iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 8 **Issue:** IV **Month of publication:** April 2020

DOI: http://doi.org/10.22214/ijraset.2020.4025

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 8 Issue IV Apr 2020- Available at www.ijraset.com

Automated Lip Reading using Word Level Sentence Prediction based on LSTMs and Attention

Harshit Singhania[1], Radha Shankarmani[2], Kumail Virani[3], Aakash Singh[4]
[1, 2, 3, 4]*Information Technology, Sardar Patel Institute of Technology,* Mumbai, India

Abstract: ***This paper describes the findings of our work in automated lip reading. The automated lip reading problem is the task of predicting the sentence spoken by a speaker on the sole basis of a video of them speaking, in this work, the audio information is not used and the sentence is predicted only from the pixel information. There are multiple works on this topic which employ deep learning algorithms; our work proposes an alternate neural network architecture using BiLSTMs with the attention mechanism. Our work also attempts to jointly train a language model on the base BiLSTM with attention model to better capture the semantics and grammar of the predicted sentence. We found that adding the bidirectional LSTM layer and the jointly trained language model instead of a base LSTM improved accuracy significantly.***
Keywords: Lip reading; long short term memory; attention; language model

I. INTRODUCTION

Our paper focuses on automated lip reading on the basis of a muted video of the speaker's face as he speaks. It has been found that humans have a poor ability of reading another person's lips, an accuracy of 21+- 11% only [1]. Lip reading to correctly identify the spoken phrase is a very difficult task for the average human, at times it may be impossible, since for some words the lip movement is very similar (homophones, letters like 'p' and 'b'), in these cases we must consider the context (previously spoken words) Potential applications of automated lip reading include silent dictation in public places, speech recognition in noisy environments, and the captioning of silent films and videos[1].

Automated Lip reading consists of 2 phases, the pre-processing phase and the classification phase, in the pre-processing phase we are expected to convert the input video frames to a form suitable for further classification by a neural network; the classification phase involves the actual classification and prediction of the sentence spoken.

Automated lip reading can be applied in a lot of fields, it can be used in smart vehicles to receive commands from the user with music playing in the background (which renders speech to text useless), it can also be used to assist the hearing impaired in cases where speech to text fails.

II. BACKGROUND REVIEW

There have been multiple papers on the task of automated lip reading using neural networks. Some are for word level predictions, wherein the model takes as input a sequence of frames and outputs a probability distribution over the vocabulary. These are [2]–[4]the obvious drawback is such an approach is, to employ them in a more pragmatic sentence level prediction task is difficult. Other works focus on sentence level predictions, here given a set of frames the model is able to predict the sentence uttered in those frames, [1], [5], [6]work on this task, in both the works the model predicts for each frame input a probability distribution over the character vocabulary typically [a to z and 0 to 9], this is clearly a more practically applicable approach.

Reference [3] suggests a word level classification model for the lip reading problem. The model takes as input a sequence of frames and outputs a probability distribution over the vocabulary of the training dataset, from this the word most likely spoken in those frames can be extracted. To apply such a model, we must have a method to segment the speaker's video into frames representing each individual word spoken. If such a method can be found, a potentially powerful system can be developed.

Reference [4] combines CNNs(to process video frames) with RNNs(to process the time series features extracted by the CNN). The dataset used was created by the authors themselves and consisted of only them speaking the numbers from zero to nine. Thus, their model's output layer consisted of only 10 units with a softmax activation to normalize into probabilities. Finally, they were able to achieve an accuracy of around 80% on their test data.

Lipnet[1] is the most successful work we found that publishes results on the GRID dataset[7], it reports a 5% word error rate on overlapped speakers. Lipnet uses spatiotemporal convolutional neural networks (STCNNs) to preprocess the input video frames, the resultant feature vector is sent to Gated Recurrent Unit (GRU) layer, the output from this is passed through a few fully connected layers and finally classifies through a softmax layer. It is trained with the Connectionist Temporal Classification (CTC) loss.

Reference [6] uses both audio and video data from multiple datasets to train a neural network using both convolutional layers and 2 recurrent layers. The video frames are processed by the convolutional layer to generate output 1; the audio data is processed using MFCC features (Mel frequency cepstral coefficients) by RNN sequence 1. The output from both of these modules is used in the dual attention mechanism by the final RNN sequence, to generate character level predictions. It reports results on the GRID dataset, this work is able to achieve a word error rate of 3%, which substantially exceeds the current state of the art, but their model uses audio information too, this audio will understandably not be available in most of the aforementioned applications where a lip reading model would be deployed.

