IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 1 September 2022, accepted 19 September 2022, date of publication 22 September 2022,
date of current version 30 September 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3208921

RESEARCH ARTICLE

Deep Learning-Based Detection of Inappropriate Speech Content for Film Censorship

ABDULAZIZ SALEH BA WAZIR, HEZERUL ABDUL KARIM, (Senior Member, IEEE),
HOR SUI LYN, (Student Member, IEEE),
MOHAMMAD FAIZAL AHMAD FAUZI, (Senior Member, IEEE),
SARINA MANSOR, AND MOHD HARIS LYE, (Member, IEEE)
Faculty of Engineering, Multimedia University, Cyberjaya 63000, Malaysia

Corresponding author: Abdulaziz Saleh Ba Wazir (zezu9512@gmail.com)

This work was supported by Telekom Malaysia (TM) Research and Development, Malaysia, under Project MMUE/180029.

ABSTRACT Audible content has become an effective tool for shaping one's personality and character due to the ease of accessibility to a huge audible content that could be an independent audio files or an audio of online videos, movies, and television programs. There is a huge necessity to filter inappropriate audible content of the easily accessible videos and films that are likely to contain an inappropriate speech content. With this in view, all the broadcasting and online video/audio platform companies hire a lot of manpower to detect the foul voices prior to censorship. The process has a large cost in terms of manpower, time and financial resources. In addition to inaccurate detection of foul voices due to fatigue of manpower and weakness of human visual and hearing system in long time and monotonous tasks. As such, this paper proposes an intelligent deep learning-based system for film censorship through a fast and accurate detection and localization approach using advanced deep Convolutional Neural Networks (CNNs). The dataset of foul language containing isolated words samples and continuous speech were collected, annotated, processed, and analyzed for the development of automated detection of inappropriate speech content. The results indicated the feasibility of the suggested systems by reporting a high volume of inappropriate spoken terms detection. The proposed system outperformed state-of-the-art baseline algorithms on the novel foul language dataset evaluation metrics in terms of macro average AUC (93.85%), weighted average AUC (94.58%), and all other metrics such as F1-score. Additionally, proposed acoustic system outperformed ASR-based system for profanity detection based on the evaluation metrics including AUC, accuracy, precision, and F1-score. Additionally, proposed system was proven to be faster than human manual screening and detection of audible content for films' censorship.

INDEX TERMS Foul language, speech recognition, key word spotting, spoken term detection, censorship, deep learning, convolutional neural network.

I. INTRODUCTION

With the increased exposure to portable and immediate screen time sources such as televisions, computers and smartphones, filtering of audio and visual contents is becoming crucial. This is because media commonly include offensive and sensitive contents, e.g., foul languages, nudity, and sexually explicit contents, which could attract the attention of users in entertainment videos, games and movies available through broadcasting channels or at online platforms. Tuttle [1] stated that most movies incorporate the usage of profanity that could negatively affect the society [2] and that she believed that this frequency would increase over the years. Broadcasting companies and media-sharing platforms are responsible in ensuring the appropriateness of contents shared to the public through their respective channels. In the case of language, censorship is a complex filtering process that provides language content appropriate to consumers due to the restrictions

The associate editor coordinating the review of this manuscript and approving it for publication was Kathiravan Srinivasan.

in personnel, time, cost and human attention that may cause indetection of content that should be removed. The objective of this research was to design and implement a censoring system to accurately detect spoken profane language in audio signals of audio and video files. Specifically, neural networks that reported intriguing properties of such techniques was utilized in facilitating the audio censorship of videos.

In recent years, the application of deep learning techniques in speech recognition has gained popularity. Various utterance types, such as spontaneous and continuous speeches, and connected and isolated words, were targeted to be detected by different speech identifying systems [3]. One of the popular techniques was Spoken Term searching technique, which could be further divided into Spoken Term Detection (STD) and Keyword Spotting (KWS). Spoken Term Detection (STD), or spoken term discovery, is the identification of recurrent speech fragments from raw speech without prior knowledge of the language, i.e., automatic retrieval of speech from a database through specific audio keywords or queries [4], [5], [6].

The speech searching could be performed either by typing the keyword directly or speaking it to an Automatic Speech Recognizer (ASR) to convert the speech into text form. Each of the retrieved files from the search process would have a label or caption with the keyword included [7]. The obvious disadvantage of this method is the requirement for all audio contents to be labeled, which is a difficult task. Furthermore, it is difficult to detect occurrences of the target keyword by searching for similar speech signals produced by the same person. Speech signals for the same keyword spoken at different times are not identical [8]. This is due to the (subconsciously) differing pitch, energy content, speech length, emphasis and pauses which could be related to other factors such as age, voice condition and mood. Therefore, it is only reasonable for the detection task to be even more challenging when more than one individual contributed to the utterances of words in the speech database due to the differences in human vocal attributes, accent, dialect, gender, age and so on. This poses challenges in STD and speech recognition task as a whole [6], [7], [8].

Spoken keyword spotting (KWS) is a fast-growing technology due to the increased usage often coupled with deep learning techniques that involves the identification of keywords in audio streams [9]. As a consequence of the rapid growth of human-machine interaction via voice, the social usage of this technology is expected to achieve sustainable growth. For instance, usage of voice assistants requires activation through specific spoken keyword, i.e., wake-up word, which reduces the computational requirement and cost of the system significantly [10]. Although the far more computational expensive ASR is not required, KWS technique utilized in voice assistants could be interpreted as a sub-problem of ASR [11]. Besides the voice assistant activation, applications of KWS are common in audio indexing, speech data mining, phone call routing, etc. [12].

One of the earlier methods of application of KWS involved the usage of large-vocabulary continuous speech recognition (LVCSR) systems [13], [14]. Such systems were deployed to decode speech signal to allow keyword to be identified in the generated lattices (i.e., in the phonetic units' representations of different sequences, given the speech signal, were likely sufficient). This approach is superior in the sense that it allows flexibility to handle changing or non-predefined keywords [15], [16], [17] (although often with performance drop when keywords are out of vocabulary [18]).

The main weakness of LVCSR-based KWS systems lies in the computational complexity dimension. Specifically, these systems require high computational resources in order to generate complex lattices [16], [19], which introduces latency [20]. Therefore, this approach is not suitable for the application of real time speech recognition and monitoring. For the application of voice assistants and machine wake-up words, the high computational resource and memory requirements also place constraints on the usage of LVCSR systems [19], [21], [22].

As deep learning techniques mature over the years, usages of deep spoken KWS systems [23], [24], [25], [26] have increased due to progressively improving performance in terms of efficiency and accuracy, in voice assistants for instance. The sequence of word posterior probabilities generated by deep neural networks is processed to identify the possible existence of keywords directly without intervention of any Hidden Markov Model (HMM) or Gaussian Mixture Model (GMM). This deep KWS method has been attracting attention due to flexible complexity of DNN generating the posteriors, or acoustic model, which is dependent on computational resource availability [27], [28], [29].

Deep spoken keyword spotting system [30], [31], [32] typically contains three main blocks [9]: 1) the speech feature extractor that converts the input signal to a compact speech representation, 2) the deep learning-based acoustic model that generates posteriors over the keyword and filler (non-keyword) classes based on the speech features, and 3) the posterior handler that processes the temporal sequence of posteriors to determine the possible existence of keywords in the input signal.

Mel-scale-related features, low-precision features, learnable filter-bank features, and other features are the most relevant speech features used in deep KWS systems [9]. Speech features based on the perceptually-motivated Mel-scale filterbank, e.g., log-Mel spectral coefficients and Mel-frequency cepstral coefficients (MFCCs), have been commonly utilized in the areas of ASR and KWS. Despite the many attempts to learn optimal, alternative representations from speech signals, Mel-scale-related features is still a safe, solid, and competitive choice to date [33].

In most deep KWS systems, both types of speech features are normalized to have zero mean and unit standard deviation prior to being input to the acoustic model in order to stabilize and accelerate training and improve model generalization [34]. The most employed speech feature type in deep KWS thus far is Mel-scale-related features, as seen in the usage of MFCCs with temporal context are used in [34], [35], [36], [37], and [38]. Particularly, application of discrete cosine transform on the log-Mel spectrogram produces the corresponding MFCCs. This transform generates approximately decorrelated features suitable for acoustic models. Since deep learning networks are capable of exploiting spectro-temporal correlations, log-Mel spectrogram is used to yield MFCCs equivalent or better ASR and KWS performance [39]. Consequently, log-Mel or Mel filter-bank speech features with temporal context is widely utilized in deep KWS systems [40], [41], [42], [43].

The acoustic model is the core of deep spoken KWS systems, in which the objective of its design is to achieve increasing accurate models with minimal computational complexity. This system block could be filled using different deep learning models, e.g., fully-connected feed-forward networks [44], [45], convolutional networks [29], and recurrent and time-delay neural networks [46], [47], which could be used as standalone models, or as a combination of two models such as Convolutional Recurrent Neural Network CRNN [48]. Studies have shown that Convolutional Neural Networks (CNNs), with less parameters, could outperform fully-connected networks in the role of acoustic model in deep KWS [29], [49], [50], [51]. An attractive property of CNNs is that the number of multiplications of the model can be easily restrained to meet the computational limitations by adjusting several hyperparameters such as filter striding, and kernel and pooling sizes, without necessarily sacrificing much performance [25].

The decision of whether a specific keyword exists in an audio stream is made after processing the sequence of posteriors produced by the acoustic model. The main posterior handling modes are non-streaming (static) mode and streaming (dynamic) mode. Non-streaming mode is a standard multi-class categorization of independent input segments (either segmented automatically or manually) formed from a single or part of word each, i.e., isolated word classification. The input segments would need to be long enough in order to cover the duration of an entire word, e.g., speech command dataset [52]. In this mode, the input segment x is often assigned to the class with highest posterior probability. The non-streaming deep KWS systems generated sharply peaked posterior distributions in [53] and [54]. One possible explanation for this phenomenon is that the non-streaming systems handle isolated, well-defined class realizations, and not the inter-class transition information, as opposed to the streaming system case. Despite that, non-streaming performance and streaming performance are apparently highly correlated [53], [54], thus causing the non-streaming KWS approaches to be more effective and relevant for KWS posterior handling.

On the other hand, streaming mode relates to continuous processing, like in real-life, of an input audio stream in which keywords are not segmented or isolated. It is possible for a given segment to not contain (parts of) the target keyword. For this mode, the inherently noisy sequence of raw posteriors, typically smoothed over time, e.g., via moving average, on a class basis [55], [56] prior to processing. Next, smoothed word posteriors are commonly utilized to make the decision of whether a keyword is present, either through comparison with a sensitivity threshold [57] or by selecting the class with highest posterior within a time sliding window [58]. One disadvantage of streaming mode processing is that false detection may occur when the same keyword realization is detected more than once in the smoothed posterior sequence as consecutive input segments may cover parts of the same keyword realization. Post processing technique would need to be employed in order to avoid this problem [26].

The current trend involves usage of KWS for voice activation voice assistants [59] and Voice Control of Hearing Assistive Devices [54]. Hence, the literature on automated speech recognition models using deep learning techniques mostly revolved around inoffensive language identification only. For instance, conversational and read speech dataset clear of profane language utterances such as LibriSpeech [60], Google's voice search traffic dataset [61], Google commands dataset [52], spoken digits dataset [62], and speech emotions dataset of conversational speech dialogues [63], [64] have been explored in recent years.

In 2020, [65] researched on the efficiency of foul language detection using pre-trained CNNs (e.g., Alexnet and Resnet50). The proposed solutions had inaccurate detection and high computational cost due to large number of network parameters, causing the system to fail to meet the requirements for real time usages, i.e., real time monitoring for profanity filtering in videos. Another work studied the categorization of isolated foul words versus isolated normal speech using a novel foul language dataset. Despite the acceptable performance on the tested dataset, the detection and localization performances within audio samples of the proposed methods (CNN and RNN) on other dataset consisting of conversational speech of continuous audios were not explored [66], [67]. In brief, the feasibility of spoken profanity detection and localization within audio files has not been proven for real time audio filtering applications.

This experiment was carried out on English profanities and its derivatives. The model utilizes the acoustic features of profanities for the purpose of detecting profane words and localize it within a continuous audio sample, unlike Automatic speech recognition (ASR) models that transcript any spoken words based on the language model that are used as a part of the whole ASR system. However, the use of ASR systems requires huge computational cost for the use of a large dataset. Furthermore, ASR systems consist of several sequenced stages including acoustic models and language models. In the scenario of detecting and localizing inappropriate speech content within a continuous audio input, requires an additional text detection model. Consequently, ASR-based systems for the detection of profanities suffers of latency. Additionally, the use of sequenced models could probably lead to a decrees of system performance, as a performance drop in one stage leads to performance drop in the following stage.

