as feature extractors in an HMM-based architecture [10–12], more recently replacing the entire processing chain [13]. For the latter, the *Long Short Term Memory* (LSTM; [14]) architecture is typically used. Consequently, our approach to the lipreading problem uses a NN that chains feed-forward layers and LSTM layers, described in detail in subsection 4.2. Manual feature extraction is no longer required. The NN inputs are now the raw mouth images, as is common in modern computer vision tasks, but stands in stark contrast e.g. to [5, 7].

2. RELATED WORK

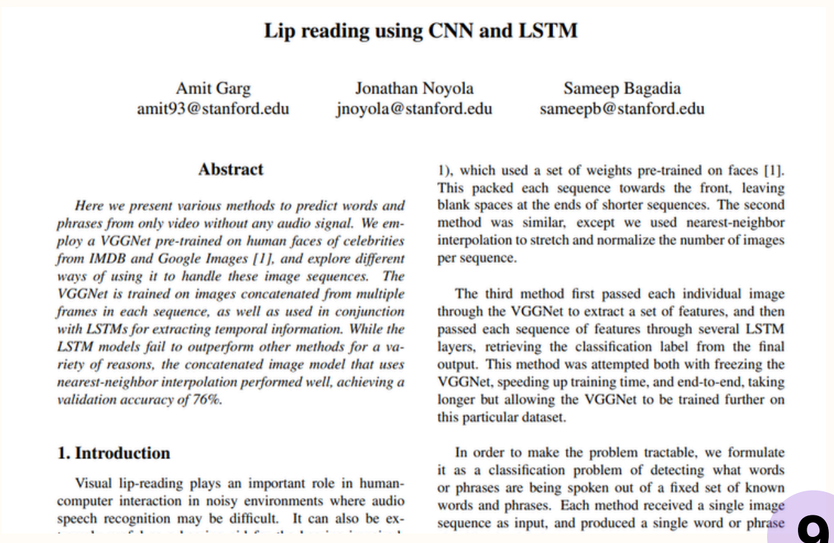
Lipreading has been used as a complementary modality for

2016

Lip reading using CNN and LSTM

Garg et al.

Apply a **VGG** pre-trained on faces to classifying words and phrases from the **MIRACL-VC1** dataset. Their best model achieves only **56,0% word classification** accuracy, and **44,5% phrase classification** accuracy,



Lip reading using CNN and LSTM

Amit Garg amit93@stanford.edu
Jonathan Noyola jnoyola@stanford.edu
Sameep Bagadia sameepb@stanford.edu

Abstract

Here we present various methods to predict words and phrases from only video without any audio signal. We employ a VGGNet pre-trained on human faces of celebrities from IMDB and Google Images [1], and explore different ways of using it to handle these image sequences. The VGGNet is trained on images concatenated from multiple frames in each sequence, as well as used in conjunction with LSTMs for extracting temporal information. While the LSTM models fail to outperform other methods for a variety of reasons, the concatenated image model that uses nearest-neighbor interpolation performed well, achieving a validation accuracy of 76%.

1. Introduction

Visual lip-reading plays an important role in human-computer interaction in noisy environments where audio speech recognition may be difficult. It can also be ex-

1), which used a set of weights pre-trained on faces [1]. This packed each sequence towards the front, leaving blank spaces at the ends of shorter sequences. The second method was similar, except we used nearest-neighbor interpolation to stretch and normalize the number of images per sequence.

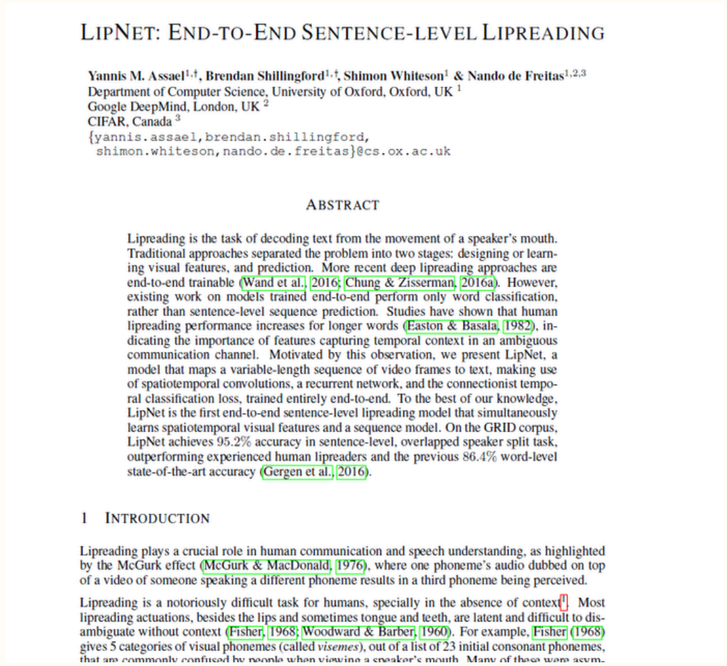
The third method first passed each individual image through the VGGNet to extract a set of features, and then passed each sequence of features through several LSTM layers, retrieving the classification label from the final output. This method was attempted both with freezing the VGGNet, speeding up training time, and end-to-end, taking longer but allowing the VGGNet to be trained further on this particular dataset.

In order to make the problem tractable, we formulate it as a classification problem of detecting what words or phrases are being spoken out of a fixed set of known words and phrases. Each method received a single image sequence as input, and produced a single word or phrase

2017

LIPNET: END-TO-END SENTENCE-LEVEL LIPREADING

LipNet was the breakthrough model that achieved **end-to-end**, **sentence-level sequence** prediction for lipreading.



LIPNET: END-TO-END SENTENCE-LEVEL LIPREADING

Yannis M. Assael[1,†], Brendan Shillingford[1,†], Shimon Whiteson[1] & Nando de Freitas[1,2,3]
Department of Computer Science, University of Oxford, Oxford, UK [1]
Google DeepMind, London, UK [2]
CIFAR, Canada [3]
{yannis.assael,brendan.shillingford,
shimon.whiteson,nando.de.freitas}@cs.ox.ac.uk

ABSTRACT

Lipreading is the task of decoding text from the movement of a speaker's mouth. Traditional approaches separated the problem into two stages: designing or learning visual features, and prediction. More recent deep lipreading approaches are end-to-end trainable (Wand et al., 2016; Chung & Zisserman, 2016a). However, existing work on models trained end-to-end perform only word classification, rather than sentence-level sequence prediction. Studies have shown that human lipreading performance increases for longer words (Easton & Basala, 1982), indicating the importance of features capturing temporal context in an ambiguous communication channel. Motivated by this observation, we present LipNet, a model that maps a variable-length sequence of video frames to text, making use of spatiotemporal convolutions, a recurrent network, and the connectionist temporal classification loss, trained entirely end-to-end. To the best of our knowledge, LipNet is the first end-to-end sentence-level lipreading model that simultaneously learns spatiotemporal visual features and a sequence model. On the GRID corpus, LipNet achieves 95.2% accuracy in sentence-level, overlapped speaker split task, outperforming experienced human lipreaders and the previous 86.4% word-level state-of-the-art accuracy (Gergen et al., 2016).

1 INTRODUCTION

Lipreading plays a crucial role in human communication and speech understanding, as highlighted by the McGurk effect (McGurk & MacDonald, 1976), where one phoneme's audio dubbed on top of a video of someone speaking a different phoneme results in a third phoneme being perceived.

Lipreading is a notoriously difficult task for humans, specially in the absence of context[1]. Most lipreading actuations, besides the lips and sometimes tongue and teeth, are latent and difficult to disambiguate without context (Fisher, 1968; Woodward & Barber, 1960). For example, Fisher (1968) gives 5 categories of visual phonemes (called *visemes*), out of a list of 23 initial consonant phonemes, that are commonly confused by people when viewing a speaker's mouth. Many of these were asym-

Method	Dataset	Size	Output	Accuracy
Fu et al. (2008)	AVICAR	851	Digits	37.9%
Hu et al. (2016)	AVLetter	78	Alphabet	64.6%
Papandreou et al. (2009)	CUAVE	1800	Digits	83.0%
Chung & Zisserman (2016a)	OuluVS1	200	Phrases	91.4%
Chung & Zisserman (2016b)	OuluVS2	520	Phrases	94.1%
Chung & Zisserman (2016a)	BBC TV	> 400000	Words	65.4%
Gergen et al. (2016)	GRID	29700	Words*	86.4%
LipNet	GRID	28775	**Sentences**	**95.2%**

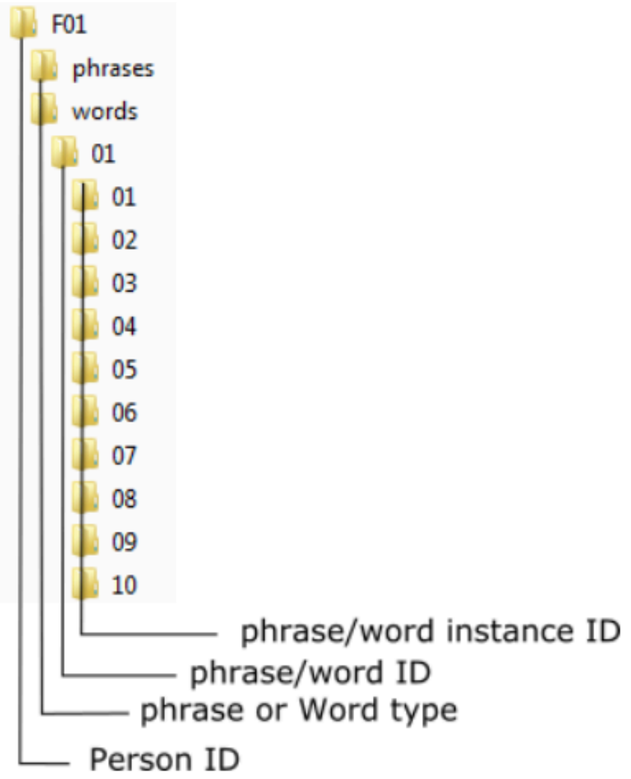
Applying Traditional Approaches For Lip Reading

