Applied Intelligence (2022) 52:9001–9016
https://doi.org/10.1007/s10489-021-02846-w

FATALRead - Fooling visual speech recognition models

Put words on Lips

Anup Kumar Gupta[1] · **Puneet Gupta**[1] · **Esa Rahtu**[2]

Accepted: 14 September 2021 / Published online: 12 November 2021

Abstract
Visual speech recognition is essential in understanding speech in several real-world applications such as surveillance systems and aiding differently-abled. It proliferates the research in the realm of visual speech recognition, also known as Automatic Lip Reading (ALR). In recent years, Deep Learning (DL) methods are being utilised for developing ALR systems. DL models tend to be vulnerable to adversarial attacks. Studying these attacks creates new research directions in designing robust DL systems. Existing attacks on images and videos classification models are not directly applicable to ALR systems. Since the ALR systems encompass temporal information, attacking these systems is comparatively more challenging and strenuous than attacking image classification models. Similarly, compared to other video classification tasks, the region-of-interest is smaller in the case of ALR systems. Despite these factors, our proposed method, Fooling AuTomAtic Lip Reading (FATALRead), can successfully perform adversarial attacks on state-of-the-art ALR systems. To the best of our knowledge, we are the first to successfully fool ALR systems for the word recognition task. We further demonstrate that the success of the attack is increased by incorporating logits instead of probabilities in the loss function. Our extensive experiments on a publicly available dataset, show that our attack successfully circumvents the well-known transformation based defences.

Keywords Visual speech recognition · Deep learning · Automatic lip reading · Adversarial attacks

1 Introduction

Deep neural networks (DNN) have become unprecedentedly effective in solving several machine-learning tasks such as autonomous navigation, face recognition, natural language processing, audio analysis and fraud detection [1]. These applications have crucial security and financial impacts; hence they must be resilient against engineered attacks targeting to influence the system behaviour. Unfortunately, these applications behave peculiarly in the presence of adversarial examples [2, 3]. Adversarial examples are deliberately crafted by an adversary to make the DNN models behave in an unintended way. These examples are fabricated by adding well crafted and human imperceptible noises in the original input. Presence of such examples may lead to catastrophic results in security-critical real-world applications. For example, in the case of autonomous car driving systems, adversarial examples could enable an adversary to cause the vehicle to take unwanted actions [4]. Similarly, an attacker can fool a face recognition system for identity theft or to jeopardise surveillance systems [5]. Understanding the generation of adversarial attacks creates new research directions in designing robust deep learning (DL) systems [6–8] and exploring explainable artificial intelligence [9] without sacrificing accuracy. Hence, it is natural that there is a surge in active research on designing adversarial attacks and defences for the DL applications. A similar behaviour can be seen in the field of image classification task where the adversarial attacks and defences are extensively studied [10].

Visual speech recognition, which is also known as Automatic Lip Reading (ALR), finds its application in diverse and novel artificial applications such as (i) speech synthesiser for people with speech impairment based on their lip movements [11]; (ii) multi-view mouth rendering

✉ Anup Kumar Gupta
msrphd2105101002@iiti.ac.in

[1] Department of Computer Science and Engineering, IIT Indore, Indore, India

[2] Computer Vision Group, Tampere University, Tampere, Finland

as assistance to people with hearing disabilities [12]; (iii) lip motion silent passwords [13]; (iv) audio-less videos transcriber and re-dubber [14]; (v) speech recognition under noisy conditions [15]; (vi) isolation of individual speakers from multi-talker simultaneous speech [16]; and (vii) extracting speech from surveillance videos for forensic study [17]. An adversarial attack on the ALR system can have profound implications such as distress to differently-abled and security breaches in surveillance systems. But attacking an ALR system is more challenging than attacking an image classification network, as the videos contain an additional temporal dimension as compared to that of a 2-D image. Despite this, performing an adversarial attack on an ALR system will create a new research direction for designing robust ALR systems. This provides us with the motivation for proposing an attack against the ALR systems. We propose a white-box attack to fool word-level ALR systems, called as Fooling AuTomAtic Lip Reading (*FATALRead*) attack. The main contributions of our work include:

1. To the best of our knowledge, we are the first one to study adversarial attacks on Lip Reading Models for the word recognition task. Traditional image based attacks are not directly applicable on ALR systems due to the presence of the temporal dimensions in videos. Similarly, existing attacks on videos cannot be directly applied to ALR systems due to a smaller region-of-interest in these systems, which lead to perceivable perturbations.
2. *FATALRead* attack successfully fools the state-of-the-art ALR systems based on sequential and temporal convolutional architectures.
3. Our proposed attack is capable of crafting adversarial examples in both targeted and untargeted scenarios, even circumventing popular transformation based defences such as feature squeezing and JPEG compression [18, 19].
4. Our performed experiments show the vulnerability of the sequential and temporal convolutional architectures to an adversarial attack in the realm of ALR. Our results are conducted on the publicly available dataset.

The rest of the paper is organised as follows. In Section 2, we provide a brief overview of the adversarial attacks and existing ALR systems. We provide the preliminaries of adversarial attacks in Section 3. We then present our proposed *FATALRead* attack in Section 4. We discuss the experimental results and evaluate *FATALRead* attack in Section 5. In Section 6, we conclude by summarising our contributions and providing the future research directions.

2 Related work

This section outlines well-known adversarial attacks for DL-based image and video classifiers followed by a description of existing ALR systems for a better understanding of the proposed attack.

2.1 Adversarial attacks on image classification

The behaviour of DL-based image classifiers in the presence of adversarial examples have been extensively studied. Szegedy et al. [20] were the first ones to show that the image classifiers can be attacked by adding imperceptible perturbations to the input. A simple yet powerful white-box adversarial attack is the Fast Gradient Sign Method (FGSM) by [2], which exploits the linear nature of Convolutional neural networks (CNNs). It calculates the perturbation by using the sign of the gradient of the loss with respect to the input and adds it to the image in the direction that maximises the loss value. The iterative version of FGSM, Iterative Gradient Sign Method (IGSM), is introduced in [21] where FGSM is applied multiple times using a small step size instead of taking a single step. Projection Gradient Descent [22], attempts to maximise the loss while keeping the perturbation small than a specified value ϵ. It starts from a random perturbation in a ϵ-ball, with the centre as the original input and takes a step in the direction which maximises the loss. The perturbation is projected back to the ϵ-ball if necessary. DeepFool was proposed in [23], which finds the closest distance from the given input to the decision boundary of the intended adversarial example. A universal attack was proposed in [24], which is capable of generating perturbations that universal with respect to both the data and the network architectures. C&W attack is proposed by [25] where the perturbations δ are restricted using the distance metrics δ_0, δ_2 or δ_∞. The method was found to be effective in breaching several defences, including defensive distillation [26].

2.2 Adversarial attacks on video models

Adversarial attacks for video classification has been explored relatively less as compared to that of image classification. The first proposed method utilises $\ell_{2,1}$ norm based optimisation for generating sparse adversarial perturbations for video action recognition models [27]. It explores the propagation of adversarial perturbations across video frames. An untargeted adversarial attack was proposed in [28] for fooling the optical flow-based action recognition systems. It generated the adversarial videos using Generative Adversarial Networks (GANs) based

architecture. Another way to attack the video classifiers is proposed in [29] where adversarial frames are added in the video instead of adding perturbations to the frames. It appends few dummy frames at the end of the video and then perturbations are added only to the new dummy frames. The attack can also be performed using dubbed Adversarial Framing (AF) [30], which keeps the image unchanged and only adds an adversarial frame on the border of the image. Video classification systems can also be fooled using flickering adversarial attack [31] which adds a uniform RGB offset to the entire frame. The offset is different for each frame of the video. It is based on the assumption that the perceptible perturbations can be mistaken as a change in lighting conditions or a typical sensor behaviour.

2.3 Lip-reading models

Humans use visual cues in addition to listening for speech understanding. These cues are often picked up subconsciously, and its extent of usage depends on the clarity of the audio channel and the degree of the hearing quality. People with hearing impairment totally rely on visual information, such as speaker's lip movement and facial expressions, to understand speech. In essence, ALR model not only just complements audio in speech recognition for noisy acoustic mediums but also plays a solitary decisive role in several real-world applications like aiding the hearing impaired and in surveillance.

The traditional ALR models used hand-engineered features to extract the visual features followed by classifications using Hidden Markov Model (HMM) or Support Vector Machine (SVM) as reported in [32]. In recent years, there is a shift of interest towards the DL techniques because they automatically learn the relevant features to improve the performance significantly as compared to the traditional methods. In [33], a Deep belief network was utilised for a continuously spoken digit recognition task and it demonstrated significant performance improvement over a baseline multi-stream audio-visual GMM/HMM system. An encoder-decoder architecture with attention is utilised in [14] for sentence level classification.