This paper makes several contributions as follows:

- A real-time censorship system for inappropriate speech content including profanities that aims to detect the audible profane content in a stream of audio whether as standalone audio file or incorporated within videos and films.
- The real-time detection, localization, and censorship speeds up the monitoring, scanning, and filtering processes of audible content moderation and reduces the physical effort of manual screening and censorship.
- The system uses a lightweight CNN with small filters and tiny architecture model that can be used using CPU and can still reduce the censorship process time. Furthermore, the proposed model can be used for future researches on the field of KWS and STD with the advantage of short inference time.
- A novel spoken profanities dataset that will be available on request by other researchers for future works in the field of inappropriate speech content. Additionally, the dataset and developed system with CNN model can also be utilized for future speech-based film rating researches.
- System evaluation is carried on with real-time videos and films dataset containing continuous audio samples that will be available with ready annotations for future works.
- A comparative analysis of proposed acoustic-based detection systems and ASR-based detection system is performed to highlights the advantages and disadvantages of both systems in terms of model's performance metrics and speed for speech term detection and localization within a continuous audio input.

In this work, KWS approaches using a novel, lightweight and distinct end-to-end neural networks (E2E CNN) was proposed for foul language identification and localization. Acoustic Log-Mel spectrograms were applied on a deep learning architecture, named CNN, to recognize and localize spoken profane language samples within continuous audio samples extracted from real videos. The reciprocal detection and localization task included foul words class and normal conversational speech class from a continuous audio input. Hence, this work is not an ASR system, where speech content if transcribed into the corresponding words. Additionally, the proposed work is not a simple audio recognition where a single spoken term from the same pool of dataset is fed into a model and classified into the corresponding label. The organization of this paper is as follows: Section II describes the study materials and methods, Section III presents the experimental settings, and evaluation metrics, Section IV details the experimental results, and Section V concludes the study and its possible implications.

II. MATERIALS AND METHODS

The datasets utilized in this study of English profanities are described in this section. Next, the methodology is explained in detail. Firstly, feature extraction process in Log-Mel spectrogram methods applied on raw audio samples is performed. Secondly, E2E CNN is used for feature learning. Thirdly, posterior handling methods are done for further processing. A short review of each method and its function are summarized in the following subsections.

A. DATASETS OVERVIEW

Three datasets were used in different experiments for various purposes. Summaries of the datasets, experiments and usages of the datasets are listed below:

1. MMUTM dataset including 4541 isolated spoken inappropriate words and 12100 isolated normal conversational speech samples. The proposed E2E CNN, and baseline models were trained and validated using the augmented data samples this dataset.
2. A subset of The Abused Project Audio Dataset (TAPAD) including 4511 isolated spoken inappropriate words. The proposed E2E CNN, and baseline models were trained and validated using the augmented samples of this dataset.
3. A new continuous speech foul language dataset that consists of 6 continuous audio samples. The E2E CNN, and baseline models were compared utilizing this dataset for the purpose of testing models on continuous speech samples.

1) MMUTM DATASET

This study uses a novel dataset (the MMUTM foul language dataset) obtained and analyzed at Multimedia University, Malaysia for a film censorship research project in collaboration with Telekom Malaysia (TM) [66]. The dataset is a selection of profane language collected through recordings and natural data samples from random videos to increase the sample variations that contributed to the dataset complexities. The first version of this dataset that is published in [66], contains nine classes of profanity (e.g. F-word) and total of 3105 isolated foul language samples. Regardless, the derivation of the aforementioned classes posed study complications regarding offensive language identification. The first version also includes a normal class representing casual speech and distinguishing profane words from normal counterparts during censorship. The normal class consist of 5100 original and 45900 augmented samples using various augmentation techniques that were detailed in [66]. However, for this work only 12100 samples of normal class were used to mitigate the issue of imbalance dataset and foul language data samples scarcity as data imbalance and rarity is a major issue for KWS systems [9]. Additionally, the effect of data augmentation has led to improving model's performance and robustness to noise [66].

For this work, MMUTM dataset have been updated with new data samples and different foul words/classes that were collected and prepared based on the approaches used in the first version that are fully described in [66]. The samples were collected through recordings in controlled and uncontrolled environments, in addition to samples extraction from existing videos. A total of 1436 isolated foul samples have been added to this novel and unique dataset. The new samples cover 10 new/different inappropriate spoken terms that contribute to 10 new classes for the dataset. For example, the word 'slut' is a new category that is added by the addition of this new samples, as this word did not exist in the original dataset. Therefore, the updated MMUTM dataset now consist of 4541 samples covering 19 different classes of foul words. MMUTM dataset were used to only train the models using two main labels (Foul vs Normal), as this study proposes to test the proposed model with continuous audio samples instead of only isolated samples. The samples' audio properties were set at 16-bits PCM, whereas the 1-channel samples were set at 16-kHz. MMUTM dataset was augmented to increase the number of samples eight times from 4541 foul and 12100 normal sample to 36328 foul and 96800 normal samples to enhance the models' robustness to noise, avoid models' over-fitting, performance improvement, and improve models' generalization as justified in [66].

The augmented dataset was then used to train proposed and baseline models. The augmentation was performed using two different augmentation techniques, which are noise incorporation (four real background noise and white noise), and pitch manipulation with two different setting to manipulate samples' pitch. Each augmentation method was used only once on each training sample. Therefore, each sample will have 8 different variation including original sample, one white noise inappropriate sample, four different background noise incorporated samples, and two pitch manipulated samples with two different pitch settings that differs from the original sample.

2) TAPAD DATASET

TAPAD dataset [68] is an open dataset, it is still a growing database and open for contribution. The dataset collection and preparation procedures are described in [68]. Dataset consists of 26365 audio files covering 75 profane words classes. Most of these audio classes have 347 MP3 files of ∼5.783 minutes each. To best of our knowledge, this dataset has not been used previously in any speech recognition/detection researches before. For this work, only a subset of TAPAD was used to train the developed models under the foul class/label. The used subset consists of 4511 samples, covering 13 profanities that are totally different from the ones used from MMUTM dataset. Although, the samples' audio properties are 32-bits, 1-channel, and at sampling frequency of 24-kHz, the samples' audio properties were set to 16-bits PCM, 1-channel, and sampled at 16-kHz to match as an input to the system. This dataset was used to only train the models using under 'foul' label/class.

TABLE 1. Testing dataset summary.

ID	Title	Sample Source	Length (hh:mm:ss)	# Foul words
1	F-Word Movie Clips	YouTube	00:08:44	1062
2	Top 10 Movies with Excessive Swearing	YouTube	00:10:39	94
3	A F_cking Video About Swearing in Movies	YouTube	00:06:11	50
4	Best Of: Tom Segura | Netflix Is A Joke	YouTube	00:05:45	12
5	Beavis and Butt-Head Do America	Film	01:21:00	414
6	Uncut Gems	Film	02:15:00	646

TAPAD dataset was augmented to increase the number of samples eight times from 4511 foul sample to 36088 foul samples to enhance the models' robustness to noise, avoid models' over-fitting, and improve models' generalization and reduce. The augmented dataset was then used to train proposed and baseline models. The augmentation was performed using the same approaches used for MMUTM dataset that are described in the previous part.

3) TESTING CONTINUOUS AUDIO DATASET

This dataset is a novel challenging database that are only used for testing and model's evaluation purposes. This data consists of six real-world audios that were retrieved from videos available on the internet, four of the samples were retrieved from YouTube videos, while the other two are a full films. Full films are used in the evaluation as this research designed to propose a solution for films to provide real time monitoring and censorship for the inappropriate speech content. As described in Table 1. The total length of the testing videos is about four hours, seven minutes, and nineteen seconds, which is ∼ 247.32 minutes in total. It is obvious that the testing dataset intensively consist of foul languages within the normal conversation speech, as the dataset consist of 1322 profanity, where all the profanities are also existed in the training dataset of MMUTM and TAPAD dataset.

The rate of foul words per minute is what makes this dataset to be challenging, as there is about 5.345 offensive words per minutes in this dataset. Additionally, this dataset is a real dataset that is taken directly to test and evaluate the trained model, which adds to how challenging this dataset. The only per-processing happened to this dataset is the properties of the audio file that were set at sampling rate of 16-kHz, 1-channel, and 19-bits PCM. This dataset was purposely created for this research. Therefore, we have labeled all this dataset by manually finding the foul words within the audio file and the corresponding timestamps, in which the profane word occurs. Therefore, the annotations of this dataset consist of the foul words and its timestamps, as this work is to predict the foul word and localize it within a long audio file. Hence, the parts of the audio samples that were not labeled as foul, are considered as normal conversational speech by default,

TABLE 2. Summary of datasets used for models training and testing purposes.

Dataset	Usage	#Samples (original)	#Samples (augmented)
MMUTM	Training and validation	4541 foul and 12100 normal	36328 foul and 96800 normal
TAPAD	Training and validation	4511 foul	36088 foul
Testing continuous audio dataset	Model's testing	6 continuous audios from videos	Not augmented

whether it contains speech, noise, silence, or even music. Table 1 details the testing dataset.

In short, the training datasets MMUTM and TAPAD complement each other, contributing to 9052 profane word samples of 32 distinct vulgar words, and 12100 normal speech samples. Both datasets were only used for training the trained and baseline models while the testing was performed using a third dataset consisting of continuous audio samples retrieved from real videos. Table 2 summarizes the usages and number of samples of the three datasets.

B. LOG-MEL SPECTRAL FEATURES

Generally, the spectral content represented through Log-Mel spectrograms characterizes the target speech. To ensure frame overlapping, vectors were applied by sliding an analysis window over a portion of the frame size. Nevertheless, following the duration and properties of target speech, unpredictable differences occur in the coefficient vectors representing a given speech [69]. Visual inputs were analyzed in temporal dimension using CNN structures, whereby 2D Log-Mel spectrograms were obtained through coefficient vectors for CNNs. The complete spectral content features were then extracted using convolution process from the time and frequency domains.

A vector of features obtained from speech and acoustic signals could represent the temporal sequence features. Log-Mel spectrograms were used to extract serial vectors. In general, signal representations were formed after several steps using feature extraction approach. Firstly, pre-emphasis step filters and places emphasis on the higher frequencies to balance voiced sounds spectrum with steep roll-off in the high-frequency region. Next, windowing step involves the division of input signal into smaller frames with overlapping window to ensure that all serial sample features are extracted. Discrete Fourier Transform (DFT) is then applied on the windowed parts, which the log of the magnitude is taken and warped into the frequency domain on a Mel scale, generating the Log-Mel frequency sequence features.

Feature vectors converted from the corresponding 16-bits PCM, 1-channel audio samples obtained at 16-kHz comprised the dataset. The visual representations in the form of frequency spectrum of energy levels of speech were defined using 101 Log-Mel frequency spectrogram coefficients. Inappropriate and safe speech spectrogram analysis was performed using the following parameters: 0.03 frame duration, 1 second segment duration, 0.015 overlap window between frames, and 40 frequency bands. Furthermore, a lightweight model with small-sized filters was proposed in order to minimize the computational resource requirement and allow the target application of real time film audio filtering to be achieved. Therefore, the generated Log-Mel spectrogram image dimensions had small size, 40-by-101 in size specifically, where 40 is the normalized frequency of times 400-kHz (40 times 400 kHz = 16-kHz) and 101 is the number of spectrogram samples used. An example of raw signals of two profane words and their corresponding spectrograms are shown in Figure 1.

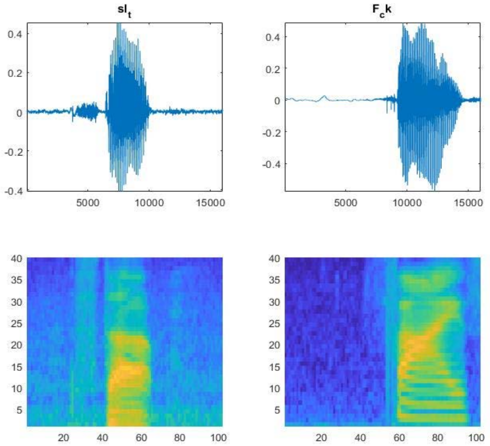

FIGURE 1. Two foul words' raw signal and the corresponding spectrograms.

C. E2E CNN

In the case of supervised CNN model, E2E learning mode is done to fine-tune parameters of the whole CNN. Since spectrogram images and labels were available during training process, supervised learning was applied. The CNN is composed of convolutional, fully connected, pooling and batch normalization layers. For detection of distinct signals, filters in horizontal and vertical lines present in CNNs were passed over input images. Mapping of image feature portions of the signals were then performed, and the classifiers were trained on the target task. Extraction of features of input images and pixel relationships were sustained by obtaining image features via small squares of input data using the convolution layers. A mathematical operation that involves two inputs, i.e., image matrix and a filter/kernel, was applied for the extraction.