Dataset

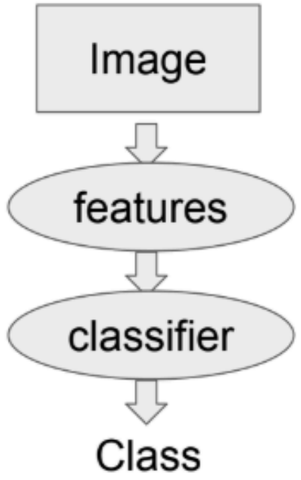
MIRACL-VC1 dataset

The dataset was created from **15 people** who spoke each of **10 words** and **10 phrases ten** times
leading to a total of 15 × 20 × 10 = **3000 instances.**
Each instance is a sequence of **color** and **depth** images of **640 × 480 pixels**.

Example sequence of color images



I . First Approach (Classical Machine Learning)



Images from dataset folders (.jpg files).

1. **Data Loading**

RGB image (H, W, 3)

Loaded images as arrays

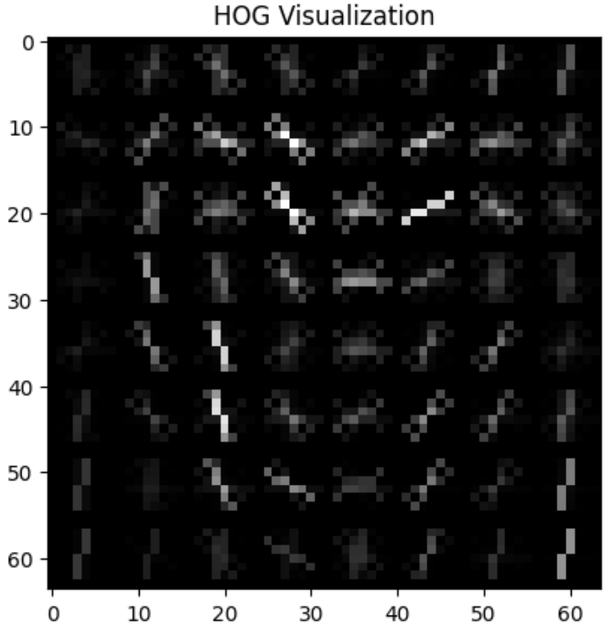
2. Preprocessing

1. **grayscale** conversion.
2. **face detection** using **Haar cascades.** (pretrained from cv2)
3. **mouth region** detection.

Cropped grayscale mouth region (h, w)

3. Feature Extraction

1. **Resize** the mouth region to a fixed size (64x64).
2. Extract **HOG features** to summarize patterns.



I . First Approach (Classical Machine Learning)

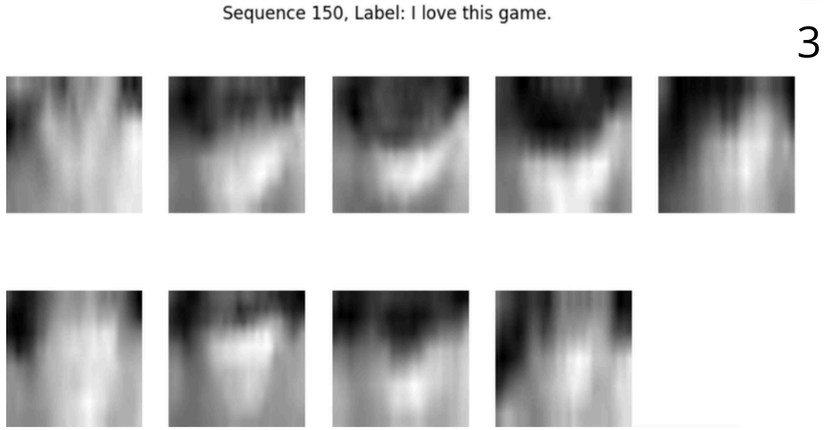
- **Dataset**
- Phrases and words **labels**

Words	Phrases
Begin	Stop navigation.
Choose	Excuse me.
Connection	I am sorry.
Navigation	Thank you.
Next	Good bye.
Previous	I love this game.
Start	Nice to meet you.
Stop	You are welcome.
Hello	How are you?
Web	Have a good time.

1. Dataset Preparation

- **image_data:** (num_samples, N)
- **label_data:** (num_samples,)

Applied **all the steps** for all sequence of images for every instance within each word/phrase ID for every speaker and attribute **each sequence** to the **label**.



2. Train-Test Split

1. **Define Proportions** (80% training, 20% testing)
2. **Shuffle Data**
3. **Split Data**

X_train: (num_train_samples, N)
X_test: (num_test_samples, N)
y_train: (num_train_samples,)
y_test: (num_test_samples,)

3. Classification

Train an **SVM classifier** with **linear Kernel** and use it for predictions.

I . First Approach (Classical Machine Learning)

- Trained SVM model
- X_test (HOG features)

4. Prediction

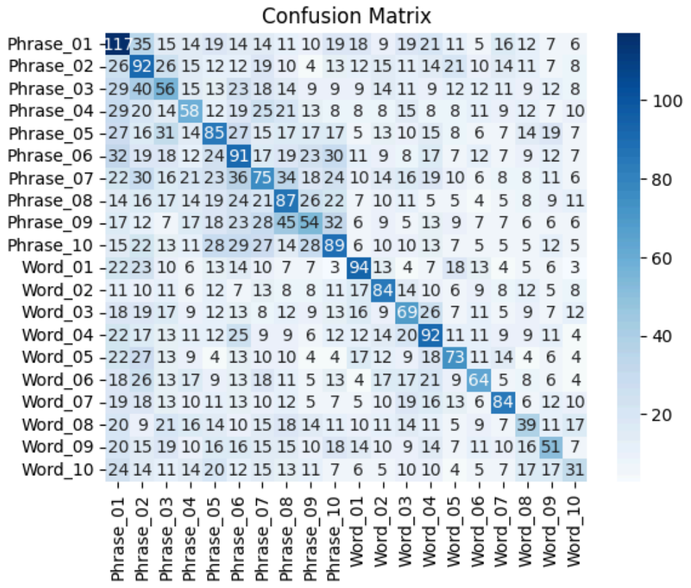
y_pred as a 1D array

True labels y_test

Use the trained SVM model to predict labels for X_test.

5. Evaluation

Model accuracy: 22.95%

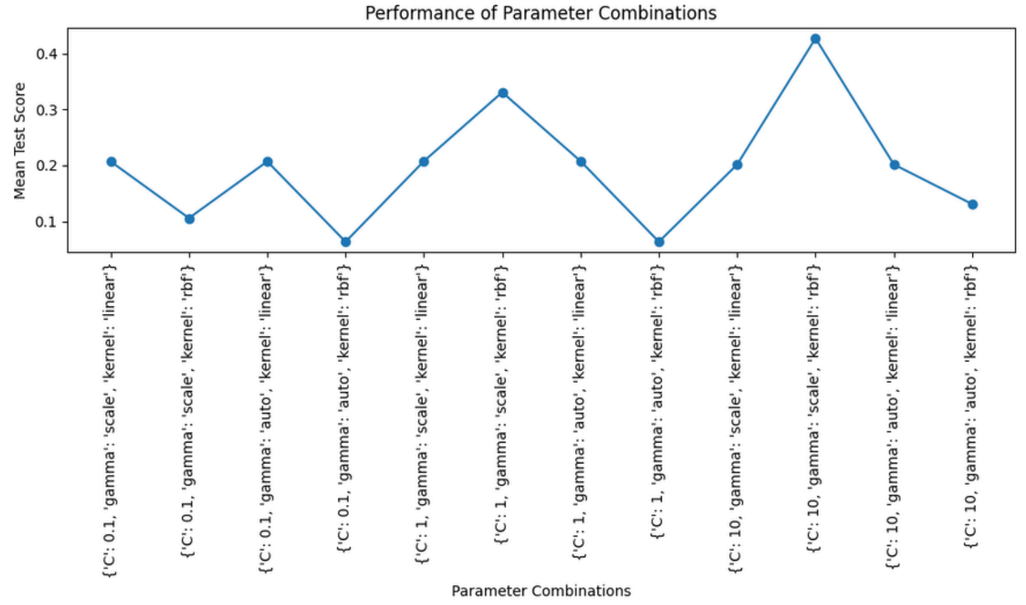


6. Hyperparameter Tuning

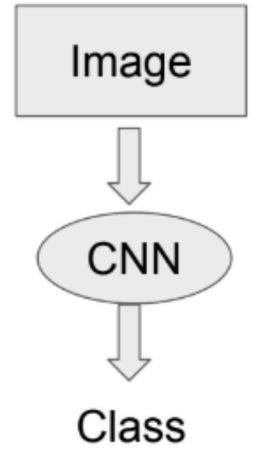
1. **Define Parameter Grid**
   - C: Regularization parameter values (**0.1**, **1**, **10**).
   - kernel: Kernel types (**linear**, **rbf**).
   - gamma: Kernel coefficient (**scale**, **auto**).
2. **Perform Grid Search** (cv=3)
3. **Select the Best Model**