III. THE GRID CORPUS

The grid corpus described in [7] is used often in research on visual speech perception, we chose this dataset because it was easily available and provided word and frame alignments for each video.

A serious barrier to develop a model for speech recognition is the unavailability of material to be spoken. You have to be very careful with the speech material as it contains instability in voice balance and uncontrollable external conditions.

In this database for speech recognition there are 1000 sentences by thirty-four different speakers from England, Scotland and one also from Jamaica and 18 men and 16 women among them with ages ranging between late teens and before 50. The sentences were spoken in the form of a six-word sequence namely color, letter, digit, command, preposition and adverb. This was done so the system has ease of recognizing the words and categorizing them into different parts hence improving machine learning.

Audio-Visual recordings are done in sound isolated rooms so there were no disturbances while recording the audio. The speakers are told to speak freely and over carefully utter the words so as to recognize normal patterns of speech.

Those utterances that were misread were repeated to be corrected and pronounced properly again. There were approximately 57 cases out of a 1000 that were needed to be repeated.

Each sentence of the grid corpus is of the form <command:4, color:4, preposition:4, letter:25, digit:10, adverb:4>, the number of possibilities for each component are indicated. As a result the vocabulary consists of a total of 51 words/letters and thus our model must predict a probability distribution over 51 components.

A detailed description of the grid corpus and its merits over other similar corpuses can be found at [7].

Table 1: THE GRID CORPUS VOCABULARY

Word type	Words
Command	Bin, lay, place, set
Color	Blue, green ,red, white
Preposition	At, by, in, with
Letter	a-z, excluding w
Digit	0-9
Adverb	Again, now, please, soon

IV. EXPERIMENTAL METHODOLOGY

As mentioned previously, the process will require 2 stages, pre-processing and classification. The following section describes each of these stages.

A. Pre-Processing

1) *MTCNN:* Multi-task Cascaded Convolutional Networks[8] is used to closely crop video frames to only the face of the speaker. We used a pre trained model available as a part of the python module ipazc/mtcnn. These frames were then reshaped to 224 x 224. The mtcnn module provides bounding boxes for each of the detected speakers in the image; Each bounding box contains pixel coordinates for the nose, mouth right edge, right eye, left eye and the mouth's left edge along with the bounding box for the entire face.

2) *VGG Face:* We used the pre-trained "VGG-Face"2 CNN by the Visual Geometry Group from University of Oxford. VGG-Face was based on the University of Oxford's VGG-16 architecture, which contains a sequence of convolution, pooling and fully connected layers. VGG-Face was trained with 2,622 identities for a total of 2.5 million face images[9]. This model's architecture will be used to process the video frames in our project. The extracted feature vector can be processed further to get the final text output. This was used to vectorize each input closely cropped video frame to a 2048 dimensional feature vector which can then be processed by a neural network.

B. Classification

After the preprocessing phase has been completed, each input video of 75 frames is converted to a (75, 2048) dimensional matrix. This is the input to the classification phase, the target labels are formulated as follows; the target for frame *i* is the word that was being spoken during that frame. So for instance, if the speaker was speaking the word "place" from frames 5 to 10 then for each of those frames the target would be the word "place". Since the vocabulary of the grid corpus is only of 53 words (51 mentioned previously; 1 "sil" when the speaker is not saying anything, commonly found at the beginning and at the end of the video and 1 "sp" we found in the alignment files, although this we removed from the training and test data) we used one hot encoding for the output with a simple softmax activation at the output layer, for a more extensive vocabulary a hierarchical softmax might be beneficial. Thus, the label vector for each video will be a matrix of shape (75, 53).

We then tried a few different models, the results of which are detailed below, all of the results mentioned are from training on a random sample of 700 videos of the 1st speaker and testing on 200 random samples from the remaining videos, the final 100 videos were used for validation.