Reduction of parameters of a specific image was allowed by the pooling layers. A common instance would be spatial pooling, i.e., downsampling or sub-sampling, which retained vital information while reducing dimensionality of each map. This pooling type could be categorized into (i) max pooling,

TABLE 3. Lightweight CNN architecture details.

Operation layer	# of filters	Feature map	Stride & Padding	Output
Image input layer				40 × 101 ×1
1st convolution layer	14	3 × 3 ×1	[1 1]	40 × 101 ×14
ReLU				40 × 101 ×14
Max pooling	14	3 × 3	[2 2]	20 × 51 ×14
2nd convolution layer	28	3 × 3 ×14	[1 1]	20 × 51 ×28
ReLU				20 × 51 ×28
Max pooling	28	3 × 3	[2 2]	10 × 26 ×28
3rd convolution layer	56	3 × 3 ×28	[1 1]	10 × 26 ×56
ReLU				10 × 26 ×56
Max Pooling	56	3 × 3	[2 2]	5 × 13 ×56
4th convolution layer	56	3 × 3 ×56	[1 1]	5 × 13 ×56
ReLU				5 × 13 ×56
Max pooling	56	1 × 13	[1 1] & 0 padding	5 × 1 ×56
Dropout layer				5 × 1 ×56
Dense layer	2 nodes	-	-	2
Softmax layer				
Classification layer				

which selects the largest element from the corrected feature map, (ii) average pooling, which takes the average value of the feature map elements, and (iii) sum pooling, which sums up all feature map elements. A flattened matrix vector under the convolution and pooling processes forms the fully-connected layer. This layer acted like a neutral network that integrated the convolution process features to build a model. In order to classify outputs related to the target task, an activation method involving SoftMax or sigmoid can be applied. Conversion of a vector of N values into a vector of N values that sums up to 1 was done by the SoftMax function. This function converts any input with positive, zero and negative values into values between 0 and 1 to allow the converted values to be interpreted as prediction probabilities.

A lightweight CNN designed and evaluated for vulgar speech content detection was experimented in this work. This CNN model was trained using E2E scheme for feature learning and classification involved categorization of inputs into one of the two classes: normal and foul. Four convolutional layers, four ReLU layers and three max pooling layers were used to build the proposed CNN model. The top (last) layers, i.e., fully-connected and SoftMax layers, allowed Log-Mel spectrogram images to be mapped for the classification task. Table 3 shows the details of the proposed CNN model architecture.

D. POSTERIOR HANDLER

In this work, non-streaming (static) KWS mode, which involved the standard multi-class categorization of independent input segments comprising a single word each (i.e., isolated word categorization), was employed. However, input segments in the training data pool did not contain isolated words only, instead, this work proposes to utilize lengthy continuous audio samples as test data samples. The audio files were passed through an automated windowing process to segment them into shorter samples of specific length. The windowed sub-sample was then input to the CNN model for class predictions performed based on the posterior probability, e.g., the class with highest posterior probability or positive detection if decision threshold was exceeded. The predicted class of the sub-sample is then assigned to the corresponding timestamps generated during windowing phase. Localization of recognized keyword within a long input audio sample could be related to the timestamps of which the sample consisted of identified profane word. Although continuous speech or audio sample was used as input, windowing process caused the inference for windowed samples to be considered as static mode. This mode is used due to its simplicity and produce a low number of false positives compared to dynamic mode. Hence, dynamic mode requires additional post-processing approaches to avoid such issue of increased false positive rate [9].

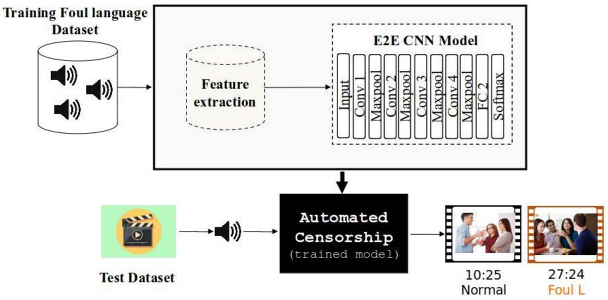

FIGURE 2. Proposed system architecture for inappropriate language detection.

III. EXPERIMENTAL SETUP

The experimental setup, performance metrics and testing results of the proposed system are discussed in this section. The experimental settings and procedures utilized for application of automated detection of profane speech content in film censorship are included in this section. The architecture of the proposed foul language detector system is illustrated in Figure 2. Feature extraction was performed on isolated samples of English language to obtain the Log-Mel spectral features, which were then sent into the CNNs for model training. Similarly, the test features were obtained from audio samples of real long audio files. These test features were used to evaluate the performance of the trained models.

The expected outputs of the system were the prediction probabilities of recognized profanity and the corresponding timestamps to allow localization of the foul word detection within test samples for film filtering. Hence, this work is not an Automatic Speech Recognition (ASR), where speech content if transcribed into the corresponding words. Additionally, the proposed work is not a simple audio recognition where a single spoken term from the same pool of dataset is fed into a model and classified into the corresponding label, as the test samples used is a continuous audio input of real-world samples that are out of the training dataset pool.

The evaluation pipeline begins consisting of serial steps with an input of real-time video sample, that is converted into an audio sample. Next, the audio sample is automatically segmented into smaller samples of fixed window length with an overlap time. After that, an automated censorship block receives the serial segments to perform an automated features computation of spectrograms and model inference. Hence, the spectrograms of the audio segments are calculated every 0.5 seconds (for example) of a continuous audio input stream to determine whether it belongs to inappropriate speech content or normal speech content. The outcome of the automated censorship block of probabilities, predictions, and utterance keywords is used to define the segment in which profane word was uttered. Furthermore, the process including windowing, spectrogram computation, and inference is carried out serially for every segment in continuous and automated manner to cover all the segments of an input sample. This process is to mimic the real-life scenario of films screening and censorship, which is presented as the test pipeline of this experimental work. Therefore, the system detects and localize a profane word in continuous audio input. Experiment implementation was carried out using Intel Core i7-8700 CPU @ 3.20 GHz, 64 GB RAM, and an NVIDIA GeForce GTX 1080 Ti GPU.

A. TRAINING ALGORITHM SETTINGS

All samples used for training and testing obtained from the datasets had similar properties of 1-channel, bit rate of 16-bits PCM, sampled at 16-kHz. CNN models were trained using E2E framework on isolated inappropriate words in MMUTM and TAPAD datasets, while testing was performed using the novel continuous audio profanity dataset composed of real videos and films. Momentum technique (Adaptive Moment Estimation) was utilized for the model training, with cross-entropy loss function applied. Regarding testing phase, segmentation was performed on the lengthy test data files using window lengths of 0.3, 0.4 and 0.5 seconds to determine the optimum segment length that could generate optimized performance metrics. After segmentation process, the fixed length, short samples were input to the trained model serially for identification of targeted profane words included in training process.

B. EVALUATION METRICS

Confusion matrix components were utilized in the calculations of evaluation metrics related to accurate detection of profane word samples. In this work, the model performance for detection of offensive spoken language was assessed using Accuracy, Precision, F1-score, True Positive Rate (TPR), False Positive Rate (FPR), and FNR as follows:

$$Accuracy = \left(\left(N_{tp} + N_{tn}\right) / N_{total}\right) \times 100 \tag{1}$$

$$TPR = R = \left(N_{tp} / \left(N_{tp} + N_{fn}\right)\right) \times 100 \tag{2}$$

$$FPR = \left(N_{fp} / \left(N_{fp} + N_{tn}\right)\right) \times 100 \tag{3}$$

$$FNR = \left(N_{fn} / \left(N_{fn} + N_{tn}\right)\right) \times 100 \tag{4}$$

$$F1 - score = \left(\frac{2\,(P \times R)}{(P + R)}\right) \times 100 \tag{5}$$

F1-score computed under precision (P) and recall (R):

$$P = \left(N_{tp} / \left(N_{tp} + N_{fp}\right)\right) \times 100 \tag{6}$$

In the equations, N_{tp}, N_{fp}, N_{fn}, and N_{total} referred to the number of true positives, false positives, false negatives, and total samples in all the segments respectively. Furthermore, the performance was evaluated using area under curve (AUC) and detection error trade-off (DET) curve. AUC was computed after plotting the receiver operating characteristic (ROC) curve which used FPR as the horizontal axis and TPR as the vertical axis. This measurement reflects the robustness of a binary classifier as the sensitivity threshold is varied. On the contrary, DET is a graphical plot of error rates for binary classification systems, i.e., graph of false rejection rate (FNR) against false alarms rate (FPR).

IV. EXPERIMENTAL RESULTS AND DISCUSSION

The audio-based foul word recognition model proposed in this research was designed to be applied for automated censorship of audio channels of films. The experimental results were obtained by running the novel test dataset, comprising of continuous video files with high inappropriate word rates per minute, through the trained models. Performance of the model was determined using performance metrics such as accuracy, F1 score, TPR, FPR and AUC. The results are discussing the model's performance based on segment lengths, probability thresholds, and process time figures.

A. SEGMENT LENGTH ANALYSIS

The experiment includes a windowing and segmentation process for the lengthy continuous test samples, before it goes to feature extraction, then inference and detection stages. Therefore, the segment length affects the detection and evaluation metrics. Hence, all the test samples were evaluated using three different segment lengths of 0.3, 0.4, and 0.5 seconds to find the optimized segment length, that produce the best and optimal system/model metrics for the detection of foul languages. Although all the test samples were tested based on different segment lengths, this paper will only demonstrate the effect of segment length on foul language detection within continuous audio samples, by highlighting the performance metrics of two samples that are sample 1 and sample 2 at a single probability threshold (th = 0.50) and three different segment lengths. Table 4 and Table 5 present the foul language detection model performance using two samples (sample 1 and 2), while Figure 3 and Figure 4 highlights the two samples performance based on average accuracy and F1-score, respectively.

Following Table 4 and Table 5, proposed model performed positively in the detection of foul language with high average accuracy, TPR, precision and F1-score, with low FNR and FPR. For example, samples 1 achieved 20.75%, 11.32%, and 3.83% FNR, for segment length of 0.3, 0.4, and 0.5 segments length respectively. Regardless, the model performance was

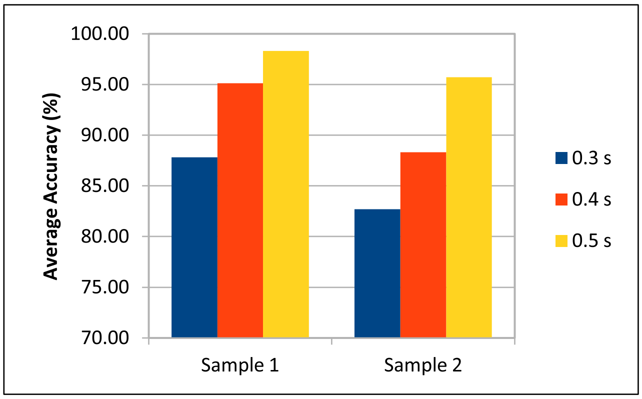

FIGURE 3. Average accuracy of samples 1 and 2 for different segment length.

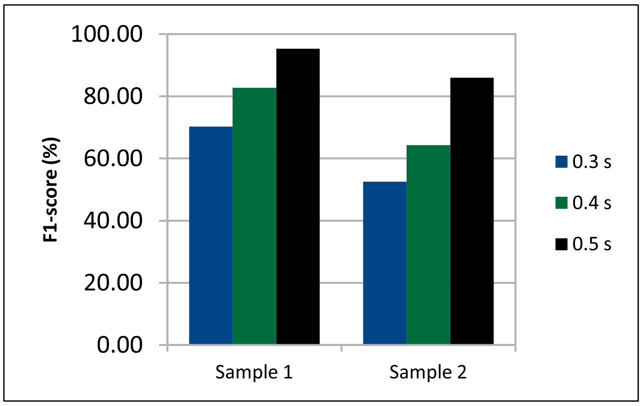

FIGURE 4. F1-score of samples 1 and 2 for different segment length.

TABLE 4. Performance metrics of sample 1 at 0.5 confidence score and different segment length.

Window length (s)	Avg. accuracy (%)	Recall (%)	FNR (%)	Precision (%)	FPR (%)	F1-score (%)
0.3	87.83	79.25	20.75	63.16	36.84	70.29
0.4	95.13	88.68	11.32	77.59	22.41	82.76
0.5	98.31	96.17	3.83	94.50	5.50	95.33

TABLE 5. Performance metrics of sample 2 at 0.5 confidence score and different segment length.