Best model accuracy: 47.65%

Best parameters: {'C': 10, 'gamma': 'scale', 'kernel': 'rbf'}



II. Second Approach (Use Deep Learning for Feature Extraction part)

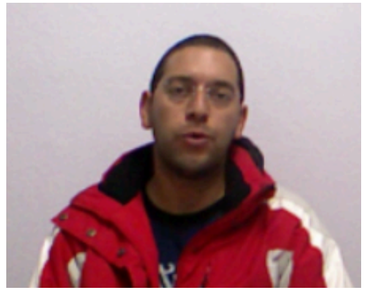


An image containing a face

only **3000 instances** is **small** for **deep learning tasks**

1.Detecting and cropping mouth using Mediapipe

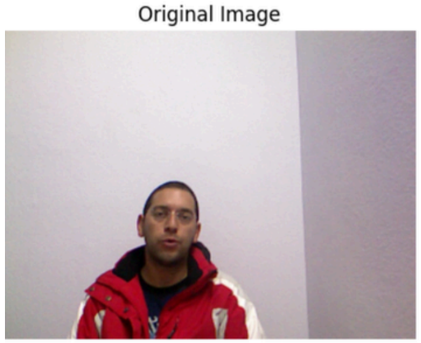
- Detect facial landmarks
- Identify specific points on the mouth (e.g., corners, center).
- Extract the mouth's bounding box and crop the image to only contain the mouth region.

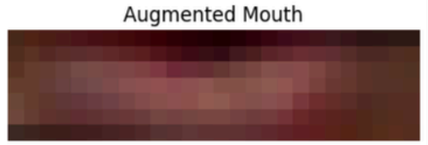


A cropped image of the mouth region.

2. Data Augmentation

1. While cropping, slightly move around the crop region by random number of pixels horizontally and vertically
2. Jitter the image by randomly increasing or decreasing the pixel values of the image by a small amount.





Augmented (transformed) mouth image.

3. Preprocessing

1. **Resize** the image to 64x64.
2. **Convert** the image to grayscale.
3. **Normalize** the pixel values to a range of 0 to 1.

II. Second Approach

- Dataset
- Word and phrase labels.
- Augmentation flag.

1.Create augmented dataset

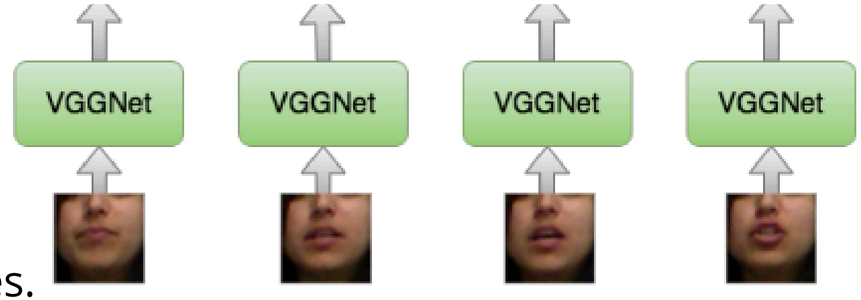
Loads image data from a and then for every image apply all the previous process and add to the corresponding sequence.

- X: A list of sequences of preprocessed mouth images.
- y: A list of corresponding labels for the sequences.

2. Build VGG Extractor

A VGG16 model (pre-trained on **ImageNet**), but without the top classification layer.

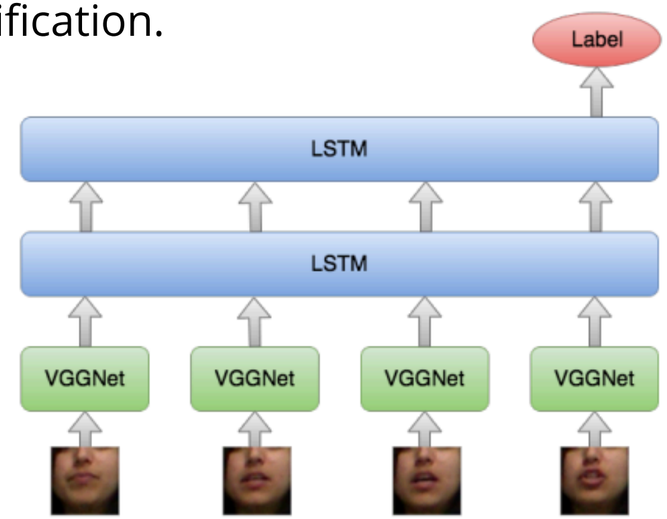
feature maps from the last convolutional layer



Input shape (sequence length, image height, image width, channels).
Number of classes for classification.

3. Build Recurrent (LSTM) model

1. **TimeDistributed Layer**: Applies the VGG feature extractor to each frame in a sequence.
2. **Flatten Layer:** Flattens the features extracted from each time step.
3. **LSTM Layers:** Long Short-Term Memory (LSTM) layers to capture temporal dependencies.
4. **Dense Layers:** Fully connected layers for classification.



II. Second Approach

4.Preprocessing

Preprocessed and padded sequences of mouth images (X).

Corresponding labels (y).

1. **Preprocess labels :** Encoding labels
2. **Preprocess sequences:** Sequences of mouth images vary in length. So we padded shorter sequences with zeros.

5.Training the Model

The trained model

6. Model Evaluation

The model is trained on the dataset using VGG16 for feature extraction and LSTM for sequence learning

The model failed because it does not handle the sequence until after feature extraction.

VGG16 as a feature extractor followed by two LSTM layers and fully connected layers might overfit or fail to converge if the dataset is small or noisy.

II. Second Approach

3D convolutional neural network (CNN) designed for **processing sequences** of image frames and **classifying** them into predefined categories.

(sequence_length, height, width, channels)