1) Base BiLSTM with attention model: As mentioned previously, the input to the classification phase is a matrix of dimensions (75, 2048). As part of this model we first passed this through a few fully connected layers to reduce the dimensionality to (75, 256). This was then passed through a BiLSTM layer with Luong style self-attention[10]. The output of the BiLSTM layer was a matrix of shape (75, 512), as part of the attention mechanism the vector at time step *i* will first compute a set of attention scores with all of the 75 time steps, these are computed as follows

$$score(i, j) = timestep(i) \bullet timestep(j)$$

$$res(i) = \sum_{j=1}^{75} score(i, j) * timestep(j)$$

Where $res(i)$ is the attention result for time step *i* and $\bullet$ Represents the vector dot product. Thus, the attention output for each time step is also a 512 dimensional vector; this is concatenated with the BiLSTM output for that time step to generate a final matrix of shape (75, 1024) for the attention + BiLSTM layers. This is then again passed through a few fully connected layers with dropout regularization to produce the final output layer with 53 units (= vocabulary) with a softmax activation.

It was found that this form of attention did not improve the word error rate significantly; it dropped from 0.157 without attention to 0.145 with attention. We can explain this by observing the attention score visualization given below generated for a random sample from the test set, it can be clearly seen that each time step attends most only to itself and assigns a near 0 score to all other time steps.

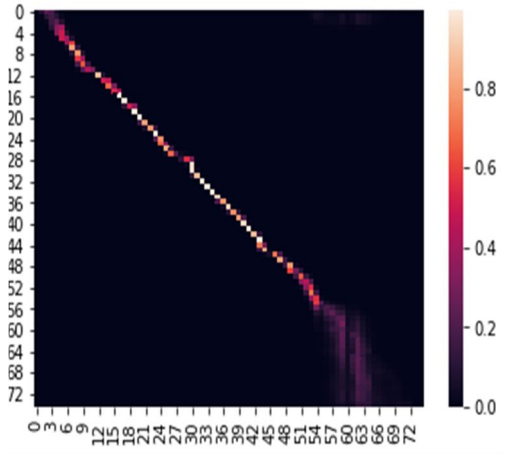


Figure 1: Attention score visualization

2) BiLSTM with attention and jointly trained language model: We concluded that adding a language model to this base BiLSTM model is likely to improve results; this was because we found after seeing our model's predictions on random samples from the test set that the model hadn't understood the semantics of the output sentence. The GRID corpus has a strict sentence structure and our model's predictions were often violating this structure. To integrate the language model we added another BiLSTM + attention layer to the outputs of the previously mentioned base BiLSTM model and jointly trained the two.

This was done by having our model output 2 matrices, one by the base BiLSTM model and the other after the 2nd BiLSTM layer was attached to the base model. Both of the outputs are of the same shape i.e. (75, 53) and the targets for both of them are also the same. The final loss which was to be minimized during back propagation is the weighted sum of these categorical cross entropy losses, the weights we found by a simple search to be [0.5, 2].

Fig 2 shows the attention score visualization of the 2nd layer for a random sample, as can be seen it is much more spread out.

Fig 3 shows the final model's architecture, the FC3 to OP2 indicates the skip connection; the outputs of FC3 were concatenated with those of FC5. Dropout regularization was used extensively but isn't shown due to a lack of space. The final loss is calculated on OP1 and OP2.