Window length (s)	Avg. accuracy (%)	Recall (%)	FNR (%)	Precision (%)	FPR (%)	F1-score (%)
0.3	82.69	67.02	32.98	43.15	56.85	52.50
0.4	88.32	81.91	18.09	52.83	47.17	64.23
0.5	95.71	91.49	8.51	81.00	19.00	85.93

improved for the longer segment length. For example, Sample 1 FNR was improved with about 16.92% when segment length used was 0.5 second, instead of 0.3 seconds. Likewise, the FNR of sample 2 were improved by around 24.47% when evaluated using 0.5 seconds compared to 0.3 second segment length. Based on Table 4 and Table 5, all the performance metrics were improved using larger segmentation length. For example, TPR/recall and precision were improved and increased drastically with longer window length, while FNR and FPR were improved and drops hugely at 0.5 second segment length.

TABLE 6. Overlap effect on performance metrics of sample 1 at 0.5 confidence score and 0.5 segment length.

Overlap time	Accuracy (%)	Recall (%)	FNR (%)	Precision (%)	FPR (%)	F1-score (%)
With overlap	98.31	96.17	3.83	94.50	5.50	95.33
Without overlap	96.01	90.36	9.64	81.35	18.65	85.62

TABLE 7. Overlap effect on performance metrics of sample 2 at 0.5 confidence score and 0.5 segment length.

Overlap time	Accuracy (%)	Recall (%)	FNR (%)	Precision (%)	FPR (%)	F1-score (%)
With overlap	95.71	91.49	8.51	81.00	19.00	85.93
Without overlap	90.63	84.09	15.91	57.29	42.71	68.15

Figure 3 and Figure 4 highlights model performance using sample 1 and sample 2 based on average accuracy and F1-score, respectively. F1-score measures the performance based on the precision and recall and produce a better view of the performance for an imbalanced dataset as in this work.

F1-score and average accuracy charts and figures show that increasing the segment length contributes into increment of model performance metrics, which are increasing accuracy, recall, precision, and F1-score. Consequently, the proposed system achieved the best performance on profane language detection using 0.5 segment length, where model achieved a high F1-score 95.33% and 85.93% for the sample and sample 2 test samples. Similarly, the model produced a high average accuracy of 98.31% and 95.71% for sample 1 and sample 2, successively. Therefore, 0.5 seconds considered as the optimal window length for the developed system. Hence, the proposed model was evaluated using 0.5 second segment length and the following detailed results were obtained based on the optimal window duration.

1) OVERLAP TIME ANALYSIS

The experiment includes an automated windowing and segmentation process for continuous test samples. Therefore, the fixed segment length affects the detection and evaluation metrics for words that are longer than window length, in addition to some keywords that might be spitted into two segments due to the automated and fixed windowing process. Hence, an overlap time was introduced to mitigate the error arises from this issue and find the optimal performance of profanities detection in a continuous sample with an automated and fixed windowing process. Although all the test samples were tested with and without overlap time, this paper only demonstrates the effect of overlap length for foul words detection within continuous audio, by detailing the performance metrics of two samples that are sample 1 and sample 2 at a single probability threshold (th = 0.50) and 0.5 segment lengths.

TABLE 8. Performance metrics of sample 1.

Threshold	Avg. accuracy (%)	Recall (%)	FNR (%)	Precision (%)	F1-score (%)	FPR (%)
0.10	98.50	97.17	2.83	88.03	92.38	11.97
0.25	98.50	97.17	2.83	92.79	94.93	7.21
0.50	98.50	97.17	2.83	92.79	94.93	7.21
0.60	98.31	96.17	3.83	94.50	95.33	5.50
0.70	97.94	96.17	3.83	95.37	95.77	4.63
0.80	97.94	96.17	3.83	95.37	95.77	4.63
0.90	96.82	96.17	3.83	95.37	95.77	4.63

TABLE 9. Performance metrics of sample 2.

Threshold	Avg. accuracy (%)	Recall (%)	FNR (%)	Precision (%)	F1-score (%)	FPR (%)
0.10	95.71	91.49	8.51	60.71	72.99	39.29
0.25	95.71	91.49	8.51	69.37	78.91	30.63
0.50	95.71	91.49	8.51	81.00	85.93	19.00
0.60	94.92	86.17	13.83	81.00	83.51	19.00
0.70	94.92	86.17	13.83	81.90	83.98	18.10
0.80	91.90	81.91	18.09	81.90	81.91	18.10
0.90	88.89	72.34	27.66	81.90	76.83	18.10

Table 6 and Table 7 present the performance of profanity detection performance using two samples (sample 1 and 2) and highlight the effect of introducing overlap time to windowing process. This overlap time is introduced to mitigate the error of keywords misdetection when uttered profane word is longer than the segment length or appears in two different segments due to the automatic windowing process of a continuous stream input. According to Table 6 and Table 7, proposed model performance was significantly improved in the detection of profanities when overlap time is introduced, which mitigates the error of foul words detection in a continuous audio sample. The overlap time produces higher performance metrics including average accuracy, recall, precision, and F1-score, with lower FNR and FPR. For example, the error of missing the target keywords for sample 1 achieved 9.64% and 3.83% FNR, for windowing process with and without overlap time, respectively.

Notably, the model performance was improved when overlap time was introduced. This is explained by FNR increase with around 5.81% when the windowing process of continuous input audio sample was executed with an overlap time. Likewise, the FNR of sample 2 were improved by around 7.4% when evaluated with overlap time. Based on Table 6 and Table 7, all the performance metrics were improved using windowing overlap time. Hence, the error arising from the issue of utterances with length larger than window length and utterance split was mitigated. Therefore, the overall foul words keywords detection in continuous audio was improved. For instance, accuracy, recall, precision, and F1-score were improved and increased significantly, while FNR and FPR were improved and dropped dramatically.

TABLE 10. Performance metrics of sample 3.

Threshold	Avg. accuracy (%)	Recall (%)	FNR (%)	Precision (%)	F1-score (%)	FPR (%)
0.10	98.11	96.00	4.00	88.89	92.31	11.11
0.25	97.84	94.00	6.00	90.38	92.16	9.62
0.50	97.57	92.00	8.00	93.88	92.93	6.12
0.60	97.30	90.00	10.00	93.75	91.84	6.25
0.70	97.04	90.00	10.00	95.65	92.74	4.35
0.80	96.77	88.00	12.00	95.74	91.71	4.26
0.90	96.50	88.00	12.00	97.78	92.63	2.22

TABLE 11. Performance metrics of sample 4.

Threshold	Avg. accuracy (%)	Recall (%)	FNR (%)	Precision (%)	F1-score (%)	FPR (%)
0.10	98.26	96.67	3.33	85.29	90.63	14.71
0.25	97.97	94.17	5.83	91.87	93.00	8.13
0.50	97.68	92.50	7.50	92.81	92.65	7.19
0.60	97.39	90.83	9.17	93.16	91.98	6.84
0.70	97.39	86.67	13.33	93.69	90.04	6.31
0.80	97.10	84.17	15.83	94.39	88.99	5.61
0.90	96.52	82.50	17.50	95.19	88.39	4.81

TABLE 12. Performance metrics of sample 5.

Threshold	Avg. accuracy (%)	Recall (%)	FNR (%)	Precision (%)	F1-score (%)	FPR (%)
0.10	98.50	96.62	3.38	88.11	92.17	11.89
0.25	98.35	95.41	4.59	90.80	93.05	9.20
0.50	98.29	94.93	5.07	92.47	93.68	7.53
0.60	98.11	94.69	5.31	92.89	93.78	7.11
0.70	97.92	94.20	5.80	93.98	94.09	6.02
0.80	97.53	94.20	5.80	95.12	94.66	4.88
0.90	96.71	94.20	5.80	95.59	94.89	4.41

TABLE 13. Performance metrics of sample 6.

Threshold	Avg. accuracy (%)	Recall (%)	FNR (%)	Precision (%)	F1-score (%)	FPR (%)
0.10	96.91	95.58	4.42	86.44	90.78	13.56
0.25	96.91	95.58	4.42	91.20	93.34	8.80
0.50	96.91	95.58	4.42	91.20	93.34	8.80
0.60	96.72	94.58	5.42	92.91	93.74	7.09
0.70	96.35	94.58	5.42	93.78	94.18	6.22
0.80	96.35	94.58	5.42	93.78	94.18	6.22
0.90	95.23	94.58	5.42	93.78	94.18	6.22

B. THRESHOLD-BASED MODEL PERFORMANCE

The performance assessment of the proposed models on the detection of foul language for the six test samples is presented in Table 8 through Table 13 for sample 1 through sample 6. Although model test was done using threshold zero through one, the tables present 0.1, 0.25, and 0.5 through 0.9 probability threshold. This is due to the common concern of threshold performance above the common 0.5 confidence score. However, all the thresholds starting from zero were used when evaluating the model using ROC and DET curves that are highlighted in the subsequent section. The results of all samples' performance are presented due to the concern of highlighting the model performance depending on different real-world samples, as different real time samples will exhibit different characteristics like audio quality, noise, pitch speed, etc. These characteristics produces different model's response in terms of target keyword detection. Therefore,

TABLE 14. Macro average performance of profanity censorship system using all test samples.

Threshold	Avg. accuracy (%)	Recall (%)	FNR (%)	Precision (%)	F1-score (%)	FPR (%)
0.10	97.67	95.59	4.41	82.91	88.54	17.09
0.25	97.55	94.64	5.36	87.74	90.90	12.26
0.50	97.45	93.94	6.06	90.69	92.24	9.31
0.60	97.13	92.07	7.93	91.37	91.69	8.63
0.70	96.93	91.30	8.70	92.40	91.80	7.60
0.80	96.27	89.84	10.16	92.72	91.20	7.28
0.90	95.11	87.97	12.03	93.27	90.45	6.73

TABLE 15. Weighted average performance of profanity censorship system using all test samples.

Threshold	Avg. accuracy (%)	Recall (%)	FNR (%)	Precision (%)	F1-score (%)	FPR (%)
0.10	97.47	95.68	4.32	85.12	89.96	14.88
0.25	97.40	95.13	4.87	89.34	92.07	10.66
0.50	97.36	94.83	5.17	91.05	92.88	8.95
0.60	97.12	93.64	6.36	92.07	92.84	7.93
0.70	96.85	93.45	6.55	93.08	93.25	6.92
0.80	96.46	92.94	7.06	93.45	93.18	6.55
0.90	95.34	92.16	7.84	93.75	92.91	6.25

a specific metric can be used as a key for optimal model that can be used for different kind of application.

Following the six samples outcome metrics, it can be obviously noted that that the model's performed well for all the test samples, based on all metrics for all the thresholds. Notably, the performance of the model varies depending on the test model. This attest to the variations of proprieties and characteristics of the different samples, which yield to different performance metrics. It was reported that model performs well in terms of average accuracy for all the thresholds, for example, a high average accuracy for all the six test samples (exceeding 95% for all samples except sample 2 where average accuracy vary between 88% to 95%). Additionally, the model achieved F1-socre above 90% for all thresholds on all test samples except sample 2 and sample 4, where sample 2 F1score varies between 72% and 85%, and sample 4 F1-score varies between 88% and 93%. Thus, implying positive sensitivity and specificity in offensive language detection. Contrarily, reported false rates (FNR) and (FPR) is considerably low, which indicates that the percentage of producing a false prediction is quite low.

Looking into the most common threshold used for deep learning application, which is 0.5 confidence score, the model produces an average accuracy of sample's detection of around 95% to 98%. The system also produced a TPR/recall of about 91% to 97%, which implies that the rate of rejection (FNR) is quite low of merely 4% to 8% FNR. Contrary, the systems exhibit a precise detection rate that can be interpreted using precision metrics as it swings between 81% and 93%. That contributes to the low false alarms (FPR) detected by the system. On the other hand, the model produced a good sensitivity and specificity, that can be elaborated with F1-score figures that is between 85% and 95%. Therefore, choosing the suitable threshold is crucial and depends on the application and test samples itself and depends on what is the acceptable rate of false alarms and false misses that can be more elaborated using ROC and DET curves.

C. AVERAGE MODEL PERFORMANCE AND DETECTION CURVES

The performance assessment of the proposed models on the detection of foul language for the six test samples have varied depending on the differences. Therefore, average model performance reported in Table 14 and Table 15. Table 14 and Table 15 presents the average performance of system for all the six test samples, in which macro average and weighted average were computed. Macro average is the average of the sum of all figures divided by the total samples, whereas weighted average was computed where each sample's figures contribute to the average numbers based on the weight of the foul words within each sample compared to total foul words for the whole dataset. The average metrics were computed for all the thresholds. However, here we just highlight similar thresholds to the thresholds analysis tables. Therefore, the models varied performance can be highlighted based on different thresholds. For instance, how precision is affected with varying thresholds. Hence, an operation threshold can be chosen depending on the optimal metrics required for the detection of profanities like minimizing FNR or minimizing FPR.