Conv3D

- Filters: 32
- Kernel size: (3, 3, 3)
- Activation: ReLU

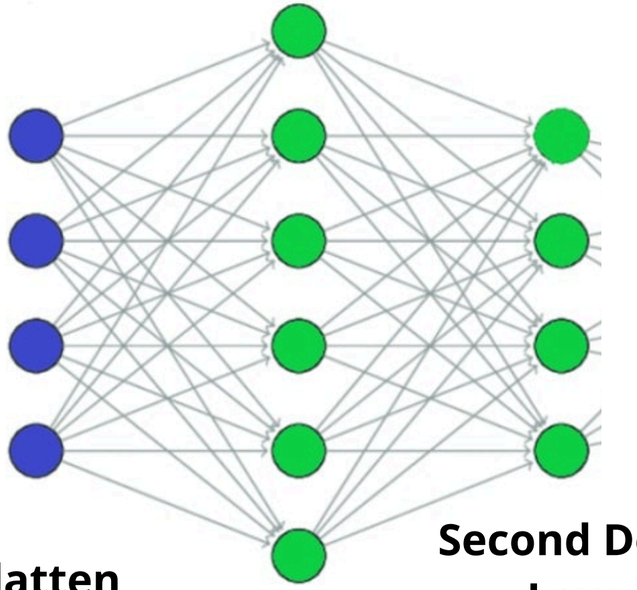
MaxPooling3D

Conv3D

- Filters: 64
- Kernel size: (3, 3, 3)
- Activation: ReLU

MaxPooling3D

Flatten

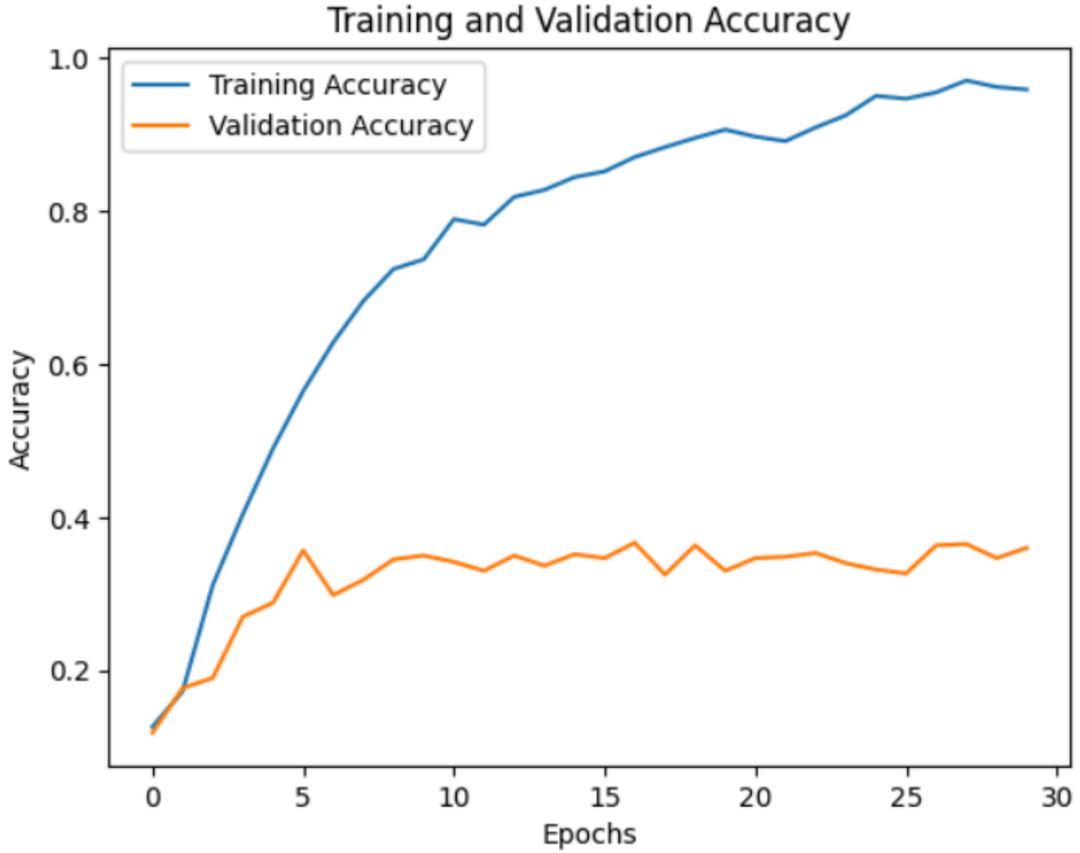


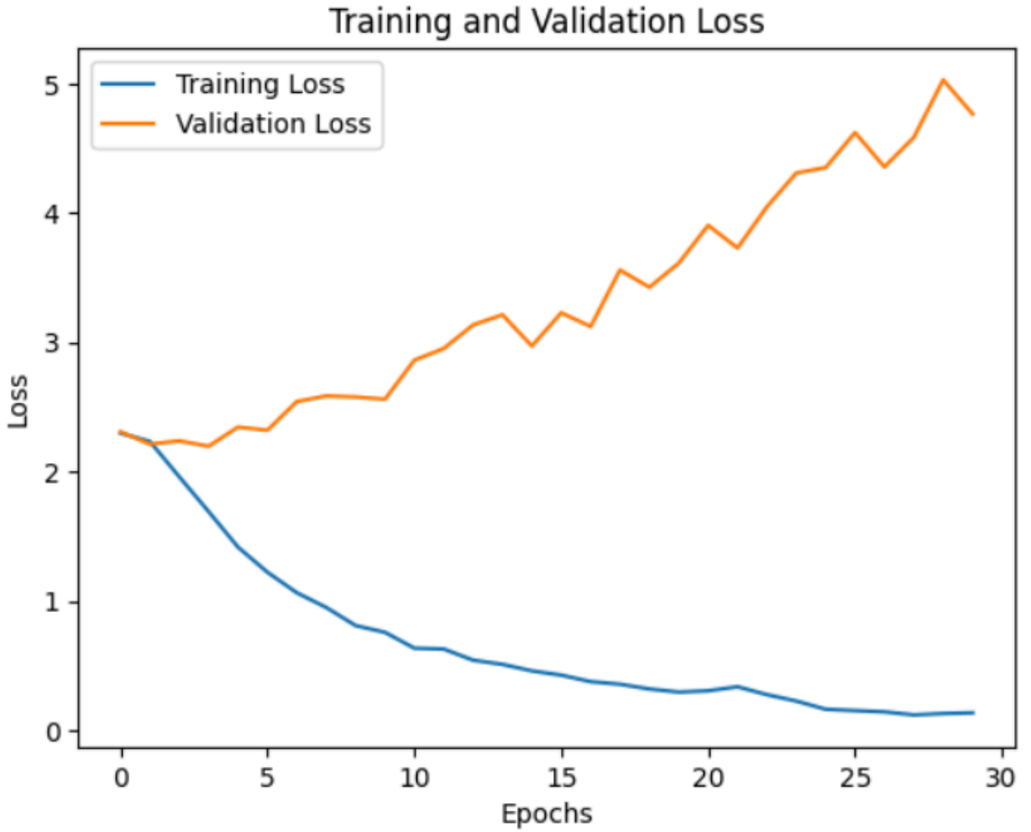
First Dense Layer
128 neurons with ReLU activation

Second Dense Layer
20 classes

Optimizer: Adam
Loss Function: Sparse categorical cross-entropy

II. Second Approach





Overfitting

LipNet : End-to-End Sentence-Level Lip Reading

How Easy is LipReading ?

how easy do you think lipreading is?

let's give it a try

Why ?

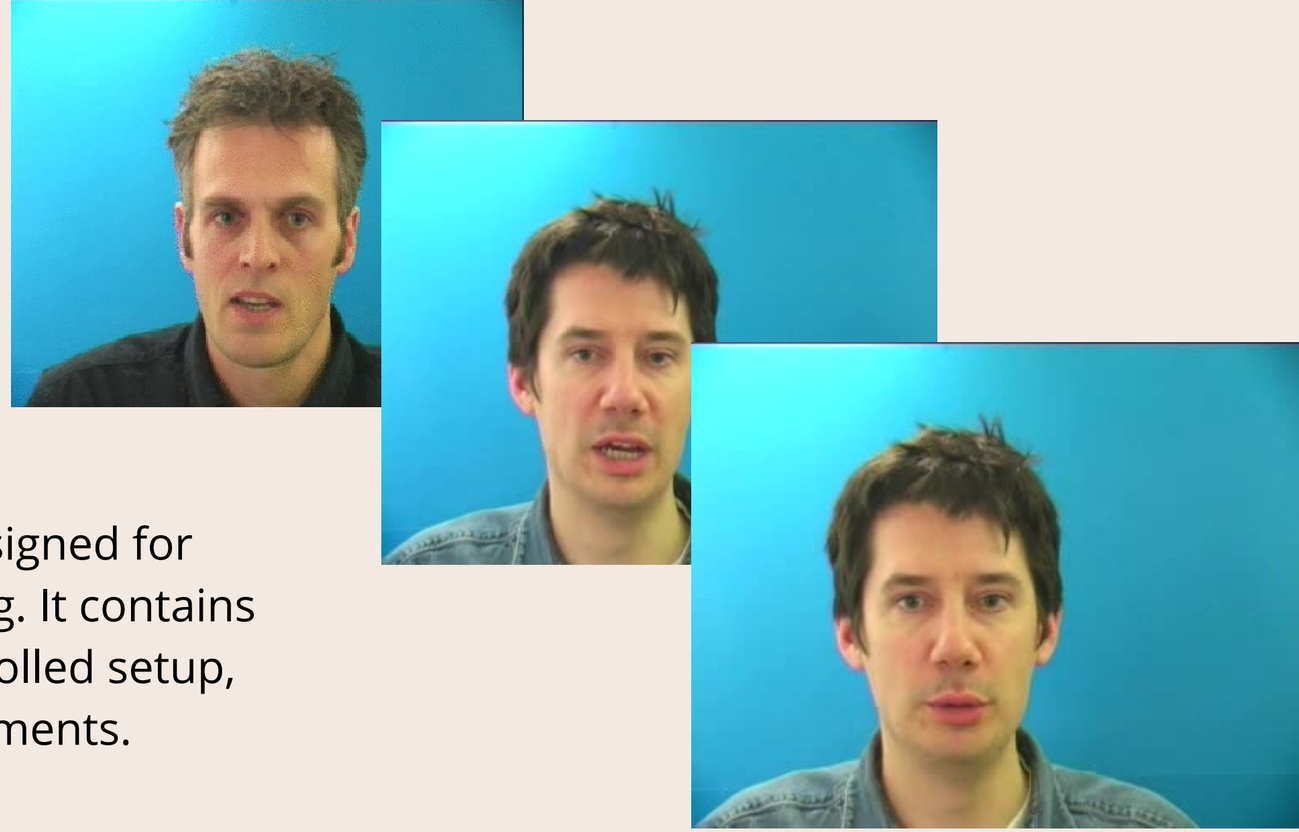
Human Limitations in Lipreading

- **Low Accuracy :**
  - Hearing-impaired individuals achieve only 17% ± 12% accuracy for a limited set of 30 monosyllabic words (Easton & Basala, 1982).
- **Ambiguity of Visemes:**
  - Fisher (1968) observed that 23 consonant phonemes can be grouped into just 5 visual categories (visemes), leading to frequent confusion.
- **Reliance on Context**

Limitations of Traditional Lipreading Approaches

- **Fragmented Pipeline:**
  - Traditional approaches separate feature extraction and prediction, making them inefficient and harder to optimize jointly.
- **Word-Level Restriction:**
  - Most existing models are limited to word classification, ignoring temporal dependencies required for sentence-level understanding
- **Reliance on Hand coded Features**
- **SOTA model Accuracy is 86% :**
  - (Gergen et al., 2016).