Features extracted using deep encoders are utilised in [34], followed by Long-Short Term Memory (LSTM) networks for modelling the temporal dynamics. Bidirectional LSTM (Bi-LSTM) is found to be more suitable than LSTM in various sequence modelling tasks [35] because LSTM only utilises the past context but Bi-LSTM utilises both the preceding and succeeding contexts. Hence, ALR system for word-level classification proposed in [36] uses 3D convolutional layer and residual network for feature extraction, followed by Bi-LSTM network for modelling the temporal dynamics. In [37], a similar network was proposed for audiovisual speech recognition that uses bidirectional Gated Recurrent units (Bi-GRUs) network instead of Bi-LSTM network.

These sequential networks tend to perform notably better than their traditional counterparts. However, they require large networks composed of a large number of parameters and hence, they are costly both in terms of time and space. Hence, they tend to be slower in both training and inference [38]. This issue is mitigated in [39] by replacing sequence network with temporal convolutional network (TCN), which provides better performance in a cost-effective manner. TCN is a variant of CNN that combines dilations and residual connections with causal convolutions. Unlike sequence networks, it can analyse a long input sequence as a whole. It is shown in [39] that TCN significantly reduces the memory requirement as compared to sequence networks. Based on this, an ALR system is recently proposed in [40] that utilises TCN. It not only achieves state-of-the-art accuracy but takes considerably less training and inference times. To the best of our knowledge, this model is the state-of-the-art for word recognition task using visual stream only, on Lip Reading in the Wild (LRW) dataset.

3 Background

Let us consider a DNN which maps the input attributes (or features) X, to the output labels (or classes) Y, by learning a non-linear mapping $f_\theta : X \rightarrow Y$. An adversarial example $\tilde{X}$ is designed by adding a well crafted imperceptible perturbations δ to the input X, such that the classification of $\tilde{X}$ is incorrect. That is, the goal is to minimise δ, such that:

$$\tilde{X} = X + \delta;\ f_\theta(\tilde{X}) = \tilde{y};\ f_\theta(X) = y;\ \text{and}\ \tilde{y} \neq y$$

where $\tilde{y}$ is the classifier prediction for $\tilde{X}$. If $\tilde{y}$ corresponds to a specific target class label provided by the attacker, then the attack is known as targeted attack. Alternatively, a classifier can also be fooled by just misclassifying the output, that is, setting $\tilde{y} \neq y$. Such attacks are known as untargeted attacks. Adversarial attacks can be classified as: (i) black-box, where the adversary does not have access to the network architecture, parameters, training data, loss and activation functions of the DNN; (ii) white-box, where the attacker has knowledge of network architecture and parameters; and (iii) grey box attacks, where the attacker has limited knowledge of the model parameters.

4 Proposed attack

In this section, we present our proposed attack *FATALRead*, to fool the ALR systems based on both sequential and TCN architectures. This section is divided into three

subsections. Our *FATALRead* attack for sequential and TCN architectures are described in Sections 4.1 and 4.2 respectively. The ways to resolve the implementation issues and successfully performing the attack, are provided in the last subsection.

4.1 Attack on sequential architecture

In this subsection, we consider ALR systems as a combination of CNN architecture followed by a sequence model. For a better understanding of *FATALRead* attack, assume that the output of the ALR system for an input video V is w_c and the output $w_c \in \mathcal{S}$ where $\mathcal{S}$ is the set of all possible outputs (which are words in our case). Also, assume that the total number of words in the vocabulary, $|\mathcal{S}| = m$. This model predicts the word w_c, when its log probability achieves the maximum value, that is,

$$\begin{aligned} w_c &= \arg\max_{w_k \in \mathcal{S}} \log P(w_k|V) \\ \text{or } \log P(w_c|V) &= \max_{w_k \in \mathcal{S}} \log P(w_k|V) \\ &= \max_{k \in [0,m]} \log P(W^{(k)}|V) \end{aligned} \tag{1}$$

Note that the term $\log P(W^{(k)}|V)$ in (1) denotes the log probability of the k^{th} word in the set $\mathcal{S}$. More generally, the vector $\log P(W|V)$ denotes log probability of all the words in the vocabulary $\mathcal{S}$, which can be calculated as the vector sum of the log probabilities at every time step t of the RNN.

$$\log P(W|V) = \sum_{t=1}^{n} \log P(W_t|V, W_{t-1}) \tag{2}$$

In sequential architectures, $P(W_t|V, W_{t-1})$ at every step t is calculated by applying *softmax* to the logits at that time step, Z_t (refer Fig. 1). The logits Z_t is computed by a sequential unit (which can be GRU, LSTM or Bi-LSTM depending on the network architecture) $r(\cdot)$, using the previous hidden state h_{t-1} and features of input V_f as extracted by CNN frontend:

$$P(W_t|V, W_{t-1}) = \text{softmax}(Z_t) \text{ and } Z_t = r(h_{t-1}, V_f) \tag{3}$$

Rewriting the (3), using the definition of the softmax and applying log operation on both the sides of the equation, we get

$$\begin{aligned} \log P(W_t|V, W_{t-1}) &= \log \left(\exp(Z_t) / \sum_{i \in \mathcal{S}} \exp\left(Z_t^{(i)}\right) \right) \\ &= Z_t - \log \sum_{i \in \mathcal{S}} \exp\left(Z_t^{(i)}\right) \end{aligned} \tag{4}$$

Rearranging (1), (2) and (4), gives

$$\begin{aligned} \log P(w_c|V) &= \max_{k \in [0,m]} \sum_{t=1}^{n} \log P(W_t^{(k)}|V) \\ &= \max_{k \in [0,m]} \sum_{t=1}^{n} \left[Z_t^{(k)} - \log \sum_{i \in \mathcal{S}} \exp\left(Z_t^{(i)}\right) \right] \end{aligned} \tag{5}$$

Since the second term in RHS of (5) is constant with respect to w_k, we formulate log probability for the correct output w_c on an input V in terms of logits Z, as

$$\log P(w_c|V) = \max_{k \in [0,m]} \sum_{t=1}^{n} Z_t^{(k)} \tag{6}$$

We use the formulation in (6) for crafting targeted and untargeted adversarial attacks in the following manner:

1. *Targeted Attack:* We now introduce our adversarial attack targeted settings. For better understanding, consider that we aim to replace the target output to w_T by adding perturbations δ in V. This can be achieved by maximising the log probability of the target w_T for the perturbed input $\tilde{V}$. That is,

$$\log P(w_T|\tilde{V}) = \max_{k \in [0,m]} \sum_{t=1}^{n} Z_t^{(k)} \text{ such that } \tilde{V} = V + \delta \tag{7}$$

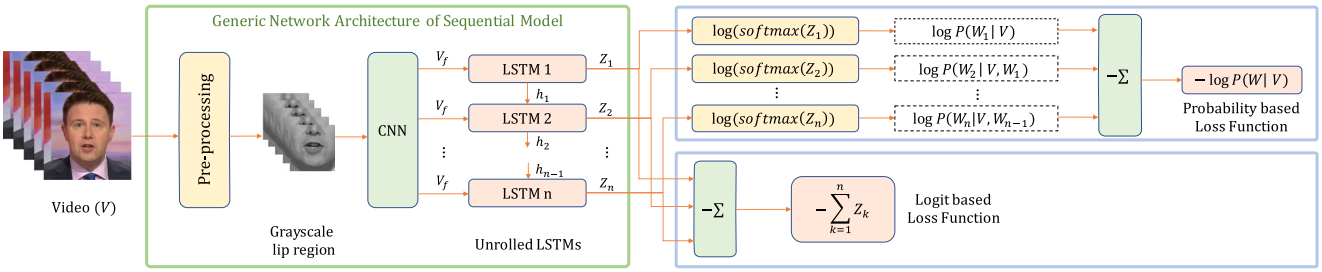

Fig. 1 Overview of our proposed *FATALRead* attack for generic sequential network architecture. We define two different loss functions based on probability and logit respectively. The desired perturbations are calculated by backpropagating the loss function. The model takes input video V, which is then preprocessed and fed to a CNN. The resultant features V_f, are then fed to an LSTM network. The LSTM at every time step t uses the previous hidden state h_{t-1} and extracted features V_f to produce logits Z_t. The proposed method could be easily extended for bi-directional sequential networks

A naive way to maximise the log probability of the target w_T is to make the value of corresponding logit $Z^{(k=T)}$ as large as possible. But this can compromise the imperceptibility of video $\tilde{V}$ when δ becomes too large. Thus, we ensure that the target w_T attains the largest logit compared to all other words in vocabulary $\mathcal{S}$. To this end, we define our loss function $J_t(\cdot)$ as the cross-entropy loss over the sum of the output logits and a one-hot encoding vector H. That is,

$$J_t(\tilde{V}, w_T) = \text{Cross-Entropy}\left(\sum_{t=1}^{n} Z_t, H^{[k=T]}\right) \tag{8}$$

where the notation $[k = T]$ denotes that the one-hot encoding vector H has a value 1 when $k = T$ and 0 otherwise. For minimising the loss function $J_t(\cdot)$, we iteratively perturb the input video guided by the gradients of the loss function $J_t(\cdot)$ with respect to the input, such that at every iteration i:

$$\tilde{V}_i = clip\left[V_i + \epsilon \cdot \text{sign}\left(\nabla_V J_t\left(\tilde{V}_i, w_T\right)\right)\right] \tag{9}$$

where ϵ, sign and ∇_V denote a multiplier to ensure the perturbations are small, operator to get the sign of a number and the gradient operation with respect to V, respectively. Note that the clipping is done to ensure that the range of the perturbed input $\tilde{V}_i$ remains intact.