Looking at the average metrics, it can be noteworthy that increasing threshold contributes to a slight drop in average accuracy (from 97.47% at 0.1 threshold to 95.34% at 0.9 threshold for weighted average) and TPR/recall (from 95.68% at 0.1 threshold to 92.16% at 0.9 threshold for weighted average). Hence, FNR increases with threshold increment (from 4.32% at 0.1 threshold to 7.84% at 0.9 threshold for weighted average). In contrast, precision increases dramatically with threshold increment from 85.12% at 0.1 threshold to 93.75% at 0.9 threshold for weighted average). Therefore, a huge drop in false detection (FPR) (from 14.88% at 0.1 threshold to 6.25% at 0.9 threshold for weighted average) occurred with threshold increment.

On the other hand, F1-score that is calculated based on precision and recall varies with changing threshold and varies between 89.96% and 93.25%. It is known that choosing the operation points depends on the rates the user wishes to achieve. For example, if F1-score matters more than all the other metrics, then choosing 0.7 confidence score as the best performing point, as it yields to the highest F1-score based on the weighted average of around 93.25% F1-score. ROC curve, AUC, and DET curve is another way of visualizing the performance of the model at all operating points. Figure 5 presents the ROC curves for all samples and the averaged figures, in which the operating curves and the relationship between TPR and FPR can be visually interpreted.

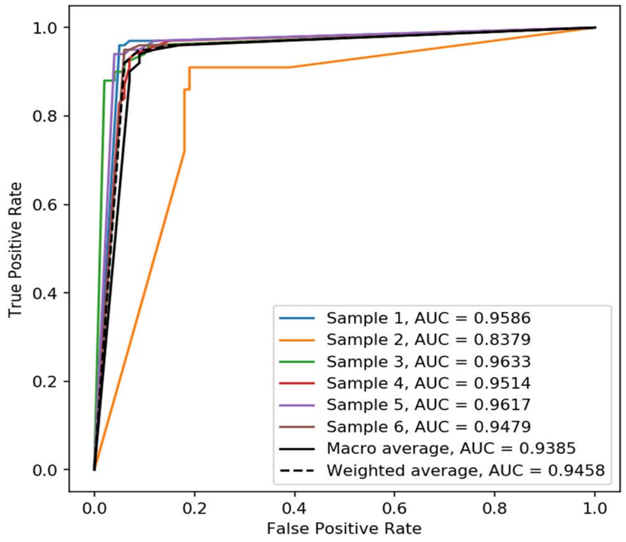

FIGURE 5. ROC curves for all samples and averaged figures.

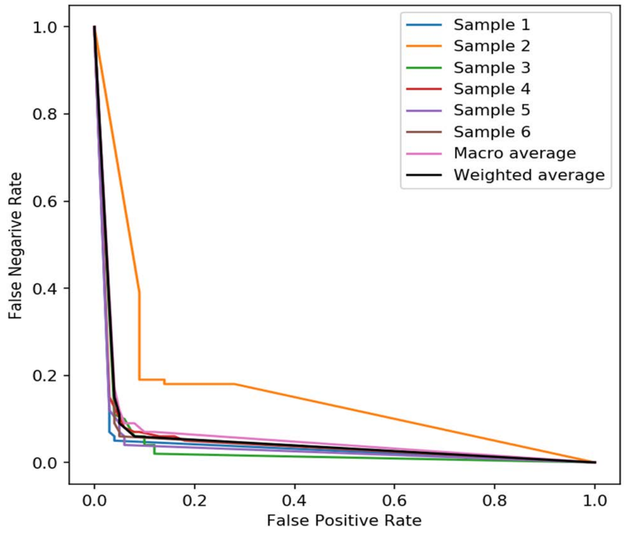

FIGURE 6. DET curves for all samples and averaged figures.

Additionally, AUC were computed based on the ROC curves for all samples with averaged AUC values. In contrary, DET curves in Figure 6 illustrates the relation between the false rates (FNR vs FPR), which indicates the values of errors can be produced at certain operating points.

AUC for ROC curves is another way to interpreter the overall model for all the operating points of thresholds from zero to one threshold/confidence score. AUC is the area under the ROC curves, which is the relation between (TPR vs FPR), in which each operating points highlights the number of correct predictions rate of foul language and the rate of wrong predictions of foul language while it is a normal speech. Table 16 elaborates the averaged values of AUC and AUC for each sample.

TABLE 16. AUC metrics for all samples and averaged figures.

Samples	AUC (%)
Sample 1	95.86
Sample 2	83.79
Sample 3	96.33
Sample 4	95.14
Sample 5	96.17
Sample 6	94.79
Macro average	93.85
Weighted average	94.58

TABLE 17. Processing and inference time of profanity censorship system.

Processing time (s)	Model inference time (ms/segemnt)
0.46 for each second	2.63

Figure 5 and Table 16 elaborates the model's performance in terms of the AUC metric. It shows the AUC for all the samples to visualize the system's performance of the different samples. The highest AUC was achieved by sample 3 of 96.33%, while the lowest was achieved by sample 2 of around 83.79%. The AUC for the rest of the sample's swings between 94.79% for sample 6 and 96.17% for sample 5. The model average performance can be highlighted with the average AUC values, where the model macro average AUC is 93.85%, and the weighted average is around 94.85. Therefore, the overall model performance lies within AUC of 94.85% and 93.85% for all the different samples.

D. SPEED ANALYSIS OF PROFANITY CENSORSHIP SYSTEM

Table 17 shows the inference time of state-of-the-art CNN model and the system overall process time from the input of continuous speech, segmentation, through detection and time estimation, where it was found that the proposed CNN has inference time of 2.63 ms (0.00263 seconds) calculated from the time step of applying the spectrogram image sample at the input to the time step of model's prediction. The reason behind that is the minimum number of parameters and lightweight CNN of small filters and few layers. According to Table 17, the average process time per each second of the long audio samples, which can be defined as the average time taken to process the input sample through all steps from segmentation to automated detection per each second, which is 0.46 seconds. This means each second of the long audio will be processed completely in 0.46 seconds, that makes this process to be real time process and even faster than the human manual films' detection, filtering, and censorship of inappropriate speech content. For example, sample 1 consist of 371 seconds in total. However, the average time will be taken to pass through the developed automated detection for film censorship process, will be around 170.66 seconds, which is less than half the length of the original sample. Hence, the proposed system yield in saving time compared to manual detection and censorship process. In addition to the

TABLE 18. Macro average performance metrics of baseline 1.

Threshold	Avg. accuracy (%)	Recall (%)	FNR (%)	Precision (%)	F1-score (%)	FPR (%)
0.10	92.32	91.21	8.79	83.91	87.41	17.09
0.25	92.07	91.03	8.97	86.87	88.90	12.26
0.50	91.67	89.76	10.24	88.91	89.33	9.31
0.60	90.93	88.13	11.87	89.72	88.92	8.63
0.70	90.93	87.71	12.29	91.27	89.45	7.60
0.80	90.43	87.71	12.29	92.12	89.86	7.28
0.90	89.03	85.91	14.09	93.21	89.41	6.73

TABLE 19. Weighted average performance metrics of baseline 1.

Threshold	Avg. accuracy (%)	Recall (%)	FNR (%)	Precision (%)	F1-score (%)	FPR (%)
0.10	96.23	94.54	5.46	84.27	88.97	15.98
0.25	96.11	93.98	6.02	88.32	90.98	11.46
0.50	95.96	93.34	6.66	90.42	91.83	9.13
0.60	95.57	91.87	8.13	91.31	91.57	8.28
0.70	95.39	91.48	8.52	92.46	91.94	7.26
0.80	94.90	90.86	9.14	92.94	91.86	6.91
0.90	93.71	89.55	10.45	93.50	91.42	6.49

TABLE 20. Macro average performance metrics of baseline 2.

Threshold	Avg. accuracy (%)	Recall (%)	FNR (%)	Precision (%)	F1-score (%)	FPR (%)
0.10	91.80	90.34	9.66	81.42	85.65	18.58
0.25	91.64	89.98	10.02	85.00	87.42	15.00
0.50	91.42	89.19	10.81	86.88	88.02	13.12
0.60	90.93	87.78	12.22	87.79	87.79	12.21
0.70	90.79	87.48	12.52	89.08	88.27	10.92
0.80	90.34	87.23	12.77	89.69	88.44	10.31
0.90	89.09	85.94	14.06	90.38	88.10	9.62

achieved high AUC metrics values, which shows this system feasibility for speech films' censorship. The results of speed analysis were reported based on system's evaluation using the CPU to mimic the real-life scenario of films screening, detection, and censorship that is always executed using CPUs.

E. BENCHMARK ANALYSIS

This work proposed to use novel datasets for model's training including MMUTM and TAPAD offensive language dataset. Additionally, this research proposed the use of a novel test dataset containing continuous speech dataset with frequent utterances of foul words. Due to the lack of studies on spoken profanity detection from continuous audio input using neural networks, this work results were mainly compared against two baseline models of which one is a recent work on MMUTM dataset [66]. The recent research has produced deep learning models for the foul language recognition of isolated keywords input on the MMUTM foul language dataset [66], but the developed RNN model were not tested on continuous real-world test samples. In this research, we addressed these issues and designed different novel models for the foul language detection that were tested on continuous audio samples. Hence, the recent work RNN model were used as baseline 1 model. Additionally, a 2-convolution layers CNN model were constructed and used as baseline 2 model, as 2-layers CNN is a common model architecture that were used for keywords and caustic sounds detection [9]. The two baseline models were re-trained and tested using the same protocol to train and test the current model and compared using several evaluation metrics.

Table 18 presents the macro average metrics of baseline 1 [66] over all test samples, whereas Table 14 details the figures of current model macro average metrics. Based on the comparison between Table 18 and Table 14, it can be noteworthy that proposed CNN model outperforms baseline 1 model based on most of the evaluation metrics except for precision and FPR, where the precision and FPR for both models are slightly the same. Based on the macro average metrics for both models, current model outperformed baseline 1 by around (5% to 6%) average accuracy, (2% to 4%) recall/TPR and FNR, and about 1% F1-score for all the thresholds. Table 19 presents the weighted average metrics of baseline 1, whereas Table 15 details the weighted average metrics of proposed CNN model. Based on the comparison between Table 19 and Table 15, it is noted that current model outperforms baseline 1 model based on all the evaluation metrics. Based on the weighted average metrics for both models, current model outperformed baseline 1 by around (1% to 2%) average accuracy, (2% to 3%) recall/TPR and FNR, (0.5% to 1%) precision, and about 1% F1-score. Thus, proposed model outperformed baseline 1.

Table 20 highlights the macro average metrics of baseline 2 over all test samples, whereas Table 14 details the figures of current model macro average metrics. Based on the comparison between Table 20 and Table 14, it can be noted that current CNN model outperforms baseline 2 model based on all the evaluation metrics Based on the macro average metrics for both models, current model outperformed baseline 2 by around 6% average accuracy, (2% to 5%) recall/TPR and FNR, (1% to 3%) precision, and about (1% to 3%) F1-score. Table 21 presents the weighted average metrics of baseline 2, whereas Table 15 details the weighted average metrics of proposed CNN model. Based on the comparison between Table 21 and Table 15, it is noteworthy that current model outperforms baseline 2 model based on all the evaluation metrics. Based on the weighted average metrics for both models, current model outperformed baseline 1 by around (1% to 2%) average accuracy, (3% to 6%) recall/TPR and FNR, (1% to 2%) precision, and about (2% to 3%) F1-score for all the thresholds. Hence, proposed model outperformed baseline 2.

ROC curve, AUC, and DET curve is a more favorable way of visualizing the performance of several model at all operating points and compare the different performance of each model. Figure 7 presents the ROC curves for current and baseline models, in which the operating curves and the relationship between TPR and FPR can be visually interpreted. Additionally, average AUC values were computed based on

TABLE 21. Weighted average performance metrics of baseline 2.