Dataset : GRID Corpus



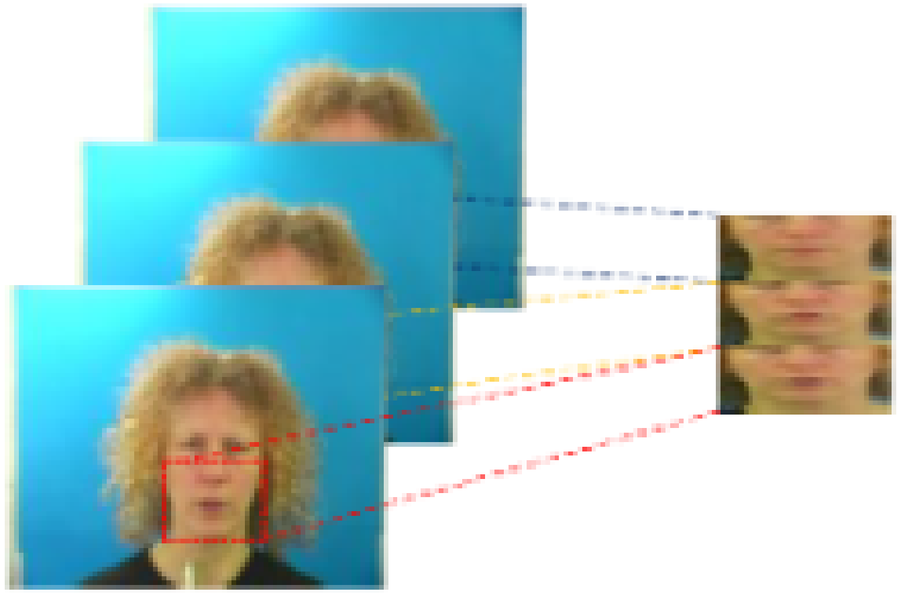
The GRID Corpus : is a large audiovisual dataset designed for research in speech recognition and machine learning. It contains recordings of speakers uttering sentences in a controlled setup, allowing for precise analysis of speech and lip movements.

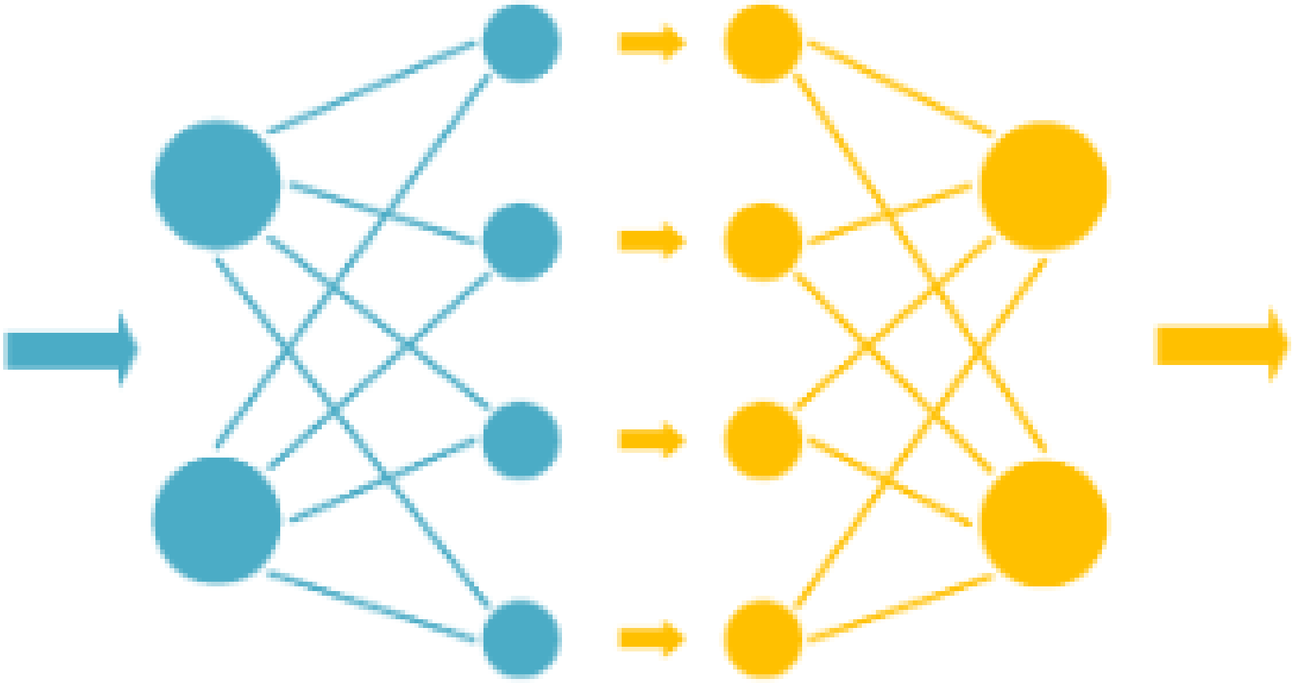
Composition:

- 34 speakers (male and female) recorded under controlled conditions.
- Each speaker utters 1,000 sentences.
- Sentences follow a structured format: command, color, preposition, letter, digit, adverb (e.g., “Place blue at C 9 now”).

Too much memory consumption : training will be conducted on person

How ?





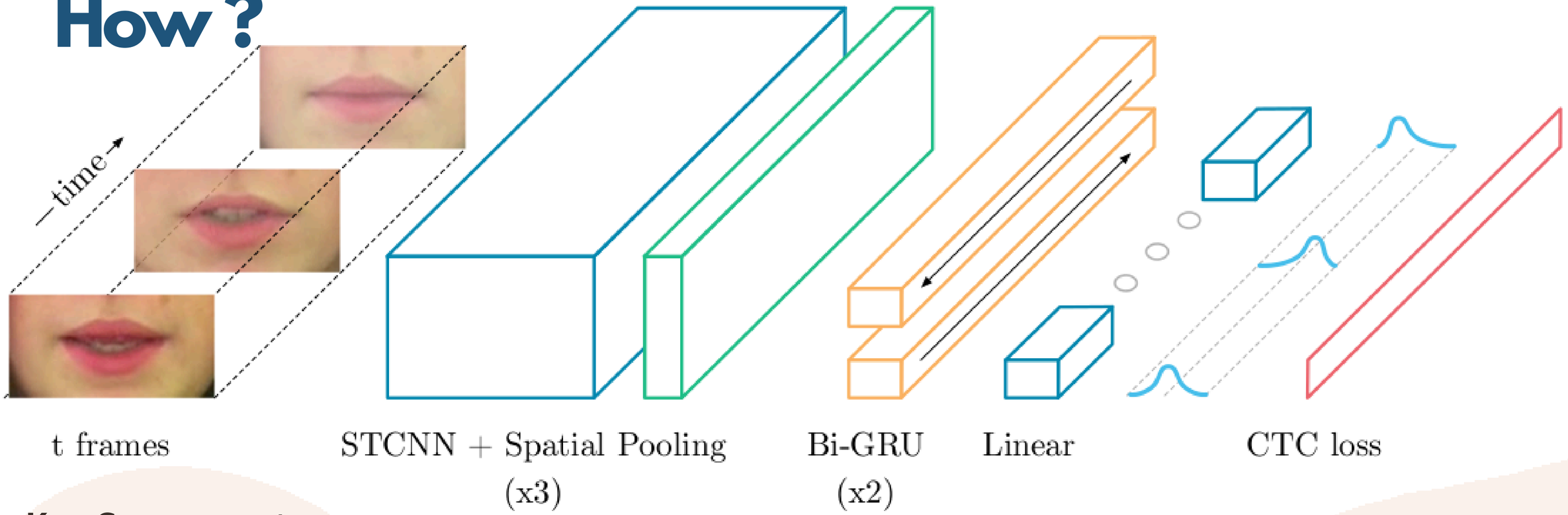
Lay
red
in
a
four
now

Mouth region crop

Feature encode

Text decode

How ?



Key Components:

- **Spatiotemporal Convolutions (STCNN):** Extract spatial (lip shapes) and temporal (lip movement) features.
- **Long Short Term Memory(LSTM):** Capture sequential dependencies over time.
- **Connectionist Temporal Classification (CTC) Loss:** Align predictions with variable-length target sequences.

SpacioTermporal Covolution(STCNN):

Purpose: Extract spatial features from each frame and capture temporal dependencies across frames.

Mathematical Formulation:

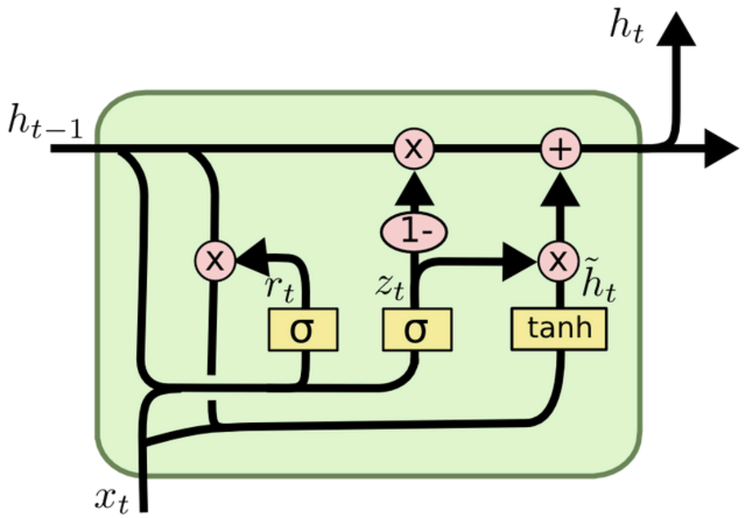
Basic 2D convolution layer from C channels to C 0 channels (without a bias and with unit stride) computes

$$[\mathrm{conv}(\mathbf{x}, \mathbf{w})]_{c'ij} = \sum_{c=1}^{C} \sum_{i'=1}^{k_w} \sum_{j'=1}^{k_h} w_{c'ci'j'} x_{c,i+i',j+j'},$$

for input x and weights w ∈ RC ×C×kw ×kh where we define xcij = 0 for i, j out of bounds :

$$[\mathrm{stconv}(\mathbf{x}, \mathbf{w})]_{c'tij} = \sum_{c=1}^{C} \sum_{t'=1}^{k_t} \sum_{i'=1}^{k_w} \sum_{j'=1}^{k_h} w_{c'ct'i'j'} x_{c,t+t',i+i',j+j'}.$$