2. *Untargeted Attack:* It is relatively easier to craft an adversarial example in an untargeted setting as compared to targeted one. Our aim here is perturb the video V such that the model predicts any arbitrary output w_{UT} such that $w_{UT} \neq w_c$. Mathematically, this can be achieved by modifying the (6) as

$$\log P(w_c|\tilde{V}) \neq \max_{k\in[0,m]} \sum_{t=1}^{n} Z_t^{(k)} \text{ such that } \tilde{V} = V + \delta \tag{10}$$

In other words, we ensure that the log probability of the ground-label w_c is not maximum. In such a case, there exist at least one label $k \neq c$, whose logits $Z^{(k)}$ is greater than the logits corresponding to the ground-label $Z^{(c)}$. We achieve this by defining our loss function $J_u(\cdot)$, as the negative of cross-entropy loss over the sum of the output logits and a one-hot encoding vector H. That is,

$$J_u(\tilde{V}, w_c) = -\text{ Cross-Entropy}\left(\sum_{t=1}^{n} Z_t, H^{[k=c]}\right) \tag{11}$$

Subsequently, we minimise the loss function $J_u(\cdot)$ by iteratively perturbing the input video using by the gradients of the loss function $J_u(\cdot)$ with respect to the input. That is, we replace $J_t(\cdot)$ with $J_u(\cdot)$ in (9) and utilise it for minimising the loss function in case of untargeted attack.

Kindly note that we have formulated our loss functions in (8) and (11) using logits instead of probabilities because it is shown in [25] that logits provide better attacking capabilities than probabilities for CNN-based image classifiers. Nevertheless, one can easily replace the logits with the probabilities to define our loss functions in terms of probabilities. For rigorous analysis, we conduct our experiments using both logit and probability based loss functions. Furthermore, ALR systems have started using Bi-LSTM [36] and Bi-GRU [41] in the recent years. Our *FATALRead* attack could be easily extended to such models with a slight modification. The log probability, as opposed to (2), in these cases are computed using:

$$\log P(W|V) = \sum_{t=1}^{n} [\log P(W_t|V, W_{t-1}) + \log P(W_t|V, W_{t+1})] \tag{12}$$

4.2 Attack on TCN architecture

In this section, we discuss the *FATALRead* attack for ALR systems based on TCN architecture. It is important to understand that the logits are obtained at every time step in the sequential models and thus, they should be consolidated to perform the adversarial attack on the sequential model. As an instance, we consolidate the logits at each time step by adding them, in (8) and (11). In contrast, the TCN based architectures provide the logits in a single step. We utilise the logits Z provided by the TCN based architectures to define the loss functions for targeted and untargeted settings, just like we define the loss functions in Section 4.1 for sequential architectures. Mathematically, our loss function $\bar{J}_t$ in targeted setting for TCN based architecture is:

$$\bar{J}_t(\tilde{V}, w_T) = \text{Cross-Entropy}\left(Z, H^{[k=T]}\right) \tag{13}$$

where $\tilde{V}$ is the perturbed video, w_T is the targeted word and H is a one-hot encoding vector with value 1 when $k = T$. Similarly, our loss function, $\bar{J}_u$ in untargeted setting for TCN based architecture is:

$$\bar{J}_u(\tilde{V}, w_c) = -\text{ Cross-Entropy}\left(Z, H^{[k=c]}\right) \tag{14}$$

where w_c denotes the ground-truth label. Kindly note that the untargeted attacks are performed by reducing the confidence in ground-truth label prediction until the model prediction is wrong. This can be achieved by utilising the negative of cross-entropy loss over the logits Z and a one-hot encoding vector H which contains the value 1 when $k = c$. Our *FATALRead* attack the ALR systems based on TCN architecture by minimising these loss functions. That is, we replace $J_t(\cdot)$ with $\bar{J}_t(\cdot)$ or $\bar{J}_u(\cdot)$ in (9) to generate

the adversarial example for targeted or untargeted setting, respectively (Fig. 2).

4.3 Implementation details

Most of the popular ALR systems use OpenCV [42] for preprocessing their input V. Some such preprocessing operations are colour space conversions (like RGB to grayscale) and image transformations (like centre or random crop). Unfortunately, OpenCV operations are non-differentiable and hence we cannot back-propagate through the preprocessing steps in order to calculate the gradients. Therefore, we implement these non-differentiable operations using the library Kornia [43, 44] which is an open-source computer vision (CV) library consisting of differential modules to solve generic CV problems. It is important to note that the modified system performs similarly to the original ALR system. Moreover, the adversarial videos generated using the modified models successfully fools the original ALR system.

We choose the value of maximum allowable perturbations (that is, intensity change) for a given pixel, in a single iteration, ϵ as 1, when the range of pixel intensity is [0,255]. We avoid selecting $\epsilon < 1$ as image intensities are integers, and thereby, such selection can disallow any change to the original image. On the other hand, choosing the value of $\epsilon > 1$, will introduce the possibility that we may overlook a point that could give similar results with fewer perturbations.

We choose the overall maximum distortion ℓ_∞ to be 10. That is, the maximum intensity difference for a pixel between the actual and adversarial input cannot exceed 10. Since the maximum possible distortion and distortion at every step is 10 and 1, respectively, an obvious choice for the number of iterations is 10.

5 Experimental results

5.1 Dataset

We conduct our experiments on publicly available LRW dataset[1] [45]. We utilise only the testing set of LRW, which consists of 50 utterances of 500 different words. These videos belong to over 1,000 different speakers, and each video is of 1.16 seconds containing 29 frames. Note that we consider only those videos for experiments that were correctly classified by the ALR system. The complete list of classes can be found in the Supplementary Document (Online Resource 1).

[1]Dataset Link: www.robots.ox.ac.uk/∼vgg/data/lip_reading/lrw1.html (For non-commercial individual research and private study use only. BBC content included courtesy of the BBC.)

5.2 Threat model

We choose the following state-of-the-art ALR systems based on sequential[2] and TCN[3] architectures models as threat models, for our experiments. Kindly note that the input given to both the models is a coloured 1.16 seconds long video with the dimensions as 29 (number of frames) × 256 (height) × 256 (width) × 3 (number of channels). The output of the models is the prediction of a word being uttered in the video.

1. *Sequential architecture* [37]*:* The network first extracts the mouth region from the video and transforms it to gray scale for reducing the time computation and network parameters. The resultant is fed to the network, which is formed by stacking a spatio-temporal convolution layer, a 34-layered residual network (ResNet) and a 2-layered Bi-GRU. This is an end-to-end audiovisual speech recognition model, but we consider the *video-only* model for conducting the experiments.
2. *TCN architecture* [40]*:* In this network, there is an additional preprocessing step as compared to [37], which performs face alignment after face detection. The frames of the input video are aligned to a reference mean face shape. It is followed by cropping of a fixed 96×96 pixels wide ROI and transforming of the cropped RGB frames to gray level. The preprocessed frames are then fed to 3D convolutional layer followed by a 18-layered ResNet and then finally to a multiscale TCN.

5.3 Performance metrics and ablation study

We use the following performance metrics to understand the efficacy of *FATALRead* attack: i) success rate which is given by percentage of successfully breached cases; ii) average Chebyshev distance which is given by averaging the adversarial noise δ_∞ over all the successful adversarial examples; and iii) average pixel-level Euclidean distance which is given by averaging the pixel-level root-mean-square distances δ_2 between the original and adversarial videos, over all the successful adversarial examples. Kindly note that the noise δ_∞ is given by the maximum intensity difference between the actual and adversarial input. We adopt the definition of δ_2 from [20], which is given as:

$$\delta_2 = \sqrt{\frac{\sum_{i=1}^{N} \left(\tilde{V}_i - V_i\right)^2}{N}}$$

[2]Available at: https://github.com/mpc001/end-to-end-lipreading

[3]Available at: https://github.com/mpc001/Lipreading_using_Temporal_Convolutional_Networks

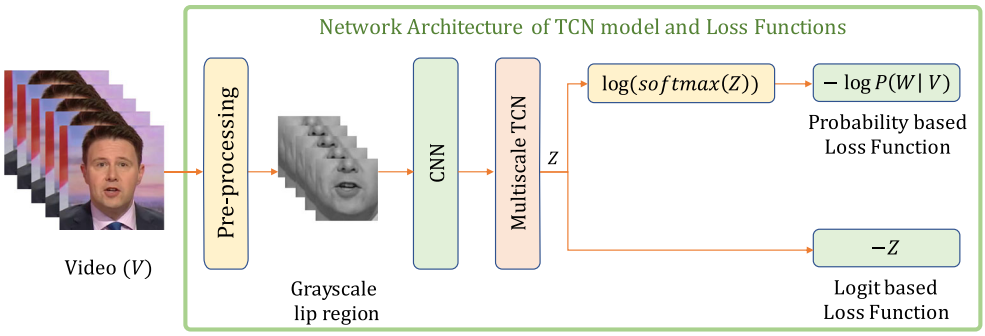

Fig. 2 Overview of our proposed *FATALRead* attack for ALR networks with TCN architecture. The model takes input a video V, which after preprocessing is passed on to a CNN, which extracts video features and feeds to a multiscale TCN. The multiscale TCN, in a single time step, outputs logits Z, which is further used to calculate the loss. We propose loss functions based on logit and probability

where $N = height \times width \times frames$, is the total number of pixels in the video. Intuitively, a higher success rate and lower average distortions imply an effective attack.