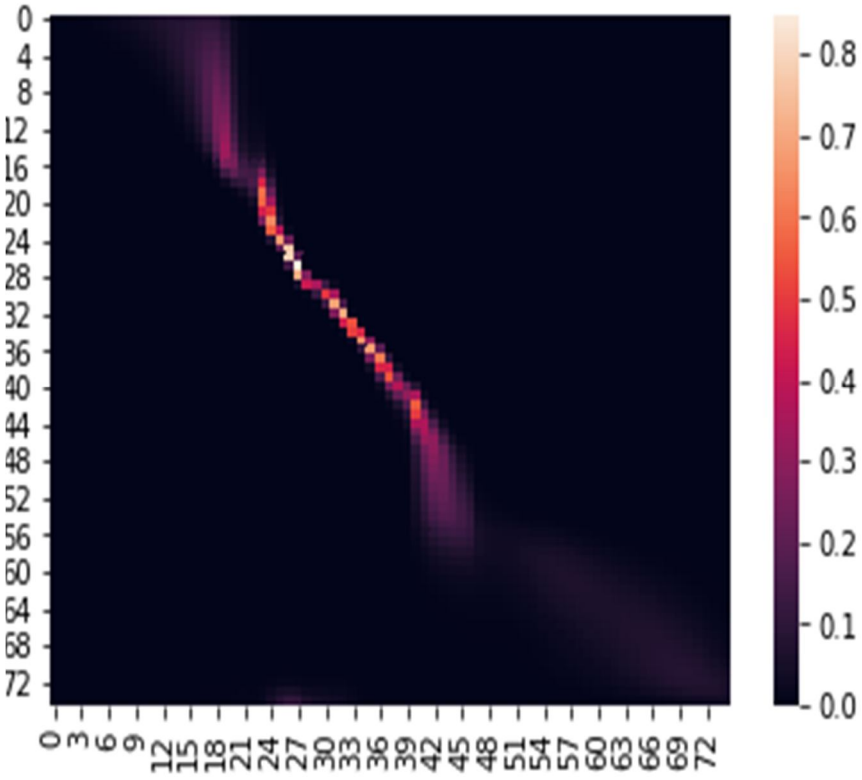


Figure 2: Attention score visual of the 2nd layer

C. Inference

As mentioned earlier, our model outputs a probability distribution over the vocabulary of words for each input frame. To get the final sentence prediction from this model extra post processing must be done. For instance, if our model predicts:

Table 2: A Prediction Example

Word predicted	bin	bin	bin	G	G	At	at
Frame Number	1	2	3	4	5	6	7

These predictions must be condensed so that the final prediction of *"bin g at"* can be generated, this was done using a simple for loop and eliminating repeated consecutive words, this has one obvious drawback that repeated words cannot be predicted directly, this can be resolved by inserting a special character between repeated words for a frame in the training dataset.

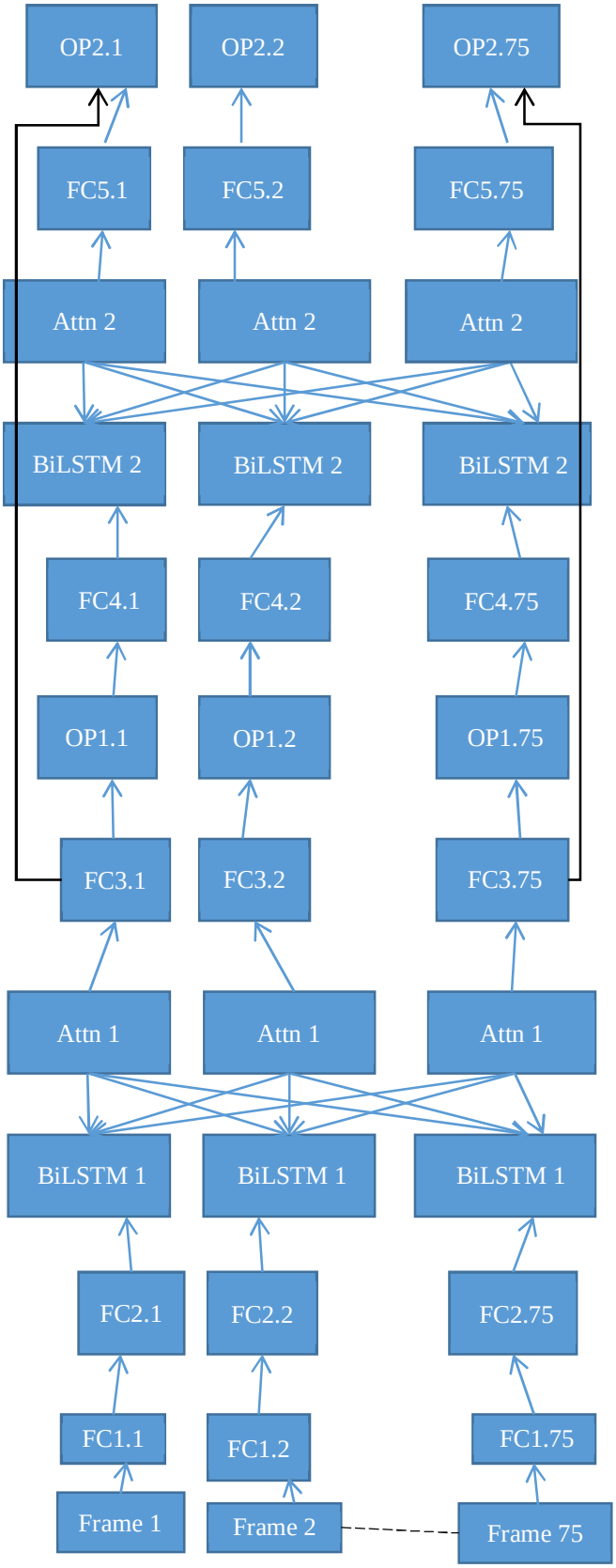


Figure 3: The model architecture. FC stands for Fully connected layer, Attn for the Luong Attention layer, OP for the output layer with *softmax* activation. All other layers have a *relu* activation.