Threshold	Avg. accuracy (%)	Recall (%)	FNR (%)	Precision (%)	F1-score (%)	FPR (%)
0.10	94.99	93.40	6.60	83.41	87.97	17.09
0.25	94.81	92.83	7.17	87.30	89.90	12.26
0.50	94.56	91.85	8.15	89.80	90.79	9.31
0.60	94.03	90.10	9.90	90.54	90.31	8.63
0.70	93.93	89.50	10.50	91.83	90.63	7.60
0.80	93.35	88.77	11.23	92.42	90.53	7.28
0.90	92.07	86.94	13.06	93.24	89.93	6.73

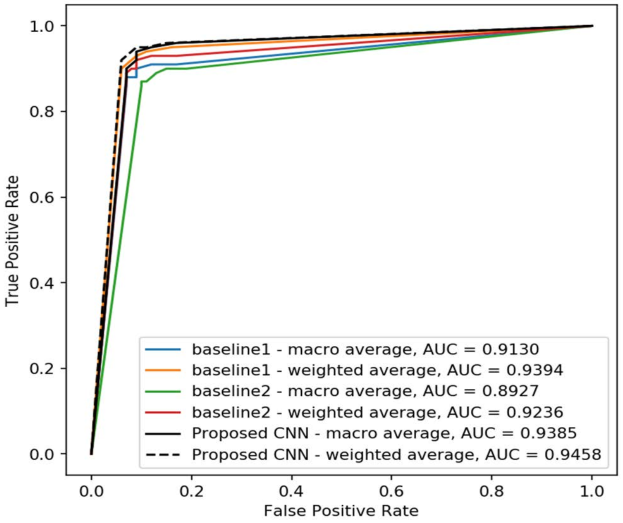

FIGURE 7. ROC curves for current and baseline models.

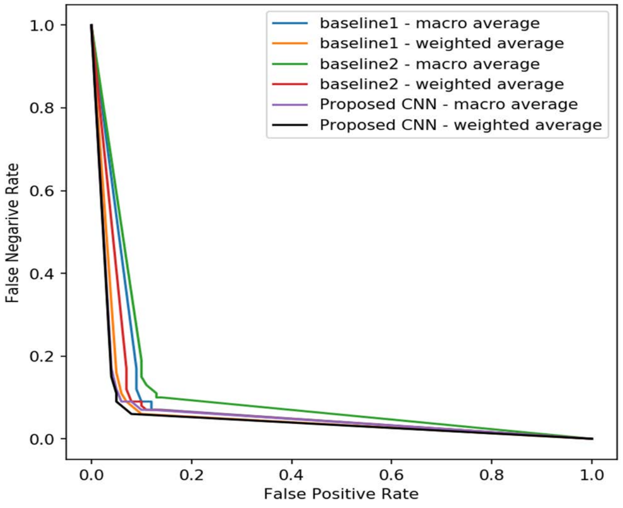

FIGURE 8. DET curves for current and baseline models.

the ROC curves. In contrast, DET curves in Figure 8 illustrates the relation between the false rates (FNR vs FPR), which indicates the values of errors can be produced at all operating points of thresholds.

TABLE 22. AUC metrics of current and baseline models for the novel inappropriate speech dataset.

Model	AUC (%)
Baseline 1 – macro average [66]	91.30
Baseline 1 – weighted average [66]	93.94
Baseline 2 – macro average [9]	89.27
Baseline 2 – weighted average [9]	92.36
Proposed CNN Model – macro average	**93.85**
Proposed CNN Model – weighted average	**94.58**

Figure 7 and Table 22 highlight the model's performance in terms of AUC metric. The highest AUC was achieved by sample 3 of around 96.33%, while the lowest was achieved by sample 2 of about 83.79%. The AUC for the rest of the samples varies with about 94.79% for sample 6 and 96.17% for sample 5. The model average performance can be highlighted with the average AUC values, where the model macro average AUC is 93.85%, and the weighted average is around 94.85. Therefore, the overall model performance lies within AUC of 94.85% and 93.85% for all the different samples. Notably, the model performs differently with different evaluation samples attesting to the different properties and characteristics of different samples. However. The model performs positively regardless of the variations.

In Table 22, we showed the outperforming results of the proposed system based on AUC metric, which is significantly better than other baseline systems, where proposed model outperformed baseline 1 algorithms with 2.55% macro average AUC and weighted average AUC of 0.64%. On the other hand, current model outperformed baseline 2 algorithms with 4.58% macro average AUC and weighted average AUC of 2.22%. Thus, current model outperformed baseline models in terms of AUC and all other metrics.

1) BENCHMARK OF ASR-BASED CENSORSHIP SYSTEM

Given the scarcity of experiments on inappropriate speech content detection, the first past of this subsection highlighted a comparative analysis using acoustic-based systems for profanity detection. On the other hand, this part benchmark the current work against previous work that uses ASR systems for the detection of profanities. Recent research proposed a solution for analyzing the video, which helps to identify the profane content through the use of text detection approaches after videos being transcribed by means of ASR systems [70]. The audio samples were extracted from the input video. Then, audio samples were converted into text using Speech-to-Text library for detection and localization of profane words. The text data samples were checked against a profanity list of words. The proposed system was tested with 50 videos collected from various sources like Facebook, YouTube etc. Additionally, some of the videos were made by authors containing profane keywords. The total length of test samples was only 1734 second (∼ 28.9 minutes). The developed profanity detection using ASR systems and text detection approaches achieved an accuracy of around 85.03% on the reported dataset [70].

The reported ASR-based system containing two stages that are Speech-to-Text phase, and text detection approach, was retrained on the list of profanities proposed in this work to

TABLE 23. Comparative figures of current and ASR-based systems for inappropriate speech detection and localization.

Profanity Detection System	AUC (%)	Accuracy (%)	Precision (%)	F1-score (%)	Time (s)
ASR-based system – macro average [70]	88.69	88.51	86.84	87.67	**0.45**
ASR-based system – weighted average [70]	91.02	89.70	90.52	90.11	
Current acoustic system – macro average	**93.85**	**97.45**	**90.69**	**92.24**	0.46
Current acoustic system – weighted average	**94.58**	**97.36**	**91.05**	**92.88**	

TABLE 24. Performance metrics of proposed and pre-trained CNN models for inappropriate speech detection and localization.

Average type	Model	AUC (%)	Accuracy (%)	Recall (%)	Precision (%)	F1-score (%)
Macro average	MobileNet [71]	90.02	92.91	89.39	84.96	87.12
	Inception v3 [72]	89.47	92.91	89.70	83.63	86.56
	Alexnet [73]	89.02	93.07	90.56	81.64	85.87
	Resnet50 [74]	91.78	95.88	92.37	88.36	90.32
	Current model	**93.85**	**97.45**	**93.94**	**90.69**	**92.24**
Weighted average	MobileNet [71]	90.55	92.85	90.45	85.23	87.76
	Inception v3 [72]	90.16	92.82	90.29	84.31	87.20
	Alexnet [73]	90.39	92.99	90.45	84.76	87.51
	Resnet50 [74]	92.43	95.15	93.62	91.02	90.97
	Current model	**94.58**	**97.36**	**94.83**	**91.05**	**92.88**

TABLE 25. Comparison between the proposed and pre-trained CNN models in terms of network parameters.

Model	Model size (MB)	Training Parameters	No. of layers	Inference time (ms)	Processing time (s)
MobileNet [71]	16	4.3M	28	82.71	0.54
Inception v3 [72]	92	23.9M	48	90.23	0.55
Alexnet [73]	244	61M	25	18.08	0.48
Resnet50 [74]	98	25.6M	50	51.23	0.51
Current model	**13**	**46.7k**	**15**	**2.36**	**0.46**

benchmark current work against ASR-based system. Then, the ASR-based system was tested using the six video samples used to test the current proposed system. The evaluation metrics and comparative figures of ASR-based system and current system are highlighted in Table 23. The two models were evaluated using AUC, accuracy, precision, and F1-score metrics. Additionally, the systems were compared based on the overall processing time of each second of the continuous input sample for the detection of inappropriate speech content. CPU was used to only assess the processing time as reported in the table.

Table 23 details the evaluation metrics and processing time per every second of test input for ASR-based system and current system that utilizes acoustic features for the detection of profanities within continuous audio input. Notably, the proposed acoustic system outperformed the ASR-based system in terms of evaluation metrics. For example, proposed system achieved AUC of 94.58% weighted average and 93.85% macro average, while ASR-based system achieved AUC of 91.02% weighted average and 88.69% macro average. The proposed acoustic system using CNN also outperformed ASR-based system by significant margin in terms of accuracy, precision, and F1-score. This can be attested to the main disadvantage of the ASR-based system, which is the use of multiple pipeline blocks serially, where text profanity detection and localization happened after speech-to-text transcription. Therefore, a failure in accurate transcription led to failure in the detection stage. Hence, the overall detection and censorship performance dropped significantly.

The ASR-based system outperformed acoustic system in terms of processing time by small margin of 0.01 second (10 ms) as ASR-based system requires 0.45 seconds, while ASR-based system requires 0.46 seconds for each second of the input for the whole process that results in the detection and localization of profanity within a given continuous audio. This attested to the multistage of acoustic system including time for segmentation process and inference time. On the other hand, ASR-based system requires multiple inference times of text detection model and ASR that explicitly contains language and acoustic models and does not require an input segmentation process. It is also noteworthy that only one inference is required for acoustic system when using CNN model, while ASR-based system requires two inferences for Speech-to-Text and text detection models.

This experiment was performed on a particular dataset of a spoken English profane words with positive outcomes in any derivation of the profanities. Nevertheless, the proposed system performance may be varied by using a different range of English verbal words or spoken utterances from different language, as the proposed model uses the direct acoustic features of utterances for the detection, unlike ASR systems where spoken terms can be transcribed based on the language models used in ASR models and accommodate wider range of keywords. However, the use of ASR models suffers of the issues that majorly concern a large dataset and large computational cost, in which the two major issues is solved in this work for the development of profane words detector. Additionally, ASR systems uses a few stages of models like acoustic models and language models. In this context, an additional text detector will need to be applied to locate the inappropriate speech content. Therefore, ASR-based systems for the detection of profane words suffers of performance metrics drop due to the sequenced models, as a failure in one stage leads to performance drop in the following stage.

2) BENCHMARK OF PRE-TRAINED MODELS

The proposed CNN model for profanity detection and censorship was further analyzed and compared with four different pre-trained CNN models, which are MobileNet [71], Inception-v3 [72], AlexNet [73], and ResNet-50 [74] as detailed in Table 24 and Table 25. The models are compared in terms of network architecture characteristics (e.g., model size and parameters) and performance metrics including AUC, accuracy, recall, precision, and F1-score. The models are compared based on the average metrics including macro and weighted averages.

The proposed system outperforms the other four pre-trained models in all evaluation metrics in both macro and weighted averages as highlighted in Table 24. The comparison table shows a different metrics for the different models. For example, the models order based on AUC metric are our proposed model, Resnet50, MobileNet, Inception-v3, then Alexnet. However, the order changes when comparing models based on accuracy, where proposed model achieved best accuracy, followed by Resnet50, Alextnet, then Inception-v3 and MobileNet achieve similar accuracy. Proposed CNN model outperformed the pre-trained model in terms of all metrics. For instance, proposed model outperformed pre-trained model based on macro averaged AUC by around 2.07%, 3.93%, 4.38%, and 4.83% compared to Resent50, MobileNet, Inception-v3, and Alexnet, respectively. Furthermore, proposed model outperformed pre-trained model based on macro averaged accuracy by around 1.57%, 4.38%, 4.54%, and 4.54% compared to Resent50, Alexnet, Inception-v3, and MobileNet, respectively.

The proposed model is also compared with the other pre-trained models in terms of network characteristics such as model size, training parameters, number of layers, inference time, and processing time as highlighted is Table 25. Proposed CNN model does not only achieve the highest detection accuracy but also has the smallest model size of 13 MB and lowest training parameters of only 46.7k. On the other hand, the largest model size belongs to Alexnet of around 244 MB. Our model achieves better performance by using only 15 of smaller size and filter, which helps to reduce the computational cost and time. In contrast, Resnet50 has the highest number of layers of up to 50 layers. The inference and overall processing time of proposed model are 2.36ms and 0,45s, respectively. Our model inference time is at almost eight times less than those of other pre-trained models. These outstanding values for the network parameters of proposed CNN model and its superior performance prove the effectiveness and efficiency of our system in automated detection and localization of inappropriate speech for censorship purpose.

As the proposed solution uses only acoustic features of a given number of profane words. Therefore, the proposed system could be used for different English profane terms or any other profanities from different language provided the use of the same procedure of data preparation that would match the acoustic features extraction methods and the proposed architecture. In this case, the CNN model must adapt to the new keywords. Several approaches are recommended to tackle the study gap of including wider range of foul words, such as executing a full or partial system retraining or introducing the well-known transfer learning approaches for CNN networks [25]. As this work was designed to detect direct utterances of profanities in continuous stream, it is recommended to consider several future developments for censorship and films rating researches. The context in which keyword is uttered is crucial to define a set of words that could represent the keyword. Therefore, considering the sequence and context of uttered words is recommended in future works.

V. CONCLUSION

This research suggested the implementation of CNN model for the detection and localization of spoken foul language in continuous speech samples with static keyword detection mode test for automated video/audio/film censorship. The current work utilizes a novel dataset of foul languages to train the model. MMUTM and TAPAD datasets were manually labeled with 2 annotations (Foul vs Normal). The CNN model was trained to classify the labels of pre-segmented isolated samples, whereas current model was tested with continuous incoming audio samples for offensive language identification. The novel test dataset consists of several real-world video samples with high rate of offensive words per minute. The model input was an extracted features of the audio samples in a form of Log-Mel spectrogram images, while the output of the whole system contains the detected foul word and timestamps of profanity occurrences within lengthy audio samples.

The proposed system performed differently based on the different properties and characteristics of test samples. However, the overall foul language detection system has performed positively with macro average accuracy ranging from 95.11% to 97.67% and weighted average accuracy of 95.34% to 97.47% for all the operating points of thresholds. Furthermore, the reported F1-score metric for model performance showed a balance between sensitivity and specificity of proposed CNN by achieving F1-score ranging from 88.54% to 90.45% macro averaged and 89.96% to 92.91% for weighted average metrics. Additionally, current model achieved a high AUC metric of the ROC curve of around 93.85% macro averaged and 94.58% weighted average AUC metrics.

The proposed lightweight CNN model was benchmarked against two baseline models that uses only acoustic features on the novel offensive language dataset. It is reported that the current model outperformed the acoustic baseline algorithms in terms of performance metrics. We showed the outperforming results of the proposed system based on AUC metric, which is significantly better than other baseline models, where proposed model outperformed baseline 1 algorithms with 2.55% macro average AUC and weighted average AUC of 0.64%. On the other hand, proposed system outperformed baseline 2 model with 4.58% macro average AUC and weighted average AUC of 2.22%. Thus, current model outperformed baseline models in terms of AUC and all other metrics. Additionally, proposed acoustic system outperformed ASR-based system for profanity detection based on the evaluation metrics including AUC, accuracy, precision, and F1-score.

This work also demonstrated that proposed system for audible and speech content processing and detection of inappropriate content within, had performed positively in terms of inference and overall process speed. It was found that the proposed CNN has inference time of 2.63 ms (0.00263 seconds), which is attested to the light-weight structure of developed model. Furthermore, the average time taken to process the input sample through all steps from segmentation to automated detection per each second, which is 0.46 seconds. This means each second of the long audio will be processed completely in 0.46 seconds, that makes this process to be real time process and even faster than the human manual films' detection, filtering, and censorship of inappropriate speech content. This attested to the light-weight structure of CNN architecture, which make the process and inference to be faster and suitable for content screening, filtering, and censorship.

REFERENCES

[1] K. Tuttle. (2018). *The Profanity Problem: Swearing in Movies*. Movie Babble. Accessed: Jan. 11, 2021. [Online]. Available: https://moviebabble.com/2018/02/20/the-profanity-problem-swearing-in-movies/

[2] S. Day. (2018). *Cursing Negatively Affects Society*. Baker Orange, Baker Univ. Media. Accessed: Jan. 6, 2021. [Online]. Available: https://thebakerorange.com/27914/voices/cursing-negatively-affects-society/

[3] S. K. Gaikwad, B. W. Gawali, and P. Yannawar, "A review on speech recognition technique," *Int. J. Comput. Appl.*, vol. 10, no. 3, pp. 16–24, Nov. 2010, doi: 10.5120/1462-1976.

[4] H. Caranica, H. Cucu, A. Buzo, and C. Burileanu, "Survey on multilingual spoken term detection," *Romanian J. Inf. Sci. Technol.*, vol. 20, no. 3, pp. 210–221, 2010.

[5] M. Versteegh, R. Thiollière, T. Schatz, X. N. Cao, X. Anguera, A. Jansen, and E. Dupoux, "The Zero resource speech challenge 2015," in *Proc. Interspeech*, 2015, pp. 3169–3173.

[6] G. Deekshitha and L. Mary, "Multilingual spoken term detection: A review," *Int. J. Speech Technol.*, vol. 23, no. 3, pp. 653–667, Sep. 2020, doi: 10.1007/s10772-020-09732-9.

[7] J. Tejedor, D. T. Toledano, P. Lopez-Otero, L. Docio-Fernandez, C. Garcia-Mateo, A. Cardenal, J. D. Echeverry-Correa, A. Coucheiro-Limeres, J. Olcoz, and A. Miguel, "Spoken term detection ALBAYZIN 2014 evaluation: Overview, systems, results, and discussion," *EURASIP J. Audio, Speech, Music Process.*, vol. 2015, no. 1, pp. 1–27, Dec. 2015, doi: 10.1186/s13636-015-0063-8.

[8] A. Mandal, K. R. P. Kumar, and P. Mitra, "Recent developments in spoken term detection: A survey," *Int. J. Speech Technol.*, vol. 17, no. 2, pp. 183–198, Jun. 2014, doi: 10.1007/s10772-013-9217-1.

[9] I. Lopez-Espejo, Z.-H. Tan, J. H. L. Hansen, and J. Jensen, "Deep spoken keyword spotting: An overview," *IEEE Access*, vol. 10, pp. 4169–4199, 2022, doi: 10.1109/ACCESS.2021.3139508.

[10] A. H. Michaely, X. Zhang, G. Simko, C. Parada, and P. Aleksic, "Keyword spotting for Google assistant using contextual speech recognition," in *Proc. IEEE Autom. Speech Recognit. Understand. Workshop (ASRU)*, Dec. 2017, pp. 272–278, doi: 10.1109/ASRU.2017.8268946.

[11] O. Vinyals and S. Wegmann, "Chasing the metric: Smoothing learning algorithms for keyword detection," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2014, pp. 3301–3305, doi: 10.1109/ICASSP.2014.6854211.

[12] Y. Zhuang, X. Chang, Y. Qian, and K. Yu, "Unrestricted vocabulary keyword spotting using LSTM-CTC," in *Proc. Interspeech*, Sep. 2016, pp. 938–942, doi: 10.21437/Interspeech.2016-753.

[13] D. R. H. Miller, M. Kleber, C.-L. Kao, O. Kimball, T. Colthurst, S. A. Lowe, R. M. Schwartz, and H. Gish, "Rapid and accurate spoken term detection," in *Proc. Interspeech*, Aug. 2007, pp. 314–317, doi: 10.21437/Interspeech.2007-174.

[14] G. Chen, O. Yilmaz, J. Trmal, D. Povey, and S. Khudanpur, "Using proxies for OOV keywords in the keyword search task," in *Proc. IEEE Workshop Autom. Speech Recognit. Understand.*, Dec. 2013, pp. 416–421, doi: 10.1109/ASRU.2013.6707766.

[15] Y. Wang and Y. Long, "Keyword spotting based on CTC and RNN for Mandarin Chinese speech," in *Proc. 11th Int. Symp. Chin. Spoken Lang. Process. (ISCSLP)*, Nov. 2018, pp. 374–378, doi: 10.1109/ISCSLP.2018.8706631.

[16] H. Hu, W. Zhang, L. Feng, Z. Wei, and Q. Chen, "Attention-based end-to-end keywords spotting," in *Proc. 9th Int. Conf. Comput. Pattern Recognit.*, Oct. 2020, pp. 479–483, doi: 10.1145/3436369.3437430.

[17] R. Rikhye, Q. Wang, Q. Liang, Y. He, D. Zhao, Y. Huang, A. Narayanan, and I. McGraw, "Personalized keyphrase detection using speaker and environment information," in *Proc. Interspeech*, Aug. 2021, pp. 4204–4208, doi: 10.21437/Interspeech.2021-204.

[18] G. Chen, C. Parada, and T. N. Sainath, "Query-by-example keyword spotting using long short-term memory networks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2015, pp. 5236–5240, doi: 10.1109/ICASSP.2015.7178970.

[19] S. Chai, Z. Yang, C. Lv, and W.-Q. Zhang, "An end-to-end model based on TDNN-BiGRU for keyword spotting," in *Proc. Int. Conf. Asian Lang. Process. (IALP)*, Nov. 2019, pp. 402–406, doi: 10.1109/IALP48816.2019.9037714.

[20] M. Sun, D. Snyder, Y. Gao, V. Nagaraja, M. Rodehorst, S. Panchapagesan, N. Strom, S. Matsoukas, and S. Vitaladevuni, "Compressed time delay neural network for small-footprint keyword spotting," in *Proc. Interspeech*, Aug. 2017, pp. 3607–3611, doi: 10.21437/Interspeech.2017-480.

[21] J. Hou, Y. Shi, M. Ostendorf, M.-Y. Hwang, and L. Xie, "Region proposal network based small-footprint keyword spotting," *IEEE Signal Process. Lett.*, vol. 26, no. 10, pp. 1471–1475, Oct. 2019, doi: 10.1109/LSP.2019.2936282.

[22] J. Hou, Y. Shi, M. Ostendorf, M.-Y. Hwang, and L. Xie, "Region proposal network based small-footprint keyword spotting," *IEEE Signal Process. Lett.*, vol. 26, no. 10, pp. 1471–1475, Oct. 2019, doi: 10.1109/LSP.2019.2936282.

[23] G. Chen, C. Parada, and G. Heigold, "Small-footprint keyword spotting using deep neural networks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2014, pp. 4087–4091, doi: 10.1109/ICASSP.2014.6854370.

[24] X. Wang, S. Sun, C. Shan, J. Hou, L. Xie, S. Li, and X. Lei, "Adversarial examples for improving end-to-end attention-based small-footprint keyword spotting," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2019, pp. 6366–6370, doi: 10.1109/ICASSP.2019.8683479.

[25] T. N. Sainath and C. Parada, "Convolutional neural networks for small-footprint keyword spotting," in *Proc. Interspeech*, Sep. 2015, pp. 1478–1482, doi: 10.21437/Interspeech.2015-352.

[26] K. Choi, G. Fazekas, M. Sandler, and K. Cho, "Convolutional recurrent neural networks for music classification," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Mar. 2017, pp. 2392–2396, doi: 10.1109/ICASSP.2017.7952585.

[27] R. Tang and J. Lin, "Deep residual learning for small-footprint keyword spotting," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2018, pp. 5484–5488, doi: 10.1109/ICASSP.2018.8462688.

[28] R. Alvarez and H.-J. Park, "End-to-end streaming keyword spotting," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2019, pp. 6336–6340, doi: 10.1109/ICASSP.2019.8683557.

[29] O. Rybakov, N. Kononenko, N. Subrahmanya, M. Visontai, and S. Laurenzo, "Streaming keyword spotting on mobile devices," in *Proc. Interspeech*, Oct. 2020, pp. 2277–2281, doi: 10.21437/Interspeech.2020-1003.

[30] B. U. Pedroni, S. Sheik, H. Mostafa, S. Paul, C. Augustine, and G. Cauwenberghs, "Small-footprint spiking neural networks for power-efficient keyword spotting," in *Proc. IEEE Biomed. Circuits Syst. Conf. (BioCAS)*, Oct. 2018, pp. 1–4, doi: 10.1109/BIOCAS.2018.8584832.

[31] B. Liu, S. Nie, Y. Zhang, S. Liang, Z. Yang, and W. Liu, "Loss and Double-edge-triggered detector for robust small-footprint keyword spotting," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2019, pp. 6361–6365, doi: 10.1109/ICASSP.2019.8682534.

[32] P. M. Sørensen, B. Epp, and T. May, "A depthwise separable convolutional neural network for keyword spotting on an embedded system," *EURASIP J. Audio, Speech, Music Process.*, vol. 2020, no. 1, pp. 1–14, Dec. 2020, doi: 10.1186/s13636-020-00176-2.

[33] I. Lopez-Espejo, Z.-H. Tan, and J. Jensen, "Exploring filterbank learning for keyword spotting," in *Proc. 28th Eur. Signal Process. Conf. (EUSIPCO)*, Jan. 2021, pp. 331–335, doi: 10.23919/Eusipco47968.2020.9287772.

[34] L. Wang, R. Gu, N. Chen, and Y. Zou, "Text anchor based metric learning for small-footprint keyword spotting," in *Proc. Interspeech*, Aug. 2021, pp. 4219–4223, doi: 10.21437/Interspeech.2021-136.

[35] B. Pattanayak, J. K. Rout, and G. Pradhan, "Adaptive spectral smoothening for development of robust keyword spotting system," *IET Signal Process.*, vol. 13, no. 5, pp. 544–550, Jul. 2019, doi: 10.1049/iet-spr.2019.0027.

[36] A. Berg, M. O'Connor, and M. T. Cruz, "Keyword transformer: A self-attention model for keyword spotting," in *Proc. Interspeech*, Aug. 2021, pp. 4249–4253, doi: 10.21437/Interspeech.2021-1286.

[37] M. Muhsinzoda, C. C. Corona, D. A. Pelta, and J. L. Verdegay, "Activating accessible pedestrian signals by voice using keyword spotting systems," in *Proc. IEEE Int. Smart Cities Conf. (ISC2)*, Oct. 2019, pp. 531–534, doi: 10.1109/ISC246665.2019.9071684.