For more rigorous experimentation, we apply our *FATALRead* attack on the ALR systems defended by the following popular transformation based defences for image classification:

1. Bit depth reduction: We reduce the 8 bit frames to 5 bits in our experiments, as suggested in [18].
2. JPEG compression: We perform the compression at quality level 75 (out of 100) for our experiments, as recommended in [19].
3. Mean filtering: The size of the filter is set to 3×3, as proposed in [18].
4. Gaussian filtering: Following the works of [21], we keep the size and standard deviation of the kernel as 3×3 and (1.5, 1.5) respectively.

Defences like image cropping-rescaling [46] and image quilting [47] require drastic image transformations and hence when applied to the frames of the input video, distorts the lip region leading to a sharp fall in the model accuracy. Therefore we do not include results on these defence strategies. We apply some recent defences such as Class-specific Image Inpainting Defence (CIIDefence) [48] to study their applicability on ALR systems. While applying CIIDefence, we observe a peculiar behaviour. The changes made due to the defence algorithm is imperceptible when viewed frame-wise. However, when the reconstructed frame are combined as a video, the changes are easily noticeable. This is due to the temporal dimension present in videos. The model accuracy dropped to approximately 5% and hence we conclude that CIIDefence cannot be directly extended to ALR systems.

5.4 Performance analysis

To understand the efficacy of our *FATALRead* attack, we perform the attack using two targets for each input video. These target labels are the second most probable label and the least probable label as predicted by the model for the given input. Intuitively, for a targeted attack, the first case seems easier to be performed and the second one harder. Table 1 shows the comparative analysis of the *FATALRead* attack, evaluated on δ_∞. For ease of understanding, we will denote the sequential architecture [37] as A_1; the TCN architecture [40] as A_2; the targeted attacks where the target is the second probable label as Targeted$_1$; and the targeted attacks where the target is the least probable label as Targeted$_2$.

It can be observed from Tables 1 and 2 that our *FATALRead* attack performs better for the untargeted setting than both the targeted settings. This is due to the fact that in targeted settings, we aim to achieve a specific target, instead of a generic one. Similarly, it can be seen that *FATALRead* attack performs better in Targeted$_1$ setting than the Targeted$_2$ setting where aim is to achieve the least probable target. The average ℓ_∞ distortion is approximately 1 when our attack is performed in either untargeted setting or Targeted$_1$ setting but it lies in the range of 3 to 8 for Targeted$_2$ setting. Similarly, it can be seen from Table 2 that the average δ_2 distortions in Targeted$_2$ setting is significantly higher than distortions in either untargeted setting or Targeted$_1$ setting.

The average δ_∞ distortions in Targeted$_2$ setting for logits based loss function are 4.32282 and 3.39340 for A_1 and A_2 respectively. These are significantly lower than the average distortions in Targeted$_2$ setting for probability based loss function which are 7.68699 and 4.65394 for A_1 and

Table 1 Performance of *FATALRead* attack on A_1: Sequential [37] and A_2: TCN [40] architectures using both logits and probability based loss functions in the presence of several well-known defences, evaluated on δ_∞ noise

	Attack Setting			Untargeted		Targeted$^*_1$		Targeted$^+_2$	
Model type	Defence used	Loss metric	Model$^\#$ Accuracy (in %)	Attack success rate (in %)	Average Distortion $\boldsymbol{\delta}_\infty$	Attack success rate (in %)	Average distortion $\boldsymbol{\delta}_\infty$	Attack success rate (in %)	Average distortion $\boldsymbol{\delta}_\infty$
A_1: Sequential [37]	None	Logit	83.30	100.00	1.02297	100.00	1.12051	99.96	4.32282
		Probability		100.00	1.02311	99.99	1.12268	21.19	7.68699
	Bit Depth Reduction	Logit	25.84	100.00	1.00012	100.00	1.10686	100.00	3.07681
		Probability		100.00	1.00135	100.00	1.11057	51.63	6.48020
	JPEG Compression	Logit	25.68	100.00	1.00042	100.00	1.09817	100.00	3.15786
		Probability		100.00	1.00467	100.00	1.10020	50.10	6.57418
	Gaussian Filtering	Logit	82.93	100.00	1.02981	100.00	1.12592	99.91	4.45745
		Probability		100.00	1.02973	100.00	1.12882	20.28	7.74286
	Mean Filtering	Logit	82.90	100.00	1.02950	100.00	1.12557	99.92	4.45474
		Probability		100.00	1.02953	100.00	1.12895	20.11	7.74460
A_2: TCN [40]	None	Logit	85.30	100.00	1.07964	100.00	1.09640	99.98	3.39340
		Probability		100.00	1.07976	100.00	1.09887	99.91	4.65394
	Bit Depth Reduction	Logit	85.22	100.00	1.06466	100.00	1.14100	100.00	3.30806
		Probability		100.00	1.06470	100.00	1.14403	99.23	5.38093
	JPEG Compression	Logit	81.96	100.00	1.06091	100.00	1.13012	100.00	3.69875
		Probability		100.00	1.06096	100.00	1.13476	87.31	8.75492
	Gaussian Filtering	Logit	83.91	100.00	1.08099	100.00	1.08156	100.00	3.48360
		Probability		100.00	1.08108	100.00	1.08447	99.99	4.61108
	Mean Filtering	Logit	83.54	100.00	1.07819	100.00	1.07987	100.00	3.48063
		Probability		100.00	1.07809	100.00	1.08260	99.98	4.57449

*: Targeted attack using the second most probable label as the target

$^+$: Targeted attack using the least probable label as the target

$^\#$: The percentage of videos that are correctly classified by the ALR system. Some of defences lead to a sharp decline in the model accuracy for sequential ALR systems

Note: We perform the attacks on only those inputs which are correctly classified by the ALR system

Table 2 Performance of *FATALRead* attack on A_1: Sequential [37] and A_2: TCN [40] architectures using both logits and probability based loss functions in the presence of several well-known defences, evaluated on δ_2 noise

Model type	Attack Setting			Untargeted		Targeted$^{*}_{1}$		Targeted$^{+}_{2}$	
	Defence used	Loss metric	Model$^{\#}$ accuracy (in %)	Attack success rate (in %)	Average distortion $\boldsymbol{\delta_2}$	Attack success rate (in %)	Average distortion $\boldsymbol{\delta_2}$	Attack success rate (in %)	Average distortion $\boldsymbol{\delta_2}$
A_1: Sequential [37]	None	Logit	83.30	100.00	0.020656	100.00	0.059020	99.96	0.129298
		Probability		100.00	0.020811	99.99	0.059089	21.19	0.157817
	Bit Depth Reduction	Logit	25.84	100.00	0.020044	100.00	0.058311	100.00	0.092047
		Probability		100.00	0.020070	100.00	0.058515	51.63	0.193877
	JPEG Compression	Logit	25.68	100.00	0.020050	100.00	0.057853	100.00	0.094471
		Probability		100.00	0.020137	100.00	0.057969	50.10	0.196688
	Gaussian Filtering	Logit	82.93	100.00	0.021111	100.00	0.060046	99.91	0.187335
		Probability		100.00	0.021131	100.00	0.060210	20.28	0.325438
	Mean Filtering	Logit	82.90	100.00	0.021726	100.00	0.061554	99.92	0.264297
		Probability		100.00	0.021728	100.00	0.061748	20.11	0.459518
A_2: TCN [40]	None	Logit	85.30	100.00	0.011570	100.00	0.058156	99.98	0.104689
		Probability		100.00	0.011585	100.00	0.058206	99.91	0.121594
	Bit Depth Reduction	Logit	85.22	100.00	0.011410	100.00	0.060526	100.00	0.102061
		Probability		100.00	0.011415	100.00	0.060696	99.23	0.166027
	JPEG Compression	Logit	81.96	100.00	0.011370	100.00	0.059948	100.00	0.114112
		Probability		100.00	0.011371	100.00	0.060204	87.31	0.270119
	Gaussian Filtering	Logit	83.91	100.00	0.011748	100.00	0.055131	100.00	0.110248
		Probability		100.00	0.011757	100.00	0.055288	99.99	0.145945
	Mean Filtering	Logit	83.54	100.00	0.011939	100.00	0.052688	100.00	0.103762
		Probability		100.00	0.011944	100.00	0.052830	99.98	0.136385