Long Short Term Memory(LSTM):



$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right)$$

$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right)$$

$$\tilde{h}_t = \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right)$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

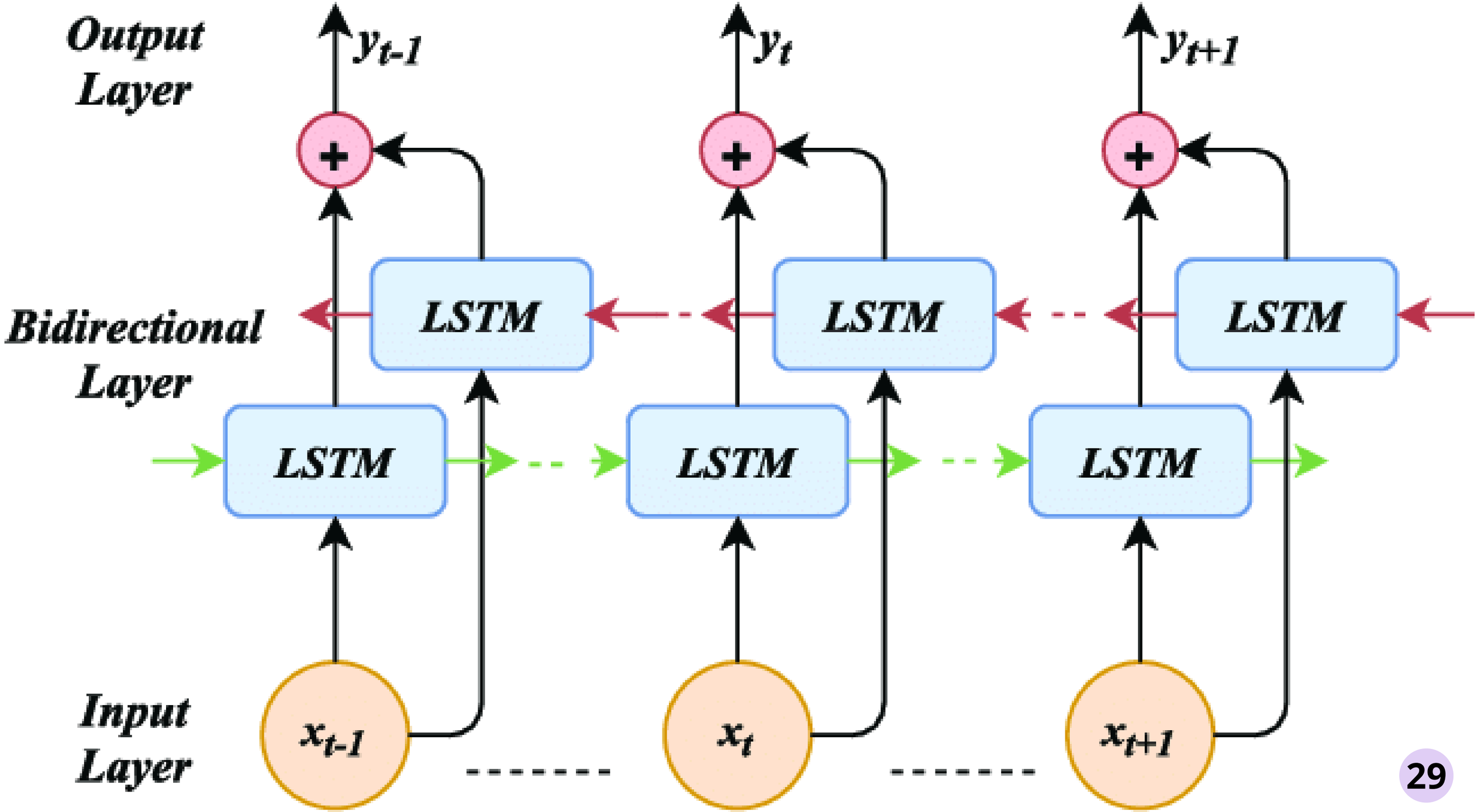
Mathematical Model of GRU (Proposed by the paper):

$$[\mathbf{u}_t, \mathbf{r}_t]^T = \text{sigm}(\mathbf{W}_z \mathbf{z}_t + \mathbf{W}_h \mathbf{h}_{t-1} + \mathbf{b}_g)$$

$$\tilde{\mathbf{h}}_t = \tanh(\mathbf{U}_z \mathbf{z}_t + \mathbf{U}_h(\mathbf{r}_t \odot \mathbf{h}_{t-1}) + \mathbf{b}_h)$$

$$\mathbf{h}_t = (\mathbf{1} - \mathbf{u}_t) \odot \mathbf{h}_{t-1} + \mathbf{u}_t \odot \tilde{\mathbf{h}}_t$$

Bi-directional ?


Output Layer
y_{t-1}
y_t
y_{t+1}
Bidirectional Layer
LSTM
LSTM
LSTM
LSTM
LSTM
LSTM
Input Layer
x_{t-1}
x_t
x_{t+1}

Connectionist Temporal Classification (CTC) Loss:

Purpose: specialized loss function used in tasks where input and output sequences have different lengths and frame-level alignment is unknown

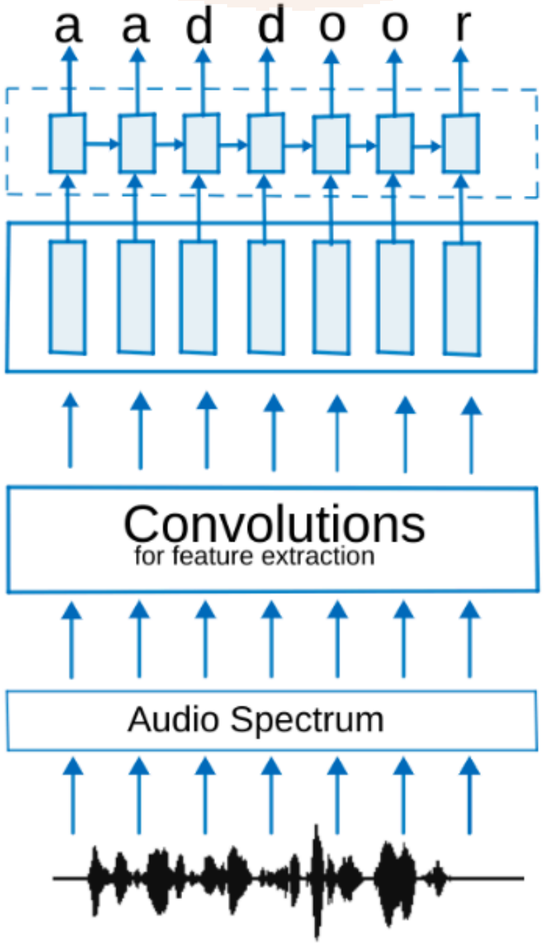
Main Issues to solve : Missalignment the input sequence (e.g., video frames or audio features) is typically longer than the output sequence (e.g., text).

Role : Automatically aligns input frames to output labels by considering all possible alignments during training.

- How do we contract the decoded output to represent our predictions?
- How should we deal with silences in the Video?
- How should we indicate repetitions of tokens as in “d-oo-r”

Output after min CTC Loss: __d_oor

How ?



How ?

Solution:

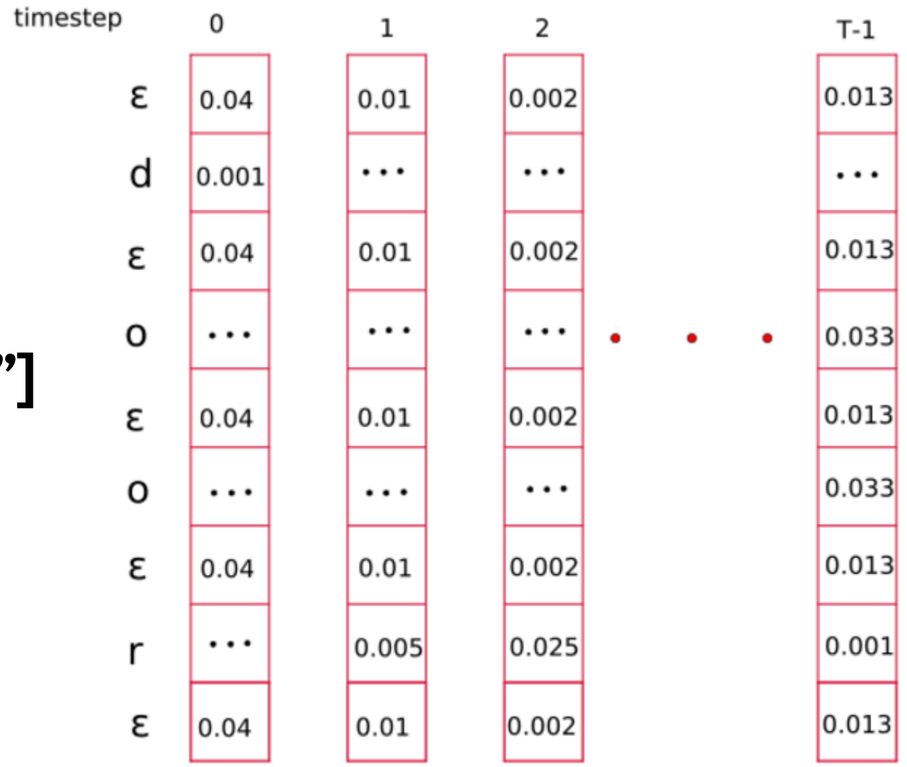
- **Blank token** into our vocabulary to cater for these dynamics : ε
- **Separator token** to indicate spaces between each word : “_”