[38] Y. Chen, T. Ko, L. Shang, X. Chen, X. Jiang, and Q. Li, "An investigation of few-shot learning in spoken term classification," in *Proc. Interspeech*, Oct. 2020, pp. 2582–2586, doi: 10.21437/Interspeech.2020-2568.

[39] S. Watanabe, M. Delcroix, F. Metze, and J. R. Hershey, *New Era for Robust Speech Recognition*. Cham, Switzerland: Springer, 2017, doi: 10.1007/978-3-319-64680-0.

[40] Y. Tian, H. Yao, M. Cai, Y. Liu, and Z. Ma, "Improving RNN transducer modeling for small-footprint keyword spotting," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Jun. 2021, pp. 5624–5628, doi: 10.1109/ICASSP39728.2021.9414339.

[41] B. Kim, S. Chang, J. Lee, and D. Sung, "Broadcasted residual learning for efficient keyword spotting," in *Proc. Interspeech*, Jun. 2021, pp. 2582–2586, 2021, doi: 10.21437/Interspeech.2020-2568.

[42] P. Zhang and X. Zhang, "Deep template matching for small-footprint and configurable keyword spotting," in *Proc. Interspeech*, Oct. 2020, pp. 2572–2576, doi: 10.21437/Interspeech.2020-1761.

[43] M. Yu, X. Ji, B. Wu, D. Su, and D. Yu, "End-to-end multi-look keyword spotting," in *Proc. Interspeech*, Oct. 2020, pp. 66–70, doi: 10.21437/Interspeech.2020-1521.

[44] H. Mostafa, "Supervised learning based on temporal coding in spiking neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 7, pp. 3227–3235, Jul. 2018, doi: 10.1109/TNNLS.2017.2726060.

[45] P. Nakkiran, R. Alvarez, R. Prabhavalkar, and C. Parada, "Compressing deep neural networks using a rank-constrained topology," in *Proc. Interspeech*, Sep. 2015, pp. 1473–1477.

[46] M. Wöllmer, B. Schuller, and G. Rigoll, "Keyword spotting exploiting long short-term memory," *Speech Commun.*, vol. 55, no. 2, pp. 252–265, Feb. 2013, doi: 10.1016/j.specom.2012.08.006.

[47] H. Sundar, J. F. Lehman, and R. Singh, "Keyword spotting in multi-player voice driven games for children," in *Proc. Interspeech*, 2015, pp. 1660–1664, doi: 10.21437/Interspeech.2015-383.

[48] M. Zeng and N. Xiao, "Effective combination of DenseNet and BiLSTM for keyword spotting," *IEEE Access*, vol. 7, pp. 10767–10775, 2019, doi: 10.1109/ACCESS.2019.2891838.

[49] R. Tang, W. Wang, Z. Tu, and J. Lin, "An experimental analysis of the power consumption of convolutional neural networks for keyword spotting," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2018, pp. 5479–5483, doi: 10.1109/ICASSP.2018.8461624.

[50] H. Wu, Y. Jia, Y. Nie, and M. Li, "Domain aware training for far-field small-footprint keyword spotting," in *Proc. Interspeech*, Oct. 2020, pp. 2562–2566, doi: 10.21437/Interspeech.2020-1412.

[51] R. Shankar, C. M. Vikram, and S. R. M. Prasanna, "Spoken keyword detection using joint DTW-CNN," in *Proc. Interspeech*, Sep. 2018, pp. 117–121, doi: 10.21437/Interspeech.2018-1436.

[52] P. Warden, "Speech commands: A dataset for limited-vocabulary speech recognition," 2018, *arXiv:1804.03209*.

[53] I. Lopez-Espejo, Z.-H. Tan, and J. Jensen, "A novel loss function and training strategy for noise-robust keyword spotting," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 29, pp. 2254–2266, 2021, doi: 10.1109/TASLP.2021.3092567.

[54] I. Lopez-Espejo, Z.-H. Tan, and J. Jensen, "Improved external speaker-robust keyword spotting for hearing assistive devices," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 28, pp. 1233–1247, 2020, doi: 10.1109/TASLP.2020.2984089.

[55] Y. Yuan, Z. Lv, S. Huang, and L. Xie, "Verifying deep keyword spotting detection with acoustic word embeddings," in *Proc. IEEE Autom. Speech Recognit. Understand. Workshop (ASRU)*, Dec. 2019, pp. 613–620, doi: 10.1109/ASRU46091.2019.9003781.

[56] X. Wang, S. Sun, and L. Xie, "Virtual adversarial training for DS-CNN based small-footprint keyword spotting," in *Proc. IEEE Autom. Speech Recognit. Understand. Workshop (ASRU)*, Dec. 2019, pp. 607–612, doi: 10.1109/ASRU46091.2019.9003745.

[57] S. Myer and V. S. Tomar, "Efficient keyword spotting using time delay neural networks," in *Proc. Interspeech*, Sep. 2018, pp. 1264–1268, doi: 10.21437/Interspeech.2018-1979.

[58] R. Kumar, V. Yeruva, and S. Ganapathy, "On convolutional LSTM modeling for joint wake-word detection and text dependent speaker verification," in *Proc. Interspeech*, Sep. 2018, pp. 1121–1125, doi: 10.21437/Interspeech.2018-1759.

[59] A. H. Michaely, X. Zhang, G. Simko, C. Parada, and P. Aleksic, "Keyword spotting for Google assistant using contextual speech recognition," in *Proc. IEEE Autom. Speech Recognit. Understand. Workshop (ASRU)*, Dec. 2017, pp. 272–278, doi: 10.1109/ASRU.2017.8268946.

[60] D. Bahdanau, J. Chorowski, D. Serdyuk, P. Brakel, and Y. Bengio, "End-to-end attention-based large vocabulary speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Mar. 2016, pp. 4945–4949, doi: 10.1109/ICASSP.2016.7472618.

[61] C.-C. Chiu, T. N. Sainath, Y. Wu, R. Prabhavalkar, P. Nguyen, Z. Chen, A. Kannan, R. J. Weiss, K. Rao, E. Gonina, N. Jaitly, B. Li, J. Chorowski, and M. Bacchiani, "State-of-the-art speech recognition with sequence-to-sequence models," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2018, pp. 4774–4778, doi: 10.1109/ICASSP.2018.8462105.

[62] A. S. M. B. Wazir and J. H. Chuah, "Spoken Arabic digits recognition using deep learning," in *Proc. IEEE Int. Conf. Autom. Control Intell. Syst. (I2CACIS)*, Jun. 2019, pp. 339–344, doi: 10.1109/I2CACIS.2019.8825004.

[63] A. Pervaiz, F. Hussain, H. Israr, M. A. Tahir, F. R. Raja, N. K. Baloch, F. Ishmanov, and Y. B. Zikria, "Incorporating noise robustness in speech command recognition by noise augmentation of training data," *Sensors*, vol. 20, no. 8, p. 2326, Apr. 2020, doi: 10.3390/s20082326.

[64] T. Anvarjon, Mustaqeem, and S. Kwon, "Deep-Net: A lightweight CNN-based speech emotion recognition system using deep frequency features," *Sensors*, vol. 20, no. 18, p. 5212, Sep. 2020, doi: 10.3390/s20185212.

[65] A. S. B. Wazir, H. A. Karim, M. H. L. Abdullah, S. Mansor, N. AlDahoul, M. F. A. Fauzi, and J. See, "Spectrogram-based classification of spoken foul language using deep CNN," in *Proc. IEEE 22nd Int. Workshop Multimedia Signal Process. (MMSP)*, Sep. 2020, pp. 1–6, doi: 10.1109/MMSP48831.2020.9287133.

[66] A. S. Ba Wazir, H. A. Karim, M. H. L. Abdullah, N. AlDahoul, S. Mansor, M. F. A. Fauzi, J. See, and A. S. Naim, "Design and implementation of fast spoken foul language recognition with different end-to-end deep neural network architectures," *Sensors*, vol. 21, no. 3, p. 710, Jan. 2021, doi: 10.3390/s21030710.

[67] A. S. B. Wazir, H. A. Karim, N. AlDahoul, M. F. A. Fauzi, S. Mansor, L. H. L. Abdullah, H. S. Lyn, and T. Z. Zulkifli, "Spoken Malay profanity classification using convolutional neural network," in *Proc. IEEE Int. Conf. Signal Image Process. Appl. (ICSIPA)*, Sep. 2021, pp. 34–38, doi: 10.1109/ICSIPA52582.2021.9576781.

[68] R. Piyush. *The Abused Project Audio Dataset (TAPAD)*. Github. Accessed: Dec. 13, 2021. [Online]. Available: https://github.com/theabuseproject/tapad

[69] G. Sharma, K. Umapathy, and S. Krishnan, "Trends in audio signal feature extraction methods," *Appl. Acoust.*, vol. 158, Jan. 2020, Art. no. 107020, doi: 10.1016/j.apacoust.2019.107020.

[70] A. Chaudhari, P. Davda, M. Dand, and S. Dholay, "Profanity detection and removal in videos using machine learning," in *Proc. 6th Int. Conf. Inventive Comput. Technol. (ICICT)*, Jan. 2021, pp. 572–576, doi: 10.1109/ICICT50816.2021.9358624.

[71] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "MobileNets: Efficient convolutional neural networks for mobile vision applications," 2017, *arXiv:1704.04861*.

[72] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2818–2826.

[73] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 2, pp. 84–90, Jun. 2012.

[74] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

ABDULAZIZ SALEH BA WAZIR received the B.Eng. degree (Hons.) from the Faculty of Electrical and Electronic Engineering, PETRONAS University of Technology, Malaysia, and the M.Eng. degree in mechatronics from the University of Malaya, Malaysia. He is currently a Researcher with the Faculty of Engineering, Multimedia University, Malaysia. He is also working on a project of inappropriate audible and visual content detection. He works closely on researches collaborations Telekom Malaysia, Hypp TV for shows and films automated censorship system using on deep learning. His main research interests include the area of deep learning, artificial intelligence, multi-media, robotics, and signal processing.

HEZERUL ABDUL KARIM (Senior Member, IEEE) received the B.Eng. degree in electronics with communications from the University of Wales Swansea, U.K., in 1998, the M.Eng. degree in science from Multimedia University, Malaysia, in 2003, and the Ph.D. degree from the University of Surrey, U.K., in 2008. He is currently a Professor with the Faculty of Engineering, Multimedia University. His research interests include telemetry, error resilience and multiple description video coding for 2D/3D image/video coding and transmission, and content-based image/video recognition. He is currently serving as the Vice Chair/the Chair-Elect for the IEEE Signal Processing Society Malaysia Section.

HOR SUI LYN (Student Member, IEEE) received the B.Eng. degree in electronics engineering majoring in computer from Multimedia University, Malaysia, in July 2020, where she is currently pursuing the master's degree with the Faculty of Engineering. Her research interests include visual sensitive detection using deep and active learning. Her domains of interests include the areas of deep learning, image processing, and computer vision.

MOHAMMAD FAIZAL AHMAD FAUZI (Senior Member, IEEE) received the B.Eng. degree in electrical and electronic engineering from Imperial College London, London, U.K., in 1999, and the Ph.D. degree in electronics and computer science from the University of Southampton, Southampton, U.K., in 2004. From May 2013 to June 2014, he was attached to the Clinical Image Analysis Laboratory, The Ohio State University, USA, where he works on digital histopathology, especially on cancer and diseases analysis. He is currently a Professor with the Faculty of Engineering, MMU. He has published more than 100 journal and conference papers. His main research interests include the area of signal and image processing, pattern recognition, computer vision, and medical imaging. He is currently an Executive Committee for IEEE Region 10 (Asia Pacific).

SARINA MANSOR received the B.Eng. degree (Hons.) in electronic and electrical from University College London, in 1998, the M.Eng.Sc. degree from Multimedia University, Malaysia, in 2002, and the D.Phil. degree in engineering science from the University of Oxford, U.K., in 2009. She is currently a Senior Lecturer with the Faculty of Engineering, Multimedia University. She is also the Program Coordinator of B.Eng. (electronics) degree majoring in computer. Her research interests include signal and image analysis, medical imaging, computer vision, machine learning, and the Internet of Things.

MOHD HARIS LYE (Member, IEEE) received the B.Eng. degree in electrical and electronics from the University of Science Malaysia (USM), in 1996, and the M.S. degree in information technology from Multimedia University (MMU), Malaysia, in 2005. He is currently a Lecturer with the Faculty of Engineering, Multimedia University. His research interests include deep learning, computer vision, and artificial intelligence. His current research interests include deep learning and active learning approach for egocentric vision and indoor surveillance for smart homes.