*: Targeted attack using the second most probable label as the target

$^{+}$: Targeted attack using the least probable label as the target

$^{\#}$: The percentage of videos that are correctly classified by the ALR system. Some of defences lead to a sharp decline in the model accuracy for sequential ALR systems

Note: We perform the attacks on only those inputs which are correctly classified by the ALR system

A_2 respectively. Furthermore, the table demonstrates that our *FATALRead* attack achieves almost 100% success rate when logits are used to calculate the loss for both the architectures. But the attack success rate is 99.91% for A_2 and drops significantly to 21.19% for A_1. It indicates that even though the probability based loss function requires larger δ_∞ perturbations, they are incapable to provide a successful attack in 78.81% cases for A_1 model. Hence, it can be inferred that the performance is significantly improved when logits are used for the loss function instead of probabilities. Moreover, there is a noticeable difference in the attack success rate between the architectures A_1 and A_2. This behaviour is attributed to the fact that a CNN, when backed by a sequence model provides an inherent defence mechanism (or gradient masking) for several adversarial attacks by additionally encompassing temporal information and complex network sequence modelling architecture [25].

The success of *FATALRead* attack is highly dependent on the choice of ℓ_∞, especially in Targeted$_2$ attack. For illustration, we depict the success rate of Targeted$_2$ attack with respect to the maximum allowable distortion ℓ_∞ for the case of logits based loss function, in Fig. 3. It can be seen that the accuracy significantly increases when we increase ℓ_∞. Furthermore, when we set ℓ_∞ equal to 6, we can achieve almost 98% and 100% success rates for logit based loss function in A_1 and A_2 respectively. Kindly note that setting a high value of distortion distance will seriously affect the imperceptibility; hence we choose the value ℓ_∞=10 where our attack performs accurately without sacrificing the significant visual perceptibility of the attacked videos. Kindly note that the success rate is 21.19% for A_2 in Targeted$_2$ setting utilising probability based loss function, but it can be improved, as shown in Fig. 3, at the expense of deteriorating the visual perceptibility, that is, allowing larger values of l_∞. In addition, Fig. 3 indicates that our *FATALRead* attack will not be successful in most of the cases for probability based loss function even increasing ℓ_∞ to significant amount. Hence, it can be inferred that the efficacy of our *FATALRead* attack can be improved when logits based loss is used instead of probability based loss. Also, this difference is more prominent in the case of Targeted$_2$ setting where the least probable target labels are selected.

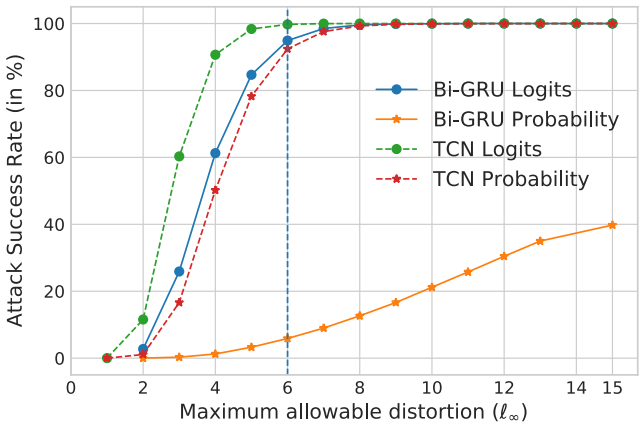

Fig. 3 Relationship between attack success rate and maximum allowable distortion (ℓ_∞) in Targeted$_2$ setting for both logit and probability based loss functions. Maximum allowable distortion denotes the upper limit of intensity difference between the original and adversarial inputs. The attack success rate is the percentage of such cases in which we were able to successfully fool the ALR system. We can observe that the logits based loss functions achieve a higher attack success rate with relatively lower distortions compared to that of probability based loss functions

It could be observed from Tables 1 and 2 that our *FATALRead* attack was able to circumvent the applied defences with an attack success rate of approximately 100%. We also observe that there is a drop in model accuracy when these defences are applied. The model accuracy for A_1 significantly deteriorates when the JPEG compression and bit depth reduction are applied. Furthermore, the attack success rate is decreased tremendously in case of A_1 as compared to A_2. It indicates that attacking A_1 is more challenging and strenuous than attacking A_2. Moreover, we found that while applying the defences, the drop in model accuracy in the case of a sequence architecture, A_1 is significantly higher compared to that of TCN architecture, A_2. This is due to the large number of parameters present in the sequential model, because of which these models tend to overfit. Such similar behaviour is observed by authors in [49] and [50].

It is evident from the results that *FATALRead* attack can successfully fool the state-of-the-art ALR systems by adding small adversarial perturbations. For visualisation, some adversarial examples generated by our *FATALRead* attack are depicted in Fig. 4. Each example shows a random clean frame from the video along with the corresponding perturbed frame. Since the perturbations only occur in the lip regions, we depict these areas separately and show the perturbations in RGB channels for these areas only. In the first example (first row), the ground-truth label for the video is "Information" but the model tends to predict "Major" after the attack. Note that the adversarial frames are visually similar to the clean frames. This is due to the fact that the adversarial frames are obtained by adding imperceptible noise to the clean frames. For better visualisation, kindly refer the Supplementary Animation (Online Resource 2). Due to the privacy constraints imposed by the owners of the LRW dataset, we are unable to include the sample videos of the dataset in the Supplementary Animation. Instead, we record our videos and try to predict the labels after passing them into the Automatic Lip Reading (ALR) system. We then perform *FATALRead* attack on those videos which were correctly classified by the ALR system.

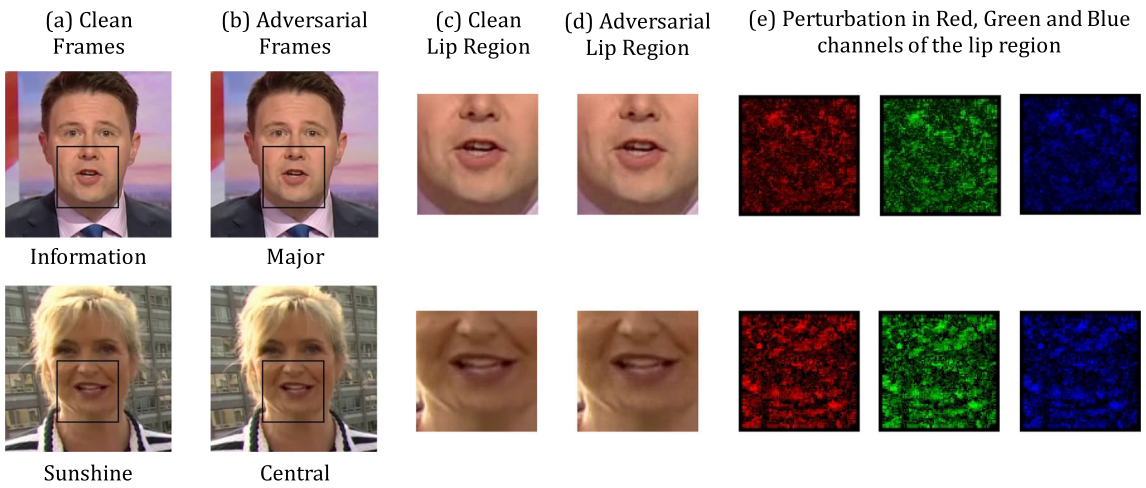

Fig. 4 Examples of adversarial frames generated by *FATALRead* in the untargeted settings on [37]. It demonstrates (**a**) the clean frames, (**b**) the corresponding adversarial frames, (**c**) lip regions in the clean frames, (**d**) lip regions in the adversarial frames and (**e**) the perturbations in each of the RGB channels. The perturbations are scaled using min-max normalisation for better visualisation. The adversarial frames are visually similar to the clean frames

5.5 Effect on temporal dimension

In this section, we discuss the effect of our *FATALRead* attack on the temporal dimension. We study the frame-wise perturbations added by the *FATALRead* attack to the videos for A_1 and A_2 architectures. For illustration, we depict the frame-wise average δ_2 and δ_∞ distortions for both the architectures in Fig. 5. It can be observed from the Fig. 5 that in the Targeted$_1$ setting, the perturbations are pre-dominantly added in either beginning or at the end of the videos. The degree of perturbation is lowest in the middle frames. Whereas, in the Targeted$_2$ setting, the perturbations are uniformly distributed throughout all the frames. We can also observe that the average distortion is significantly lower when the logit based loss function is employed instead of the probability based loss function. This difference between the distortion values is considerably greater in Targeted$_2$ setting compared to Targeted$_1$ setting. The reason for such behaviour is that it is more difficult to perform an attack in the Targeted$_2$ setting compared to Targeted$_1$ setting. We also

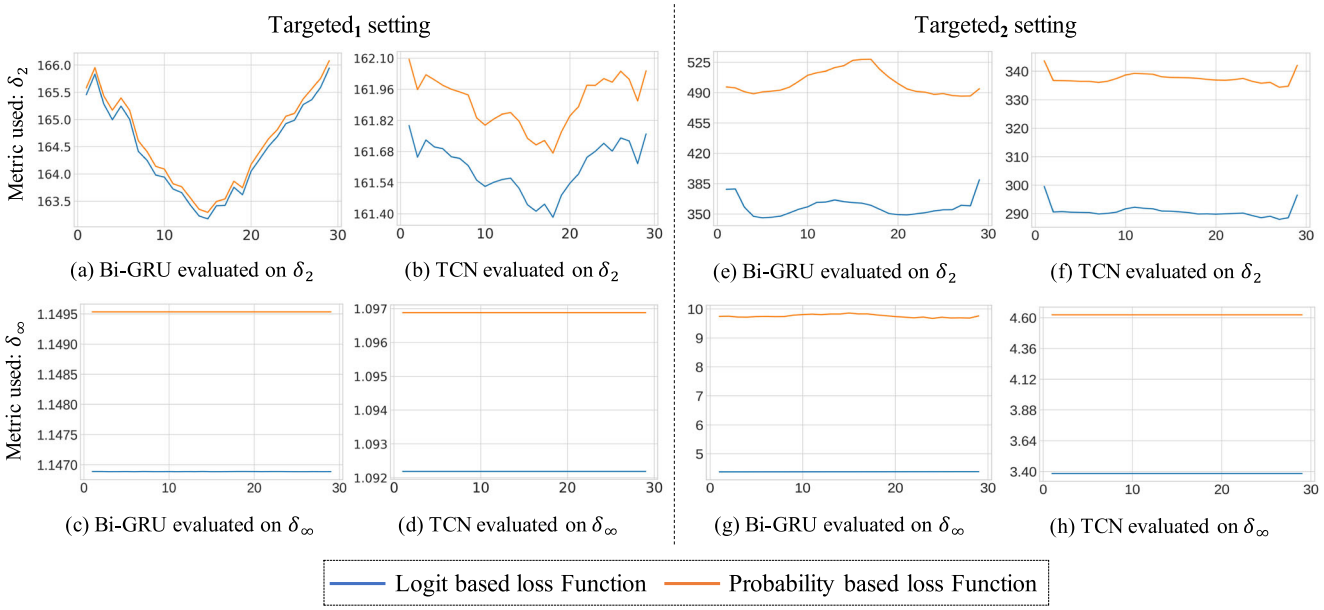

Fig. 5 Illustration of the average frame-wise δ_2 and δ_∞ distortions for A_1: Sequential [37] and A_2: TCN [40] architectures in Targeted$_1$ and Targeted$_2$ settings using both logits and probability based loss functions. The x-axis denotes the video frames indices and the y-axis denotes the average distortion values. We can observe that in Targeted$_1$ setting, the perturbations are pre-dominantly added in either beginning or at the ending frames of the video, whereas in the Targeted$_2$ setting, the perturbations are added uniformly throughout all the frames

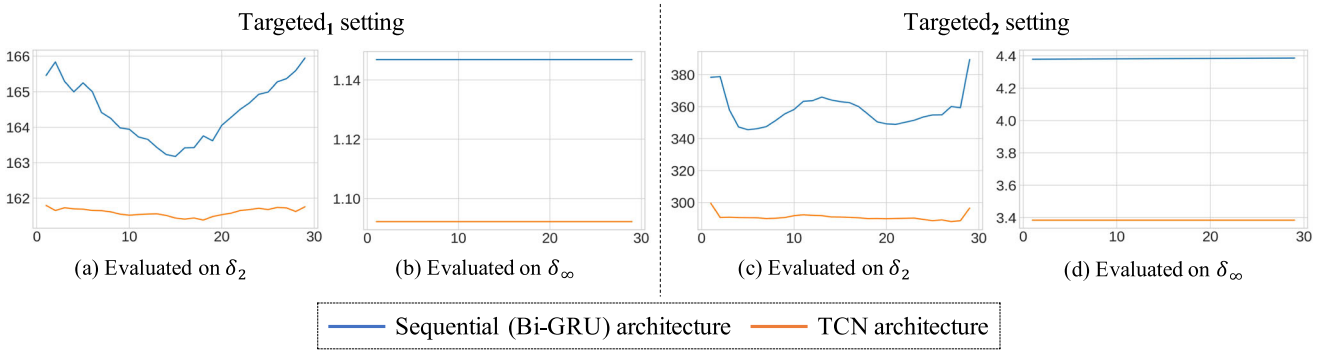

Fig. 6 Comparison of average frame-wise δ_2 and δ_∞ distortions in A_1: Sequential [37] and A_2: TCN [40] architectures using the novel logit based loss function in Targeted$_1$ and Targeted$_2$ settings. The x-axis denotes the video frames indices and the y-axis denotes the average distortion values. We can observe that the distortion values added for A_1 are significantly higher than the values added for A_2. This difference in distortion values is more prominent in the Targeted$_2$ setting compared to the Targeted$_1$ setting

perform a comparative analysis of how the proposed attack affects the temporal dimension of both the architectures A_1 and A_2. We can observe from the Fig. 6 that the distortion values added for A_1 are higher than the values added for A_2, in both Targeted$_1$ and Targeted$_2$ settings. It indicates that attacking A_1 is more challenging and strenuous than attacking A_2. This observation is in line with the results shown in Tables 1 and 2.

5.6 Discussion

In this section, we will discuss the efficacy of existing well-known attacks for image and video classification models when they are utilised for attacking the ALR systems:

1. **FGSM and IGSM** - FGSM is a one-step attack which is designed to be fast rather than optimal. Its iterative version IGSM adds finer perturbations at every step and performs better than FGSM [21]. *FATALRead*, along with probability based loss function, is an improvement to IGSM. Furthermore, when we utilise our novel logits based loss function instead of probabilities based loss function, the efficacy of *FATALRead* increases. Thus we expect better results compared to FGSM and IGSM attacks in ALR system.
2. **Carlini & Wagner** - C&W attack proposes to simultaneously minimise the added perturbations as well as make the network misclassify the same time [25]. *FATALRead* explicitly sets the perturbation to a minimum. We choose the value of maximum allowable perturbations for a given pixel, in a single iteration, $\epsilon = 1$. Setting a $\epsilon < 1$ can disallow any change to the original frame while saving it. Refer Section 4.3 for more details. Thus, the C&W attack will behave in a similar way as the proposed attack.
3. **GAN based attacks** - The existing GAN based attacks and defences aim at reconstructing the image, which is a challenging task considering the large dimensionality

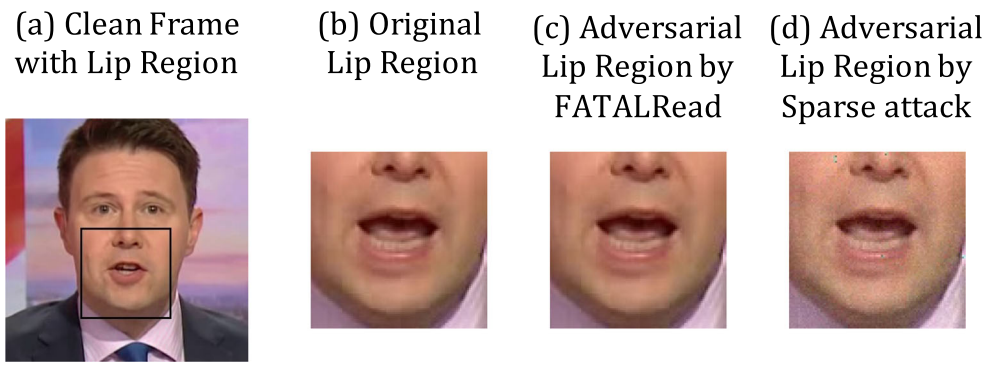

Fig. 7 Illustration of frames generated by the sparse attack [27] in Targeted$_2$ setting on [37]. It shows (**a**) a clean frame with the lip region enclosed by a border; (**b**) lip region in the clean frames; (**c**) lip region in the adversarial frames generated by the proposed *FATALRead* attack and (**d**) lip region in the adversarial frames generated by the sparse attack [27]. The perturbations added by the sparse attack are perceptible and significantly greater compared to the perturbations added by *FATALRead*

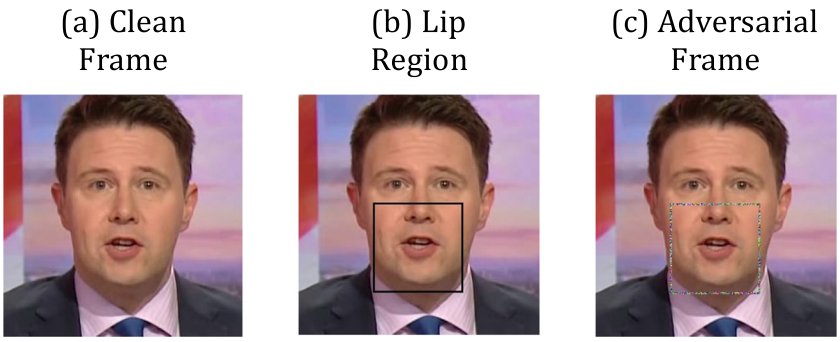

Fig. 8 Example of adversarial frames generated by AF [30] in untargeted setting on [37]. It shows (**a**) a clean frame, (**b**) a clean frame with the lip region enclosed by a border and (**c**) the corresponding adversarial frame. It demonstrates that the adversarial border added by [30] is perceptible to the human eye

of the input space. Moreover, in these approaches, the perceptibility of the reconstructed images decreases [48]. Furthermore, the attack will be computationally expensive because the domain of the ALR increases by many folds (as we have 19 images instead of 1).

4. **Video attacks** - When the adversarial attack is performed by adding sparse perturbations, the sparsity of perturbations can be easily perceived due to its unnatural pattern [31]. We extend the sparse attack proposed in [27] to ALR systems, with two sparsity values, 80% and 90%. We observe that if perturbations are added only in a few frames, the distortions are amplified. It happens because the dimensions of lip regions are small. It can be seen from Fig. 7 that perturbations added by the sparse attack are perceptible and significantly greater compared to the perturbations added by *FATALRead*. Similarly, attacking the ALR systems by appending spurious frames at the end of the video or adding adversarial borders around the lip region seems unnatural to the observers and can be easily perceived. One such example is shown in Fig. 8, where we extend the AF attack proposed in [30]. Furthermore, the applicability of universal attacks in our case is restricted because it requires access to the entire dataset to create adversarial frames or borders [51]. In contrast, our *FATALRead* is successful in creating adversarial examples for unseen samples. We further carried out experiments by adding a uniform RGB offset to the entire frame and adding the offset only to the desired lip region. We found that this "flickering" behaviour proved to be distracting to viewers. The issue was aggravated when the offset was applied only to the lip region.

FATALRead is a practical attack as it provides visually imperceptible adversarial videos which can be saved and reused to fool the ALR systems later. We can save the perturbed videos because the value of ϵ is set to 1 in our *FATALRead* attack (refer 4.3). In contrast, several previous works have set the value of ϵ less than 1, and hence, the perturbed videos cannot be saved properly due to quantisation error.

6 Conclusions

Visual speech recognition or ALR is an important tool for understanding speech in several real-world applications. Modern ALR systems mainly utilise DNN networks for achieving high performance, but they are vulnerable to adversarial attacks. However, attacking ALR systems is challenging and strenuous as they encompass temporal information. This motivated us to perform an attack on ALR systems. In this paper, we proposed a novel attack, named *FATALRead* attack, to fool the current state-of-the-art ALR models based on sequential and temporal convolutional architectures. Our conducted experiments have demonstrated that the success of crafting adversarial examples increases when logits are incorporated instead of probabilities in the loss function of our *FATALRead* attack. Moreover, we showed that our proposed attack successfully circumvents the popular defences of image classification.

The success of such attacks is rooted from the fact that while building a DNN model, the main focus is to improve its accuracy instead of making it resilient. We strongly believe that our work will create a new research direction in understanding, designing and defending the ALR systems without sacrificing the accuracy. This work is a starting point, and we will extend this work by further exploring the possibility of improving existing defences or building new defences to mitigate the proposed attack. We also wish to extend the work by studying adversarial attacks on audio-visual speech recognition models.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s10489-021-02846-w.

Acknowledgements We would like to thank the respective authors for providing code and pretrained models. We would also like to thank BBC for providing the Lip Reading Words in the Wild (LRW) dataset.

We are also thankful to the anonymous reviewers for their valuable suggestions to improve the quality of the paper. This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Funding This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Availability of data and material (data transparency) Dataset is publicly available at: www.robots.ox.ac.uk/~vgg/data/lip_reading/lrw1.html

Code availability (software application or custom code) Code is publicly available at: https://github.com/AnupKumarGupta/FATALRead-Fooling-Visual-Speech-Recogntion

Declarations

Conflicts of interest/Competing interests The authors declare that they have no conflict of interest.

References

1. Liu W, Wang Z, Liu X, Zeng N, Liu Y, Alsaadi FE (2017) A survey of deep neural network architectures and their applications. Neurocomputing 234:11–26. https://doi.org/10.1016/j.neucom.2016.12.038. https://www.sciencedirect.com/science/article/pii/S0925231216315533
2. Goodfellow IJ, Shlens J, Szegedy C (2015) Explaining and Harnessing Adversarial Examples. In: International Conference on Learning Representations, (ICLR). https://research.google/pubs/pub43405/
3. Gupta P, Rahtu E (2019) MLAttack: Fooling Semantic Segmentation Networks by Multi-layer Attacks. In: German Conference on Pattern Recognition (GCPR). Springer, pp 401–413. https://doi.org/10.1007/978-3-030-33676-9_28
4. Modas A, Sanchez-Matilla R, Frossard P, Cavallaro A (2020) Toward robust sensing for autonomous vehicles: An adversarial perspective. IEEE Signal Process Mag 37(4):14–23. https://doi.org/10.1109/MSP.2020.2985363
5. Goswami G, Agarwal A, Ratha N, Singh R, Vatsa M (2019) Detecting and mitigating adversarial perturbations for robust face recognition. Int J Comput Vis 127(6):719–742. https://doi.org/10.1007/s11263-019-01160-w
6. García J, Majadas R, Fern´andez F (2020) Learning adversarial attack policies through multi-objective reinforcement learning. Eng Appl Artif Intell 96:104021. https://doi.org/10.1016/j.engappai.2020.104021. https://www.sciencedirect.com/science/article/pii/S0952197620303043
7. Sun X, Sun S (2021) Adversarial robustness and attacks for multi-view deep models. Eng Appl Artif Intell 97:104085. https://doi.org/10.1016/j.engappai.2020.104085. https://www.sciencedirect.com/science/article/pii/S0952197620303419
8. Xu J, Du Q (2020) TextTricker: Loss-based and gradient-based adversarial attacks on text classification models. Eng Appl Artif Intell 92:103641. https://doi.org/10.1016/j.engappai.2020.103641. https://www.sciencedirect.com/science/article/pii/S0952197620300956
9. Marino DL, Wickramasinghe CS, Manic M (2018) An adversarial approach for explainable AI in intrusion detection systems. In: (IECON) Annual Conference of the IEEE Industrial Electronics Society. IEEE, pp 3237–3243. https://doi.org/10.1109/IECON.2018.8591457
10. Yuan X, He P, Zhu Q, Li X (2019) Adversarial examples: Attacks and defenses for deep learning. IEEE Trans Neural Netw Learn Syst 30(9):2805–2824. https://doi.org/10.1109/TNNLS.2018.2886017
11. Ephrat A, Halperin T, Peleg Shmuel (2017) Improved speech reconstruction from silent video. In: International Conference on Computer Vision Workshops (ICCV-W). IEEE, pp 455–462. https://doi.org/10.1109/ICCVW.2017.61
12. Fernandez-Lopez A, Sukno FM (2018) Survey on automatic lip-reading in the era of deep learning. Image Vis Comput 78:53–72. https://doi.org/10.1016/j.imavis.2018.07.002
13. Ezz M, Mostafa AM, Nasr AA (2020) A silent password recognition framework based on lip analysis. IEEE Access 8:55354–55371. https://doi.org/10.1109/ACCESS.2020.2982359
14. Chung JS, Senior A, Vinyals O, Zisserman A (2017) Lip reading sentences in the wild. In: Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, pp 3444–3453. https://doi.org/10.1109/CVPR.2017.367
15. Adeel A, Gogate M, Hussain A, Whitmer WM (2019) Lip-reading driven deep learning approach for speech enhancement. IEEE Trans Emerg Top Comput Intell:1–10. https://doi.org/10.1109/TETCI.2019.2917039
16. Ephrat A, Mosseri I, Lang O, Dekel T, Wilson K, Hassidim A, Freeman WT, Rubinstein M (2018) Looking to listen at the cocktail party: a speaker-independent audio-visual model for speech separation. ACM Trans Graph 37(4):112:1–112:11. https://doi.org/10.1145/3197517.3201357
17. Rothkrantz L (2017) Lip-reading by surveillance cameras. In: Smart City Symposium Prague (SCSP). IEEE, pp 1–6
18. Xu W, Evans D, Qi Y (2018) Feature Squeezing: Detecting Adversarial Examples in Deep Neural Networks. In: Network and Distributed Systems Security Symposium (NDSS). https://wp.internetsociety.org/ndss/wp-content/uploads/sites/25/2018/02/ndss2018_03A-4_Xu_paper.pdf
19. Dziugaite GK, Ghahramani Z, Roy DM (2016) A study of the effect of JPG compression on adversarial images. arXiv:1608.00853
20. Szegedy C, Zaremba W, Sutskever I, Bruna J, Erhan D, Goodfellow IJ, Fergus R (2014) Intriguing properties of neural networks. In: Bengio Y, LeCun Yx (eds) International conference on learning representations, ICLR. https://research.google/pubs/pub42503.pdf
21. Kurakin A, Goodfellow I, Bengio S (2017) Adversarial examples in the physical world. In: International Conference on Learning Representations (ICLR). https://openreview.net/forum?id=HJGU3Rodl
22. Madry A, Makelov A, Schmidt L, Tsipras D, Vladu A (2018) Towards deep learning models resistant to adversarial attacks. In: International conference on learning representations, ICLR. https://openreview.net/forum?id=rJzIBfZAb
23. Moosavi-Dezfooli S, Fawzi A, Frossard P (2016) Deepfool: A simple and accurate method to fool deep neural networks. In: IEEE conference on computer vision and pattern recognition, CVPR. IEEE Computer Society, pp 2574–2582. https://doi.org/10.1109/CVPR.2016.282
24. Moosavi-Dezfooli S, Fawzi A, Fawzi O, Frossard Pa (2017) Universal adversarial perturbations. In: IEEE conference on computer vision and pattern recognition, CVPR. IEEE Computer Society, pp 86–94. https://doi.org/10.1109/CVPR.2017.17