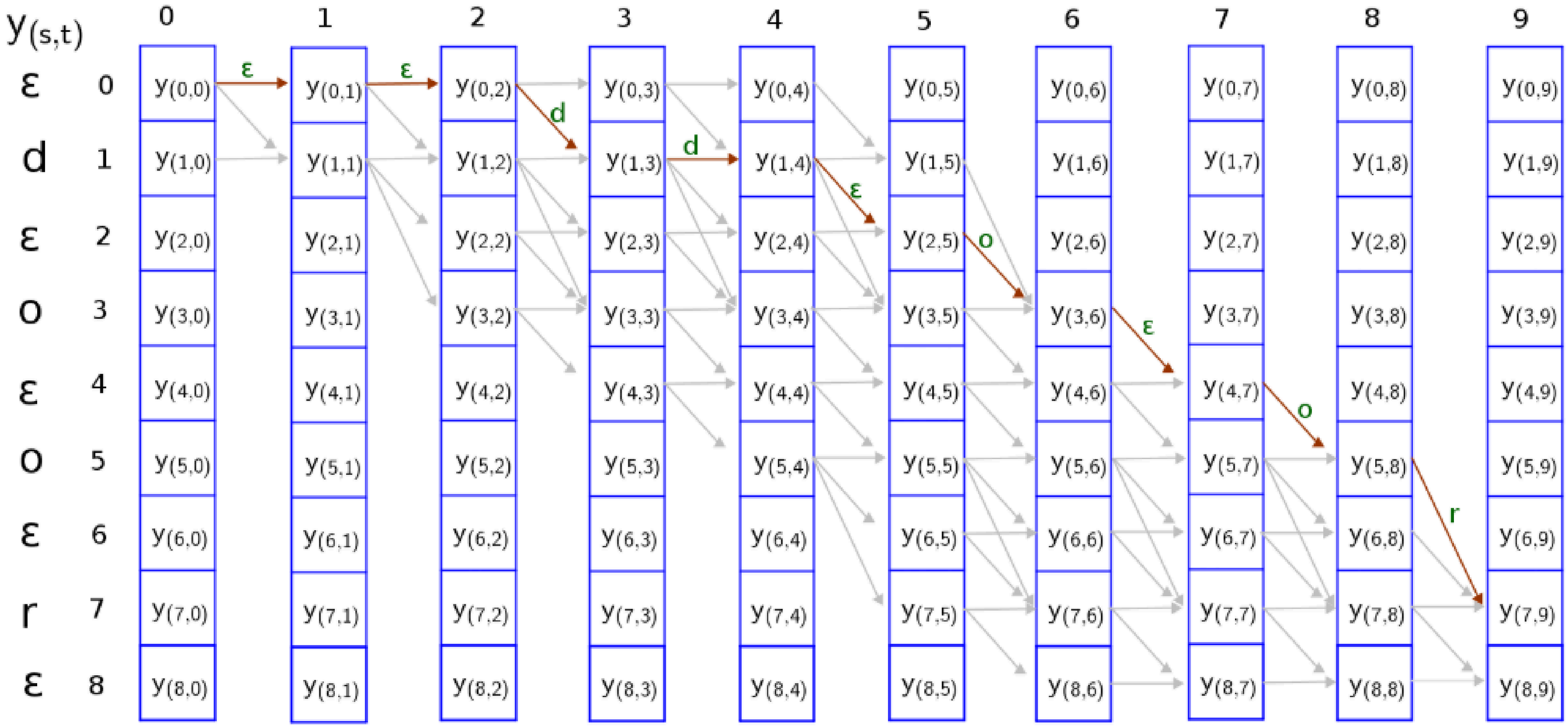
a door

[“ε”, “a”, “_”, “d”, “o”, “o”, “r”]

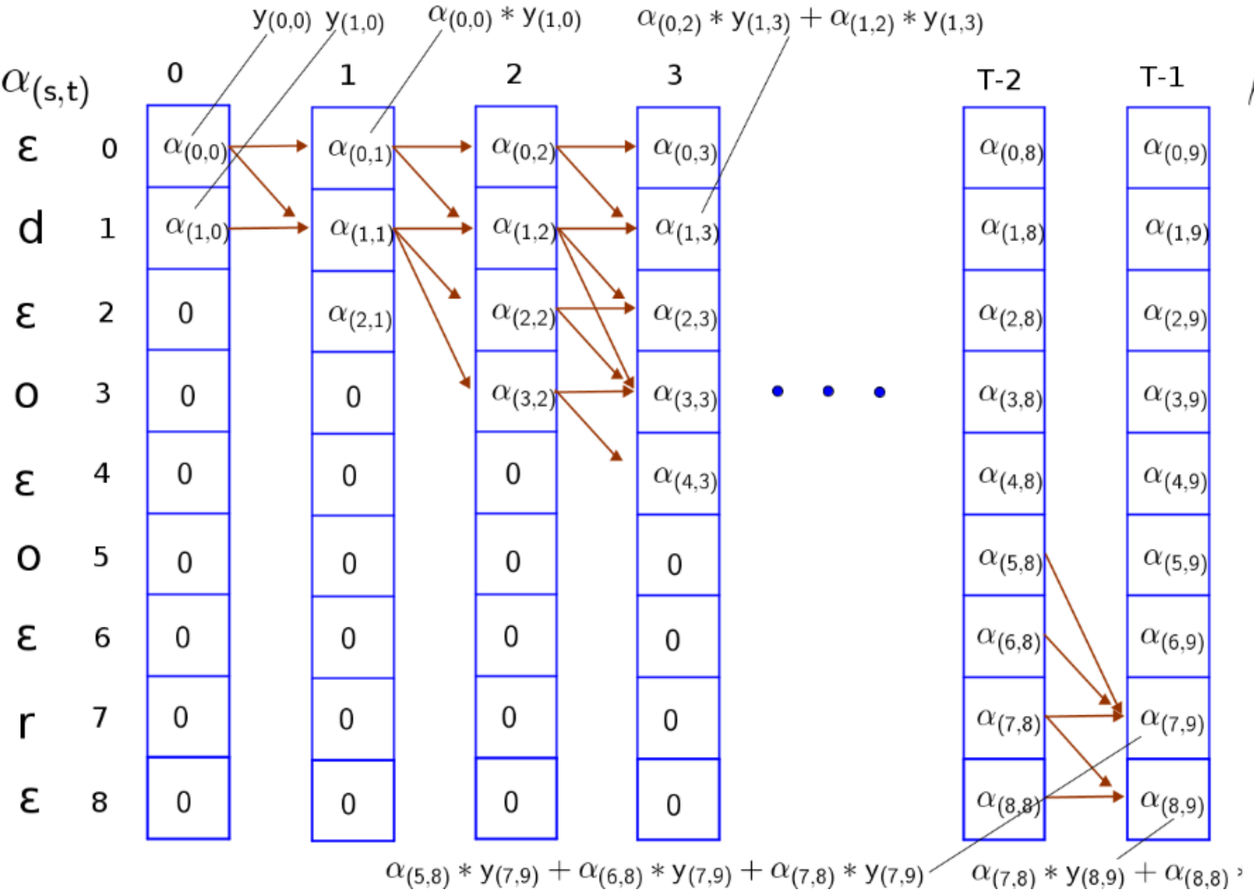
[“ε”, “a”, “ε”, “_”, “ε”, “d”, “ε”, “o”, “ε”, “o”, “ε”, “r”, “ε”]

[“ε”:0, “_“:1, “a”: 2, “b”:3, … ,”z”:28]

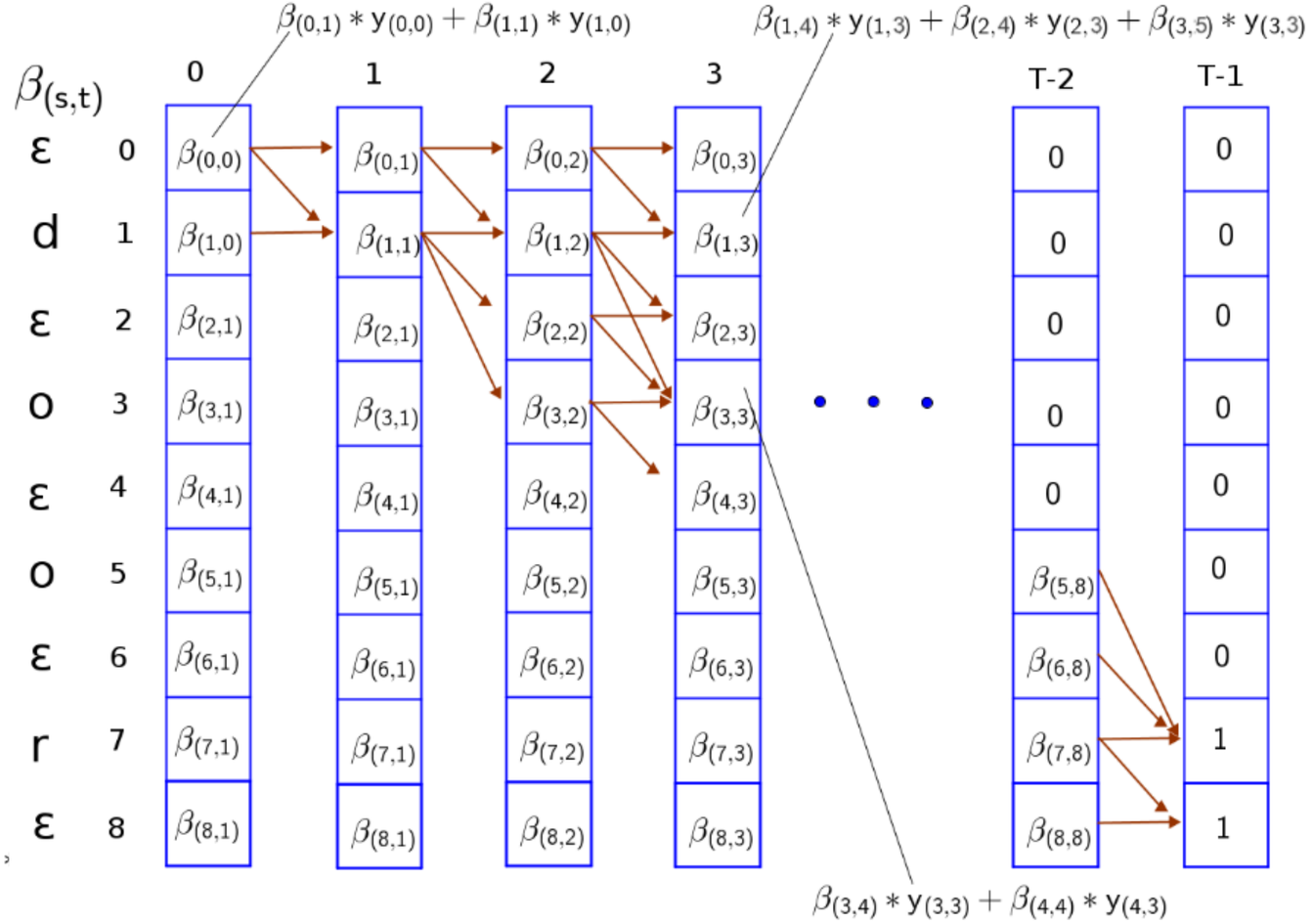



y(s,t)
0
1
2
3
4
5
6
7
8
9
ε 0
d 1
ε 2
o 3
ε 4
o 5
ε 6
r 7
ε 8
y(0,0) y(0,1) y(0,2) y(0,3) y(0,4) y(0,5) y(0,6) y(0,7) y(0,8) y(0,9)
y(1,0) y(1,1) y(1,2) y(1,3) y(1,4) y(1,5) y(1,6) y(1,7) y(1,8) y(1,9)
y(2,0) y(2,1) y(2,2) y(2,3) y(2,4) y(2,5) y(2,6) y(2,7) y(2,8) y(2,9)
y(3,0) y(3,1) y(3,2) y(3,3) y(3,4) y(3,5) y(3,6) y(3,7) y(3,8) y(3,9)
y(4,0) y(4,1) y(4,2) y(4,3) y(4,4) y(4,5) y(4,6) y(4,7) y(4,8) y(4,9)
y(5,0) y(5,1) y(5,2) y(5,3) y(5,4) y(5,5) y(5,6) y(5,7) y(5,8) y(5,9)
y(6,0) y(6,1) y(6,2) y(6,3) y(6,4) y(6,5) y(6,6) y(6,7) y(6,8) y(6,9)
y(7,0) y(7,1) y(7,2) y(7,3) y(7,4) y(7,5) y(7,6) y(7,7) y(7,8) y(7,9)
y(8,0) y(8,1) y(8,2) y(8,3) y(8,4) y(8,5) y(8,6) y(8,7) y(8,8) y(8,9)
ε
ε
d
d
ε
o
ε
o
r

Forward



Backward



$$P_{(seq_t,t)} = \sum_{s=0}^{S} \frac{\alpha_{s,t}\beta_{s,t}}{y_{s,t}} \quad \rightarrow \quad l = -\sum_{t=0}^{T-1} logP_{(seq_t,t)}$$

$$\gamma_{s,t} = \alpha_{s,t}\beta_{s,t}$$

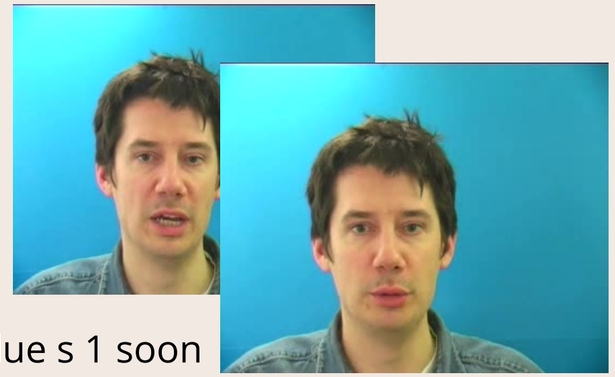
B(_A__AAAA_BBBCCCC) = **B**(A_A_BBBB_CC) = AABC

Define $\mathbf{B}^{-1}$ to map a label sequence **z** to the set of all possible label sequences (paths in π) that collapse to **z**. So {**B**(x) | x ϵ $\mathbf{B}^{-1}$(y)} = y.

Therefore, we can consider the likelihood of a given labelling **z** as the sum of the probabilities of all the paths that can collapse to **z**.

$$p(\mathrm{z}\,|\,\mathrm{x};\theta) = \sum_{\pi \in B^{-1}(\mathrm{z})} p(\pi\,|\,\mathrm{x};\theta)$$

Data Loading - Preprocessing

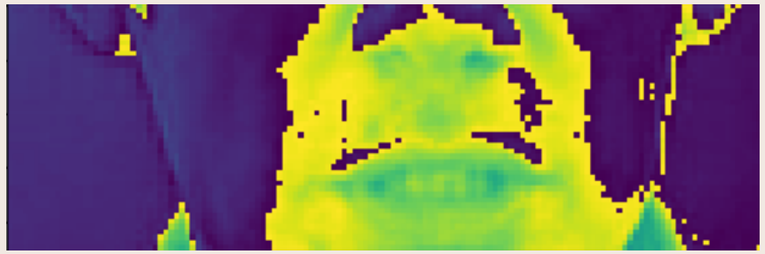


bin blue s 1 soon

bin blue f 2 now

5 8 22 2 ... 24 22

5 8 22 2 ... 23 25



X 75

Data Loading :

- **Pipeline :** Loading a batch of 2 tensors, each 75 frame along with each respective alignement

Normalization :

- Each frame is normalized and rendered down to minimise memory use
- one channel for each image in every frame

```
(75, 46, 140, 1)
```

Training and Predicitons

```
[31]    def __init__(self, dataset) -> None:
            self.dataset = dataset.as_numpy_iterator()

        def on_epoch_end(self, epoch, logs=None) -> None:
            data = self.dataset.next()
            yhat = self.model.predict(data[0])
            decoded = tf.keras.backend.ctc_decode(yhat, [75,75], greedy=False)[0][0].numpy()
            for x in range(len(yhat)):
                print('Original:', tf.strings.reduce_join(num_to_char(data[1][x])).numpy().decode('utf-8'))
                print('Prediction:', tf.strings.reduce_join(num_to_char(decoded[x])).numpy().decode('utf-8'))
                print('~'*100)
```

```
[32] model.compile(optimizer=Adam(learning_rate=0.0001), loss=CTCLoss)
```

```
#Making Callbakcs
checkpoint_callback = ModelCheckpoint(os.path.join('models','checkpoint'), monitor='loss', save_weights_only=True)
schedule_callback = LearningRateScheduler(scheduler)
example_callback = ProduceExample(test)
model.fit(train, validation_data=test, [illegible]callbacks=[checkpoint_callback, schedule_callback, example_callback])
```

```
Epoch 1/100
  3/450 [..............................] - ETA: 6:23:04 - loss: 184.2946
```

Future and Perspectives

State of the art model (SOTA): As of 2022 LipNet as it is now is no longer the SOTA model for LipReading tasks

1	**CTC/Attention**	1.2	✓	Visual Speech Recognition for Multiple Languages in the Wild			2022	
2	**LCANet**	2.9	×	LCANet: End-to-End Lipreading with Cascaded Attention-CTC			2018	
3	**LipNet** (with Face Cutout)	2.9	×	Can We Read Speech Beyond the Lips? Rethinking RoI Selection for Deep Visual Speech Recognition			2020	ResNet
4	**WAS**	3	✓	Lip Reading Sentences in the Wild			2016	
5	**LipNet**	4.6	×	LipNet: End-to-End Sentence-level Lipreading			2016	

Future and Perspectives

Areas of reaserch :

- Implication of LLMs in the text produced by the last SOTA model in the aspect of early parkinsons decease detection - other medical anomalise as well
- LipNet In the Wild : Generalizable model that makes lipnet work under any condition with any language
- DeepFace gestures comunication : like LipReading but Face reading for patients with Jaw anomalise
- More Efficient and faster models are needed : Small Vision Language Models trained specifically for the task needed

THANK YOU FOR YOUR ATTENTION