V. RESULTS

The final word error rate results of our various models are shown below in table 3. These are all speaker dependent results; the models were trained on a 700 video random subset of the 1st speaker in the grid corpus.

The LSTM model performs the worst and this can be explained by taking an example, consider table 2 mentioned previously; the LSTM model is unable to handle the word transitions effectively, consider the transition from predicting the word *bin* at time step 3 to predicting *g* at time step 4, the LSTM model must predict *g* only on the basis of frames 0 to 4 even though the actual word might have been said in a later frame, changing to a BiLSTM model changes this entirely and now at every time step the context of every other time step is available to the model, thus, the accuracy improves multi fold. Finally, the BiLSTM with language model provides the best accuracy.

Table 3: RESULTS

MODEL	WORD ERROR RATE
Simple LSTM	0.75
BiLSTM without attention	0.157
BiLSTM with attention	0.145
BiLSTM with attention and jointly trained language model	0.125

VI. FUTURE WORK

The final word error rate of our model is 0.125 or 12.5 % on speaker dependent testing, Lipnet [1] on the other hand achieves a word error rate of 4.8%. We can improve our model in a number of ways

A. Since the model was trained only on the 1st speaker's videos, the accuracy could be improved by training on the remaining speakers as well.

B. Training the convolutional feature extractor end to end is likely to improve the accuracy. After closely cropping the lips with MTCNN as mentioned before, we can add a convolutional feature extractor and train it end to end with the language model and base BiLSTM model instead of using VGG Face; since, VGG Face was trained on entire faces and not just the lips of speakers; we must train our own CNN feature extractor. Using only lip data is likely to provide better results.

C. Fine tuning the VGG model's weights is likely to provide an improvement as the model will be able to learn better features for each frame.

D. Adding multiple levels of attention and BiLSTM layers instead of just 1 layer, this approach has been proven useful in multiple other works, the transformer architecture in [5] also uses 6 levels of these combination of layers.

REFERENCES

[1] Y. M. Assael, B. Shillingford, S. Whiteson, and N. de Freitas, "LipNet: End-to-End Sentence-level Lipreading," pp. 1–13, 2016, [Online]. Available: http://arxiv.org/abs/1611.01599.

[2] A. Au and A. Heins, "Automated Lip Reading using Delta Feature Preprocessing and LSTMs," 2017.

[3] M. Wand, J. Koutník, and J. Schmidhuber, "Lipreading with long short-term memory," *ICASSP, IEEE Int. Conf. Acoust. Speech Signal Process. - Proc.*, vol. 2016-May, pp. 6115–6119, 2016, doi: 10.1109/ICASSP.2016.7472852.

[4] Y. Lu and H. Li, "Automatic lip-reading system based on deep convolutional neural network and attention-based long short-term memory," *Appl. Sci.*, vol. 9, no. 8, 2019, doi: 10.3390/app9081599.

[5] T. Afouras, J. Son Chung, and A. Zisserman, "Deep lip reading: A comparison of models and an online application," *Proc. Annu. Conf. Int. Speech Commun. Assoc. INTERSPEECH*, vol. 2018-Septe, pp. 3514–3518, 2018, doi: 10.21437/Interspeech.2018-1943.