25. Carlini N, Wagner D (2017) Towards evaluating the robustness of neural networks. In: IEEE Symposium on Security and Privacy (SP). IEEE, pp 39–57. https://doi.org/10.1109/SP.2017.49
26. Papernot N, McDaniel P, Wu X, Jha S, Swami A (2016) Distillation as a defense to adversarial perturbations against deep neural networks. In: IEEE Symposium on Security and Privacy (SP). IEEE, pp 582–597. https://doi.org/10.1109/SP.2016.41
27. Wei X, Zhu J, Yuan S, Su H (2019) Sparse adversarial perturbations for videos. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol 33. AAAI Press, pp 8973–8980. https://doi.org/10.1609/aaai.v33i01.33018973
28. Inkawhich N, Inkawhich M, Chen Y, Li H (2018) Adversarial attacks for optical flow-based action recognition classifiers. arXiv:1811.11875
29. Chen Z, Xie L, Pang S, He Y, Tian Q (2021) Appending adversarial frames for universal video attack
30. Zajac M, Zołna K, Rostamzadeh N, Pinheiro PO (2019) Adversarial framing for image and video classification. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol 33. AAAI Press, pp 10077–10078. https://doi.org/10.1609/aaai.v33i01.330110077
31. Pony R, Naeh I, Mannor S (2020) Over-the-air adversarial flickering attacks against video recognition networks. arXiv:2002.05123
32. Hao M, Mamut M, Yadikar N, Aysa A, Ubul K (2020) A survey of research on lipreading technology. IEEE Access 8:204518–204544. https://doi.org/10.1109/ACCESS.2020.3036865
33. Vakhshiteh F, Almasganj F, Nickabadi A (2018) Lip-reading via deep neural networks using hybrid visual features. Image Anal Stereol 37(2):159–171. https://doi.org/10.5566/ias.1859. https://www.ias-iss.org/ojs/IAS/article/view/1859
34. Petridis S, Pantic M (2016) Deep complementary bottleneck features for visual speech recognition. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp 2304–2308. https://doi.org/10.1109/ICASSP.2016.7472088
35. Liu G, Guo J (2019) Bidirectional LSTM with attention mechanism and convolutional layer for text classification. Neurocomputing 337:325–338. https://doi.org/10.1016/j.neucom.2019.01.078
36. Stafylakis T, Tzimiropoulos G (2017) Combining residual networks with LSTMs for lipreading. In: International Speech Communication Association (INTERSPEECH), pp 3652–3656. https://www.isca-speech.org/archive/Interspeech_2017/abstracts/0085.html
37. Petridis S, Stafylakis T, Ma P, Cai F, Tzimiropoulos G, Pantic M (2018) End-to-end audiovisual speech recognition. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, pp 6548–6552. https://doi.org/10.1109/ICASSP.2018.8461326
38. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin I (2017) Attention is all you need. In: Advances in neural information processing systems (NIPS), pp 5998–6008. https://papers.nips.cc/paper/7181-attention-is-all-you-need
39. Bai S, Kolter JZ, Koltun V (2018) An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. arXiv:1807.00458
40. Martinez B, Ma P, Petridis S, Pantic M (2020) Lipreading using temporal convolutional networks. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, pp 6319–6323. https://doi.org/10.1109/ICASSP40776.2020.9053841
41. Assael YM, Shillingford B, Whiteson S, de Freitas N (2016) Lipnet: sentence-level lipreading. arXiv:1611.01599
42. Bradski G (2000) The OpenCV Library. Dr. Dobb's J Softw Tools 25:120–125
43. Riba E, Fathollahi M, Chaney W, Rublee E, Bradski G (2018) Torchgeometry: when PyTorch meets geometry. https://drive.google.com/file/d/1xiao1Xj9WzjJ08YY_nYwsthE-wxfyfhG/view?usp=sharing
44. Riba E, Mishkin D, Ponsa D, Rublee E, Bradski G (2020) Kornia: an Open Source Differentiable Computer Vision Library for PyTorch. In: IEEE Winter Conference on Applications of Computer Vision (WACV), pp 3674–3683. https://doi.org/10.1109/WACV45572.2020.9093363
45. Chung JS, Zisserman A (2016) Lip reading in the wild. In: Asian Conference on Computer Vision (ACCV). Springer, pp 87–103. https://doi.org/10.1007/978-3-319-54184-6_6
46. Graese A, Rozsa A, Boult TE (2016) Assessing threat of adversarial examples on deep neural networks. In: IEEE International Conference on Machine Learning and Applications (ICMLA). IEEE, pp 69–74. https://doi.org/10.1109/ICMLA.2016.0020
47. Guo C, Rana M, Cisse M, van der Maaten L (2018) Countering adversarial images using input transformations. In: International Conference on Learning Representations (ICLR)
48. Gupta P, Rahtu E (2019) CIIDefence: Defeating adversarial attacks by fusing class-specific image inpainting and image denoising. In: IEEE International Conference on Computer Vision (ICCV), pp 6708–6717. https://openreview.net/forum?id=SyJ7ClWCb
49. Graves A, Schmidhuber J (2005) Framewise phoneme classification with bidirectional LSTM and other neural network architectures. Neural Netw 18(5-6):602–610. https://www.sciencedirect.com/science/article/pii/S0893608005001206
50. Graves A, Fernández S, Schmidhuber J (2005) Bidirectional lstm networks for improved phoneme classification and recognition. In: International Conference on Artificial Neural Networks (ICANN). Springer, pp 799–804. https://doi.org/10.1007/11550907_126
51. Hayes J, Danezis G (2018) Learning universal adversarial perturbations with generative models. In: IEEE security and privacy workshops, SP workshops. IEEE Computer Society, pp 43–49. https://doi.org/10.1109/SPW.2018.00015

Anup Kumar Gupta has received his Master of Science (Research) degree in Computer Science and Engineering from the Indian Institute of Technology Indore, India, in 2021 and is currently pursuing his doctoral degree from the same institute. His area of interest includes deep learning, computer vision and adversarial attacks.

Dr. Puneet Gupta is currently working as an Assistant Professor in the Department of Computer Science and Engineering (CSE), Indian Institute of Technology Indore. Prior to that, he was a post-doctoral researcher at Tampere University, Finland, where he worked to make deep learning (DL) architectures more secure and reliable. Before that, he was a member of the Machine Vision group in Embedded Methods and Robotics, TCS Research and Innovation. He received his doctoral degree from the Department of CSE, Indian Institute of Technology Kanpur, India, in 2016. His area of research includes bio-metrics, image processing, computer vision and machine learning. He has published several papers in reputed International Journals and International Conferences.

Dr. Esa Rahtu received his PhD degree from the University of Oulu in 2007. Currently, he is an Assistant Professor at Tampere University (TUNI) in Finland. Prior to joining TUNI, Rahtu was a senior researcher at the Center of Machine Vision research at the University of Oulu in Finland. In 2008, he was awarded a post-doctoral research fellow funding by the Academy of Finland. His main research interests are in computer vision and deep learning.