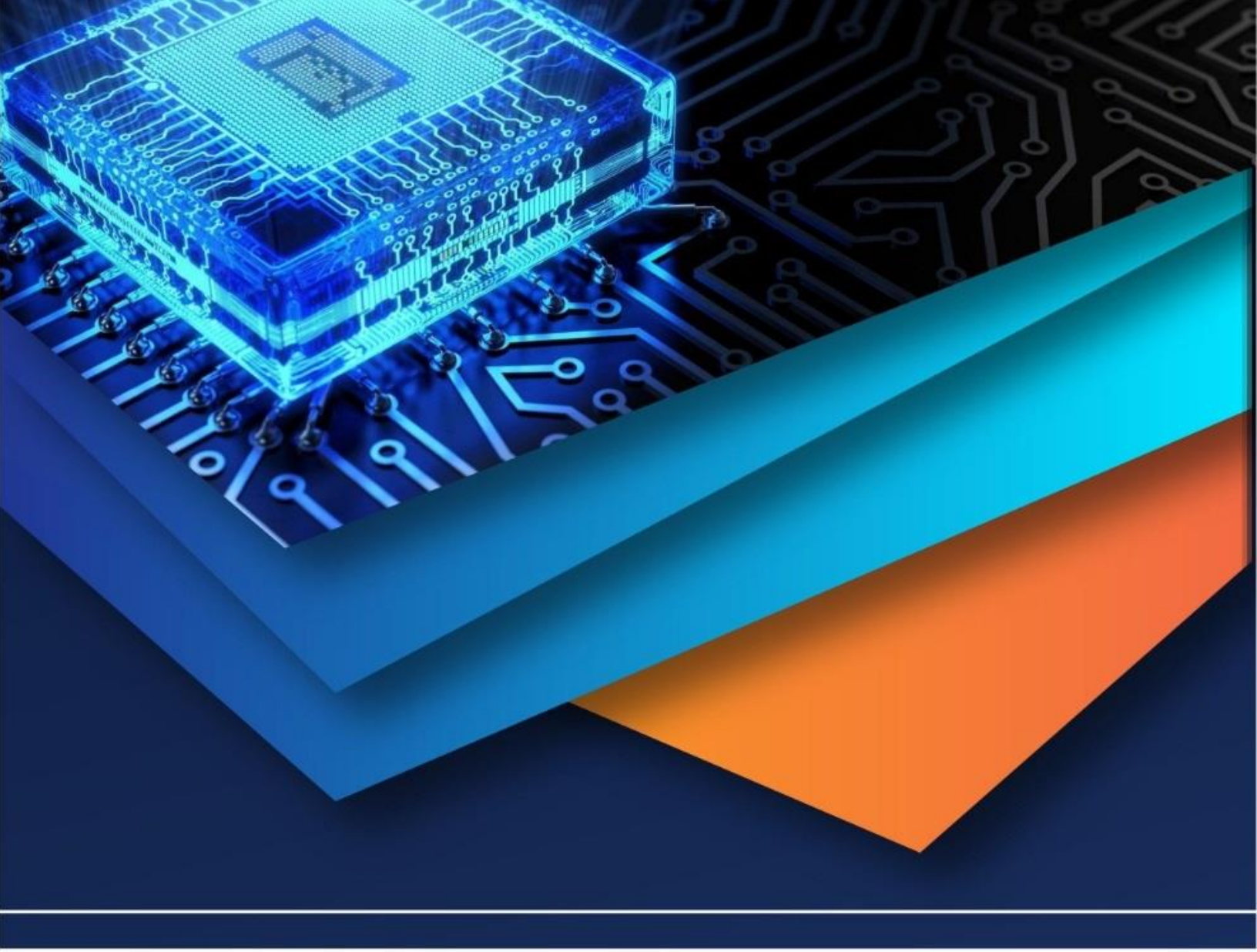
[6] J. S. Chung, A. Senior, O. Vinyals, and A. Zisserman, "Lip reading sentences in the wild," *Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017*, vol. 2017-Janua, pp. 3444–3450, 2017, doi: 10.1109/CVPR.2017.367.

[7] M. Cooke, J. Barker, S. Cunningham, and X. Shao, "An audio-visual corpus for speech perception and automatic speech recognition," *J. Acoust. Soc. Am.*, vol. 120, no. 5, pp. 2421–2424, 2006, doi: 10.1121/1.2229005.

[8] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, "Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks," *IEEE Signal Process. Lett.*, vol. 23, no. 10, pp. 1499–1503, 2016, doi: 10.1109/LSP.2016.2603342.

[9] O. M. Parkhi, A. Vedaldi, and A. Zisserman, "Deep Face Recognition," no. Section 3, pp. 41.1-41.12, 2015, doi: 10.5244/c.29.41.

[10] M. T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," *Conf. Proc. - EMNLP 2015 Conf. Empir. Methods Nat. Lang. Process.*, pp. 1412–1421, 2015, doi: 10.18653/v1/d15-1166.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY