INFOTECH
Journal of Applied Sciences, Informatics, Computing and Knowledge Society

Volume 4, No. 2, July/December, 2023

Surveillance Performance Analysis of Vision Tasks in Common Device Applications

Alexander Oliver Chag[1]

[1] University of Cape Town – Sudáfrica, alexanderoliverchag@hotmail.com , https://orcid.org/0009-0004-2730-6728

ABSTRACT

This study delves into the effectiveness of keyword spotting and handgun detection tasks, widely employed for optimizing device control and surveillance systems. While deep learning approaches dominate these tasks, their performance is predominantly assessed in datasets of exceptional quality. This research aims to scrutinize the efficacy of these tools when applied to information captured by commonplace devices, such as commercial surveillance systems with standard resolution cameras or smartphone microphones. To achieve this, we propose the creation of an audio dataset comprising speech commands recorded from mobile devices and various users. The audio analysis involves an evaluation and comparison of state-of-the-art keyword spotting techniques against our own model, which surpasses baseline and reference approaches, yielding an impressive 83% accuracy. For handgun detection, we fine-tune YOLOv5 to tailor the model for accurate handgun detection in both images and videos. The model is rigorously tested on a novel dataset featuring labeled images from commercial security cameras. This comprehensive evaluation ensures a robust assessment of the model's adaptability and performance in real-world scenarios, providing valuable insights for the development and deployment of surveillance applications on common devices.

Keywords: *Artificial Intelligence; Renewable Energy; Cybersecurity; Sustainable Agriculture*

Received:
03 October, 2023

Accepted for publication:
17 November, 2023

INFOTECH
Journal of Applied Sciences, Informatics, Computing and Knowledge Society
Volume 4, No. 2, July/December, 2023

Análisis del Rendimiento de Vigilancia de Tareas Visuales en Aplicaciones de Dispositivos Comunes

RESUMEN

Este estudio profundiza en la efectividad de las tareas de detección de palabras clave y detección de pistolas, ampliamente empleadas para optimizar el control de dispositivos y los sistemas de vigilancia. Mientras que los enfoques de aprendizaje profundo dominan estas tareas, su rendimiento se evalúa predominantemente en conjuntos de datos de calidad excepcional. Esta investigación tiene como objetivo examinar la eficacia de estas herramientas cuando se aplican a información capturada por dispositivos comunes, como sistemas de vigilancia comerciales con cámaras de resolución estándar o micrófonos de teléfonos inteligentes. Para lograr esto, proponemos la creación de un conjunto de datos de audio que incluya comandos de voz grabados desde dispositivos móviles y diversos usuarios. El análisis de audio implica una evaluación y comparación de las técnicas de detección de palabras clave de última generación frente a nuestro propio modelo, que supera los enfoques de referencia, logrando una impresionante precisión del 83%. Para la detección de pistolas, ajustamos finamente YOLOv5 para adaptar el modelo a la precisa detección de pistolas en imágenes y videos. El modelo se prueba rigurosamente en un conjunto de datos novedoso que presenta imágenes etiquetadas de cámaras de seguridad comerciales. Esta evaluación integral garantiza una evaluación sólida de la adaptabilidad y el rendimiento del modelo en escenarios del mundo real, proporcionando ideas valiosas para el desarrollo e implementación de aplicaciones de vigilancia en dispositivos comunes.

Palabras clave: *Inteligencia Artificial; Energías Renovables; Ciberseguridad; Agricultura Sostenible*

INTRODUCTION

In the contemporary landscape, numerous surveillance systems incorporate multiple interconnected sensors to capture information from diverse modalities, such as video and audio. The intriguing aspect of current surveillance platforms lies in their smart capabilities, which leverage artificial intelligence algorithms to automate surveillance tasks. For example, a system equipped with a video camera should autonomously make decisions based on specific events of interest, such as triggering an alarm upon detecting a handgun. Similarly, microphones strategically placed in certain locations can be used to monitor specific commands or identify potentially dangerous phrases.

These smart capabilities rely on various computer vision and speech recognition strategies. Object detection, implemented in high-resolution cameras across city points, detects items like handguns, license plates, and vehicles. Meanwhile, keyword spotting is integrated into smart devices like Amazon Alexa, enabling the detection of specific events like glass breaking through multiple microphones. While these approaches demonstrate acceptable performance in specific cases, their effectiveness in scenarios involving information from commonly used devices remains uncertain. This includes audio captured by smartphones or videos from commercial security camera systems.

This paper aims to investigate and assess handgun detection and keyword spotting using information from sensors in commercial security systems and smartphones. The goal is to shed light on the actual performance of algorithms when applied to data from surveillance systems commonly found in various locations in Mexico and typical smartphones. Concerning handgun detection, the focus is on videos from security systems equipped with cameras and internet connectivity, configured in residences, stores, or restaurants.

Concerning keyword spotting, our focus lies in assessing the performance of recognizing voice commands from audio captured by mobile devices. The objective is to understand the feasibility of integrating handgun detection and keyword spotting into platforms that typically lack high-performance devices for video and audio capture. To the best of our knowledge, this marks the first instance where state-of-the-art algorithms are being studied, adapted, and evaluated within the context of Mexico. The primary contributions of this paper can be summarized as follows:

1. Constructing a dataset for handgun detection using information obtained from commercial cameras installed in residences or stores in Mexico.
2. Developing a dataset for keyword spotting, encompassing various voice commands in Spanish. These commands were recorded using diverse mobile devices and featured different voices, including both female and male voices within the age range of 18 to 50.
3. Proposing and evaluating a strategy for keyword spotting, specifically aimed at recognizing voice commands in audio recordings captured by smartphones.
4. Adapting and evaluating state-of-the-art algorithms for handgun detection and tracking in videos.

The remainder of this paper is structured as follows. In Section 2, we present a comprehensive review of related work in the realms of computer vision and speech recognition. Sections 3 and 4 delve into our proposed architecture for Keyword Spotting and the adapted handgun detection strategy, respectively. Within these sections, we introduce the proposed datasets and elaborate on the methodology employed in their construction. Section 5 provides details on the experiments conducted and the evaluation performed for this study. Finally, Section 6 outlines our key conclusions and highlights avenues for future research.

Related Work

Keyword Spotting

Keyword Spotting (KWS) is designed to identify pre-defined keywords or sets of keywords within a continuous audio stream. Wake-word detection, in particular, has become a pivotal application of KWS [3]. Various strategies have been employed to address this task, with deep learning methods taking precedence in recent years. Arik et al. [1] proposed a Convolutional Recurrent Neural Network (CRNN) approach for solving the Keyword Spotting problem.

This approach utilizes different CRNN architectures based on Long Short-Term Memories (LSTMs) [5] and Gated Recurrent Units (GRUs) [2]. These architectures apply convolutions over mel spectrograms computed from the audio data, achieving performances of up to 99.6% accuracy in a speech dataset.

Another approach, introduced by Zhang et al. [15], leverages a Deep Stepwise Separable Convolution Neural Network (DS-CNN) that achieved an accuracy of up to 95.4%. This result was obtained using the dataset described in [13], consisting of one-word audio data pronounced in English. This one-word format for audio data is also adopted in our work for training the proposed model.

Several authors have explored the use of Separable Convolutions for raw audio data. For instance, the work outlined in [8] utilized mel coefficients and incorporated a SincLayer proposed by Ravinelli et al. [10] for improved feature extraction of audio data. This model achieved up to 96.4% accuracy over the dataset described in [13].

In this study, we introduce a model designed to extract features from audio data through the computation of mel spectrograms for each audio segment. Additionally, our approach leverages the benefits of Deep Stepwise Separable Convolutions for audio classification. As demonstrated later in this paper, our proposed method exhibits superior performance compared to baseline models and reference methodologies.

Handgun Detection

The task of handgun detection can be viewed as a specialized case within the broader context of object detection. Various approaches have been explored to address the object detection problem, with a key focus on achieving optimal precision while minimizing memory requirements.

Numerous methods have been proposed in this domain, with deep learning-based approaches standing out as particularly noteworthy. For instance, ResNet [4] and ResNetV2 [14], which are Residual Networks (ResNets), were introduced in 2015 and have proven to be prominent models for object detection, sharing similarities with the challenges encountered in keyword spotting.

Table 1 illustrates the allocation of audio samples across the training, validation, and test sets within the generated dataset.

Training	Testing	Validation
928	318	135

ResNets play a crucial role in mitigating the vanishing gradient problem, particularly as networks grow deeper in terms of layer count.

More recently, alternative architectures have gained popularity, showcasing promising outcomes. Noteworthy examples include MobilenetV2 [11], YOLOV5 [6], and EfficientDet [12]. These architectures offer the advantage of achieving high performance with minimal computational costs in terms of time and memory, a significant improvement over earlier methods for the same task.

For instance, MobilenetV2 is specifically designed for execution on mobile devices, while YOLOV5 and EfficientDet excel in achieving real-time detection capabilities.

Presently, YOLOV5 stands as the state-of-the-art model for object detection. In this context, we will adapt and evaluate YOLOV5 for handgun detection. Experimental evaluations indicate that, by utilizing YOLOV5, we can develop models achieving a 65% accuracy in gun detection within security camera images. Moreover, the model demonstrates acceptable performance in detecting potential fire alerts within videos.

Keyword Spotting

The neural network architecture employed for audio command recognition draws inspiration from prior studies, as detailed in [8, 15, 10]. This paper introduces a novel architecture designed specifically for recognizing audio commands in the Spanish language, leveraging a compact audio dataset for model training. Subsequent sections detail the methodology for dataset creation, data preprocessing steps, and the architecture tailored for this task.

Dataset

The proposed neural network model for Keyword Spotting (KWS) underwent evaluation using two distinct datasets. The first dataset originates from [13], while the second was meticulously generated for this study by manually recording audio samples from diverse users and smartphones. The distribution of audio files within the generated dataset, categorized for training, validation, and testing, is outlined in Table 1.

Table 1 illustrates that the KWS dataset consists of 1381 audio files, each featuring a single audio command with an average duration of 3 seconds per file. These audio files were recorded using mobile devices and subsequently converted into wav files. For this study, voices of both females and males

within the age range of 18 to 50 were employed to compose the dataset. The predefined audio commands recorded in this dataset include: 'secunet,' 'encender,' 'apagar,' and 'tranquilizate.'

The word 'secunet' functions as the wake-word, anticipating the execution of either the 'encender' or 'apagar' command, thereby activating or deactivating the alarm, respectively. Notably, the term 'tranquilizate' serves a dual purpose as both a wake-word and a command, triggering the alarm in a discreet mode using the secret phrase 'tranquilizate tranquilizate' (akin to "take it easy").

Additionally, a negative class labeled 'unknown' was curated using audio files captured by mobile devices, capturing ambient sounds like traffic noise and conversations.

Data Preprocessing

This study aims to identify audio commands from raw wave audio data. However, it is essential to preprocess the input data for training and testing the proposed neural network architecture. Specifically, the input data underwent standardization using the Python module librosa [7]. The length of each audio was reduced to a sequence comprising 30,000 data points, representing an average duration of 1 second per audio file. For this purpose, the start and end points for standardizing the audio sequence were randomly selected, creating diverse audio sequences for a single command.

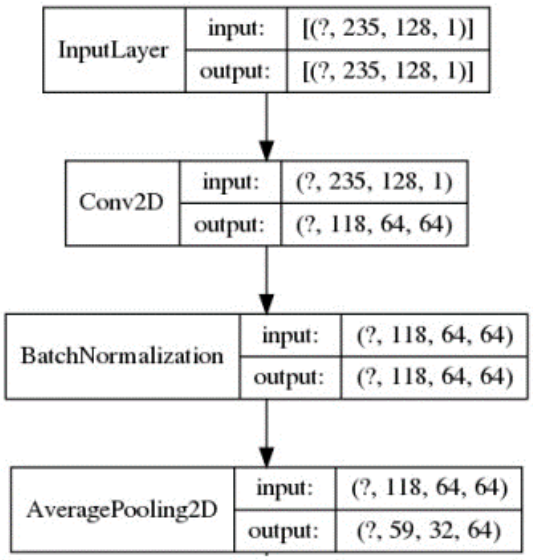

Figure 1 illustrates the initial convolutional layer in the proposed model

Model Architecture

The model architecture employed in this study incorporates certain concepts from [8]. Specifically, mel spectrograms are computed for each audio file, serving as the input data for the neural network model. The model comprises four primary components.

The first component involves a two-dimensional convolutional layer, utilizing the activation function log(|x| + 1) [8]. The structure of this initial convolutional layer is detailed in Figure 1.

The second component encompasses a block of intermediate layers tasked with encoding the highly discriminative features of the input data. Following this process, Global Average Pooling is applied to retain the most salient activation values.

Finally, the fourth component consists of dense layers employed for audio classification. The following paragraphs provide a detailed description of each of these main components.

— **First Convolutional Layer:** This layer features a two-dimensional convolution applied to the mel spectrograms, utilizing the activation function f(x) = log(|x| + 1) as proposed in [8]. The structure of this initial convolutional layer is illustrated in Figure 1.

— **Intermediate Layers:** Based on the intermediate layers of the Deep Stepwise Convolutional Neural Network - Short (DS-CNN-S) described in [15], each layer comprises a two-dimensional separable convolution with the Relu activation function, followed by batch normalization and Average Pooling with a window size of 2 × 2. At the layer's end, a Dropout of probability 0.6 is applied to prevent overfitting and increase the number of training epochs.

— **Global Average Pooling**: Applied at the conclusion of the four intermediate layers, Global Average Pooling is employed to reduce dimensionality.

— **Dense Layers**: The dense layers serve to classify the input data into the categories outlined in 4.2.1.

The proposed model features a total of 56,133 parameters, with 55,493 being trainable and 640 non-trainable parameters corresponding to those whose loss function cannot be optimized using the training data.

Handgun Detection

As previously mentioned, the approach to handgun detection involves adapting YOLOV5 [6]. This section introduces the dataset created for this task, the modified model, and the parameters applied in training the handgun detection model.

Dataset

This study focuses on handgun detection in images captured by real-life cameras in Mexico, aiming to evaluate deep learning models, particularly YOLOV5 [6]. For this evaluation, a new dataset was constructed using information extracted from actual surveillance systems. Images and videos depicting real-life situations in Mexico, predominantly involving robberies, were manually gathered from YouTube. The dataset comprises 107 positive images (with a gun) and 129 negative images (without a gun), serving as a test set for the model in Section 5.2.

To train the handgun detection model, transfer learning was employed, utilizing databases from [9]. These datasets follow the YOLO format, wherein each image corresponds to an XML file containing box coordinates encapsulating all guns in the image. For our model, various weapon types, including knives, were included during training, with specific evaluation focused on handguns. This decision was based on the hypothesis that diverse scenarios involving various weapons might share contextual visual information beneficial to the model.

The [9] dataset encompasses various types of short guns and knives, both included in the final model. Manual annotations during training were used to extract images intended for recognition, as well as parts of images devoid of guns or knives. This approach aimed to minimize false positives detected by the model. Notably, the images in this dataset are not sourced from security cameras but are utilized for transfer learning and handgun detection within the realm of surveillance systems.

4.2 Modified Model As mentioned earlier, the adapted model for handgun detection is YOLOv5 [6], a fully convolutional model. Unlike other object detection models, YOLOv5 simultaneously predicts the class and position of the object, a process typically separated in works such as RCNNs. This simultaneous learning of detection and tracking steps provides YOLOv5 with an advantage, especially in object detection for low-resolution images.

It's important to note that YOLOv5 comes in four different sizes, distinguished by the number of convolutional layers used. For this research, the medium-sized model was employed. Specifically, this model underwent training with the dataset outlined in Section 4.1.

The data from Section 4.1 was divided into 80% for training and 20% for testing. The model underwent training for 300 epochs.

Table 2. Proposed Model Confusion Matrix

Confusion matrix					
	amb	ap	enc	tran	sec
amb	35	3	1	0	0
ap	0	51	3	3	3
enc	2	1	59	2	6
tran	1	3	2	57	7
sec	0	2	8	7	62

Table 3. Proposed Model Metrics by Class

Metrics by class				
	accuracy	precision	recall	f1-score
amb	0.978	0.90	0.92	0.91
ap	0.9334	0.85	0.85	0.85
enc	0.9214	0.84	0.81	0.83
tran	0.9214	0.81	0.83	0.82
sec	0.8962	0.78	0.79	0.79

RESULTS

Keyword Spotting Results

The Keyword Spotting model proposed in this work underwent evaluation using the collected dataset described in Section 3.1. To assess the effectiveness of the proposed model, we employed metrics such as accuracy, F1 score, precision, and recall. A comparative analysis was conducted against a DS-CNN model proposed by [15], an LSTM-based convolutional model from [1], a model utilizing two simple dimensional convolutions, and a model adapted from [8], which replaced mel spectrograms with mel coefficients, as used in this work. The confusion matrices for each training are also presented for comparison among the various models evaluated in this work. In the following subsections, we detail

each of the models used for comparison. Evaluations were performed with different datasets described earlier, and the performance metrics presented in the preceding paragraph were utilized.

Evaluation of the Proposed Model

This experiment aims to observe the performance of the proposed model on a dataset comprising real-life instances collected by users and their smartphones. The model underwent training for 5000 epochs with a batch size of 100 samples. After training, the confusion matrix obtained by evaluating the model on the test set is presented in the tables below. The x-axis of the confusion matrix denotes the predictions made by the model, while the y-axis corresponds to the labels of the respective audio files. The order of appearance of the audio commands for this dataset is 'ambiente,' 'apagar,' 'encender,' 'tranquilizate,' 'secunet.'

Table 4. Proposed Model Average Metrics

Average metrics			
Accuracy	F1 score	recall	precision
0.8301	0.8388	0.83787	0.8400

Table 5. Simple Convolution Layers Model Confusion Matrix

Confusion matrix					
	amb	ap	enc	tran	sec
amb	38	0	1	0	0
ap	3	47	4	4	2
enc	1	0	61	4	4
tran	1	6	10	46	7
sec	5	5	10	7	52

These classes are denoted in the confusion matrix with the abbreviations 'amb', 'ap', 'enc', 'tran,' and 'sec,' respectively. The results, as depicted in Table 2, demonstrate the exceptional performance of the proposed model. In this experiment, the overall accuracy is 83.01% on the test set and 88.14% on the validation set. Accuracy and F1 scores by class are detailed in Table 3, while average metrics are summarized in Table 4.

Table 6. Simple Convolutional Model Metrics by Class

Metrics by class				
	accuracy	precision	recall	f1-score
amb	0.9654	0.97	0.79	0.87
ap	0.9245	0.78	0.81	0.80
enc	0.8931	0.87	0.71	0.78
tran	0.8774	0.66	0.75	0.70
sec	0.8742	0.66	0.80	0.72

Table 7. Simple Convolution Model Average Metrics

Average metrics			
Accuracy	f1 score	Recall	Precision
0.76729	0.77534	0.78888	0.773

Table 8. CLSTMNN Model Confusion Matrix

Confusion matrix					
	amb	ap	enc	tran	sec
amb	36	1	0	2	0
ap	4	47	0	1	8
enc	2	1	28	21	18
tran	0	3	8	36	23
sec	4	8	12	7	48

Evaluation of the Simple Convolutional Model

The objective of this experiment is to assess the performance of the specific architecture of the Simple Convolutional Model. This model underwent training for 5000 epochs with a batch size of 100. As outlined in Table 5, the model achieved an overall accuracy of 76.72% on the test set, with a slightly higher accuracy of 76.73% on the validation set. Metrics by class are detailed in Table 6, revealing that the proposed model outperforms the Simple Convolutional Model across various metrics. Average metrics for this model are presented in Table 7, showing lower performance compared to the results obtained by our proposal in Table 4.

Evaluation of Convolutional Long Short-Term Memory Neural Network (CLSTMNN)

This experiment aims to compare the performance of the Convolutional Long Short-Term Memory Neural Network (CLSTMNN). The results are obtained from a model that utilizes Long Short-Term

Memory neural networks after applying a convolutional layer to the mel spectrograms. The confusion matrix for this model is provided in Table 8.

Metrics by class are outlined in Table 9, and average metrics are summarized in Table 10. These tables illustrate that the model achieved an accuracy of 61.32% on both the test and validation sets, which is notably lower than the performance of the previously evaluated models at this stage.

Table 9. CLSTMNN Model Metrics by Class

Metrics by class				
	accuracy	precision	recall	f1-score
amb	0.9591	0.92	0.78	0.85
ap	0.9182	0.78	0.78	0.78
enc	0.805	0.40	0.58	0.47
tran	0.7956	0.51	0.54	0.53
sec	0.7484	0.61	0.49	0.55

Table 10. CLSTMNN Model Average Metrics

Average metrics			
Accuracy	f1 score	Recall	Precision
0.6132	0.6351	0.6456	0.6362

Evaluation of the Deep Stepwise Separable Convolutions Model

The objective of this experiment is to assess the performance of the Deep Stepwise Separable Convolutions model. To achieve this, we trained a model based on the architecture proposed in [15], which employs separable convolutions. This model underwent training for 4500 epochs with a batch size of 100. The confusion matrix resulting from this model, with the specific settings previously outlined, is presented in Table 11.

Metrics by class for this model are detailed in Table 12, while average metrics are summarized in Table 13. These results indicate that the Deep Stepwise Separable Convolutions model emerges as our closest competitor in terms of performance. We hypothesize that the combination of convolutions and recurrent information within the network plays a crucial role in capturing discriminative information.

Table 11. Deep Stepwise Separable Convolutions Model Confusion Matrix

Confusion matrix					
	amb	ap	enc	tran	sec
amb	38	1	0	0	0
ap	0	49	1	3	7
enc	4	3	56	2	5
tran	1	4	3	52	10
sec	3	4	4	3	65

Table 12. Deep Stepwise Separable Convolutions Model Metrics by Class

Metrics by class				
	accuracy	precision	recall	f1-score
amb	0.9717	0.97	0.83	0.89
ap	0.9277	0.82	0.80	0.81
enc	0.9308	0.80	0.88	0.84
tran	0.9182	0.74	0.87	0.80
sec	0.8868	0.82	0.75	0.78

Table 13. Deep Stepwise Separable Convolutions Model Average Metrics

Average metrics			
Accuracy	f1 score	Recall	Precision
0.8176	0.8176	0.8333	0.82363

Model Utilizing Mel Coefficients

In this concluding experiment for Keyword Spotting, our focus is on assessing the overall performance of traditional features commonly employed in Speech Recognition for this task. Thus, in this section, we present the results obtained from the model utilizing mel coefficients instead of the mel spectrogram of the audio, a concept akin to that presented in [8]. This model underwent training for 4500 epochs with a batch size of 100, and the resulting confusion matrix of predictions is presented in Table 14.

The overall accuracy obtained from Table 14 is 77.24%. Metrics by class are detailed in Table 15, while average metrics are summarized in Table 16.

Comparing the results presented in the tables above, we can evaluate the varied performance of the models as shown in Table 17. This table provides different accuracy values obtained by the various models over the test set. From these results, there is empirical evidence suggesting that the proposed

model achieves the best results for the Keyword Spotting task within the studied domain, specifically for identifying commands in audios collected by users utilizing their smartphones.

Table 14. Model Using Mel Coefficients Confusion Matrix

Confusion matrix					
	amb	ap	enc	tran	sec
amb	37	0	1	0	1
ap	0	48	3	4	5
enc	0	1	63	2	4
tran	1	3	4	52	10
sec	4	6	6	7	56

Table 15. Model Using Mel Coefficients Metrics by Class

Metrics by class				
	accuracy	precision	recall	f1-score
amb	0.978	0.95	0.88	0.91
ap	0.9308	0.80	0.83	0.81
enc	0.934	0.90	0.82	0.86
tran	0.9025	0.90	0.82	0.77
sec	0.8648	0.71	0.74	0.72

Results on Vision Tasks

Image Classification Evaluation

The objective of this experiment is to assess the performance of the adapted YOLOv5 architecture in predicting the presence of a handgun in images captured by commercial security cameras. For this purpose, we employed the dataset described in Section 4.1 within an image classification framework. To gauge the algorithm's performance, we established a framework for evaluating gun detection performance. Hence, we assess this adapted YOLOv5 by utilizing the final score in the prediction as a probability threshold to determine the ultimate class. Tables 18 and 19 illustrate that reducing the probability threshold enhances accuracy and F1-scores, albeit at the cost of increased false positive predictions, while simultaneously reducing false negatives.

Table 16. Model Using Mel Coefficients Average Metrics

Average metrics			
Accuracy	f1 score	Recall	Precision
0.80503	0.81544	0.8200	0.8127

Table 17. Comparison Between Proposed Models for Keyword Spotting

Comparison between models	
Model	Accuracy
Model Proposed	**0.8301**
Simple convolutions	0.76729
DSCNN-S	0.8176
Mel coefficients	0.80503
CLSTM	0.6132

Table 18. F1-Score for Handgun Class and Accuracy of YOLOv5 Detecting Images with a Handgun Using Different Probability Thresholds. The Dash Denotes That There Were No Positive Predictions

p_t/Metric	F1-score	Accuracy
0.8	-	54.47%
0.6	7.21%	56.17%
0.4	15.13%	57.02%
0.2	24.42%	57.87%
0.1	39.74%	61.28%
0.05	49.71%	62.13%
0.01	64.66%	65.11%

Table 19. Confusion Matrices of Detection of Images with a Handgun. These Are the Results Using Different Probability Thresholds with YOLOv5

p_t		Positive	Negative
0.8	Pos pred	0	0
	Neg pred	107	128
0.6	Pos pred	4	0
	Neg pred	103	128
0.4	Pos pred	9	3
	Neg pred	98	125
0.2	Pos pred	16	8
	Neg pred	91	120
0.1	Pos pred	30	14
	Neg pred	77	114
0.05	Pos pred	44	26
	Neg pred	63	102
0.01	Pos pred	75	50
	Neg pred	32	78

Table 20. Results of Object Detection in the Test Set Using YOLOv5. It Shows the Number of Predicted Boxes, Number of Real Boxes, and Number of Intersected Boxes Using Different Probability Thresholds.

p_t/Metric	Predicted	Real	Intersected
0.8	0	108	0
0.6	4	108	4
0.4	9	108	12
0.2	15	108	24
0.1	25	108	44
0.05	36	108	70
0.01	74	108	125

This is an important parameter to determine when the security system is implemented, since a false positive and false negative might have different consequences, depending on the application. The specific way we use the probability threshold pt for each result is as follows: the test images will be label as 0 and 1, where 1 means that the image has a gun and 0 that the image does not have a gun. With these labels, we compute the Confusion matrix, Accuracy and F1-score.

Table 21. Results from the Algorithm in 5 Videos and All the Different Confidence Thresholds Used. It Shows the Results of Accuracy, F1-Score, Intersected Boxes, and Detected Boxes. The Dash Denotes That There Were No Positive Predictions

Video	p_t	Acc.	F1	Intersec.	Detec.
Video 1	0.8	34.44%	-	0	0
	0.6	34.44%	-	0	0
	0.4	41.11%	18.46%	6	6
	0.2	55.55%	68.25%	16	67
	0.1	61.11%	75.52%	26	84
	0.05	63.33%	77.55%	34	88
	0.01	65.55%	79.19%	70	90
Video 2	0.8	5.48%	-	0	0
	0.6	5.48%	-	0	0
	0.4	6.85%	2.86%	1	1
	0.2	9.59%	8.33%	2	3
	0.1	17.81%	25%	6	11
	0.05	41.10%	55.67%	13	28
	0.01	84.93%	91.85%	34	66
Video 3	0.8	64.77%	-	0	0
	0.6	64.77%	-	0	0
	0.4	65.91%	6.25%	0	1
	0.2	69.32%	40%	0	14
	0.1	65.91%	46.42%	1	25
	0.05	69.32%	60.87%	1	38
	0.01	50%	56.86%	2	71
Video 4	0.8	15.91%	-	0	0
	0.6	15.91%	-	0	0
	0.4	18.18%	5.26%	2	2
	0.2	27.27%	37.25%	10	28
	0.1	44.32%	59.50%	20	47
	0.05	79.54%	88.46%	26	82
	0.01	84.09%	91.36%	48	88
Video 5	0.8	84.27%	-	0	0
	0.6	84.27%	-	0	0
	0.4	88.76%	44.44%	4	4
	0.2	86.52%	40%	4	6
	0.1	82.02%	33.33%	5	10
	0.05	67.42%	21.62%	8	23
	0.01	49.44%	21.05%	12	43

Object Detection Evaluation

The primary objective of this model is to identify the location of the gun within an image. Consequently, specific metrics become crucial for assessment. As outlined in Section 4.1, we possess annotations indicating the "box" containing the gun in annotated images. For testing, our focus is solely on images from the test set that feature a gun. The following metrics will be computed for these images:

—**Predicted boxes:** The total number of predicted boxes across all images in the test set.

—**Real boxes:** The total number of boxes in the annotations.

—**Intersected boxes:** The total number of predicted boxes that intersect with at least one real box.

Similar to Section 5.2.1, we only consider boxes with a probability greater than a given threshold pt. The evaluation is conducted for pt $\in \{0.8, 0.6, 0.4, 0.2, 0.1, 0.05, 0.01\}$. The results, displayed in Table 20, reveal a limited number of intersections between boxes. Higher probability thresholds yield fewer detected handguns, but with increased accuracy. As the prediction threshold decreases, the number of intersected boxes improves, albeit at the expense of detection accuracy. The insights from Section 5.2.1 can guide the selection of the optimal probability threshold for specific applications.

Video Detection Evaluation

This experiment aims to assess the effectiveness of the proposed handgun detection strategy in videos. The algorithm underwent evaluation using five distinct videos from security cameras within the collected dataset, each lasting approximately 30 seconds. For this assessment, we sampled 3 frames per second for detection. The results of these experiments are detailed in Table 21.

As observed, a confidence threshold of 0.4 appears to yield optimal performance in videos. This threshold minimizes false positives, demonstrating accuracy in its predictions. In the context of videos, false positives carry greater significance, given the analysis of 3 frames per second. Consequently, false positives are more costly than false negatives.

CONCLUSIONS

This paper contributes by establishing a dataset for Keyword Spotting in Spanish and Handgun detection. We evaluated adapted versions of state-of-the-art methods, showcasing their performance

across diverse scenarios. These tasks are designed to be applicable in real-life situations, particularly within commercial surveillance systems.

The objective is to conduct a comprehensive study demonstrating the efficacy of cutting-edge methodologies, capable of complementing security platforms in commercial systems. This is crucial, considering that evaluations of object detection and keyword spotting often rely on data collected by devices inaccessible to many individuals, particularly in countries like Mexico.

The experimental outcomes presented in this paper indicate that the proposed architecture for Keyword Spotting serves as a viable alternative, achieving accuracy levels surpassing 83%. Concerning handgun detection, the adapted methods exhibit potential, provided we identify the specific threshold conducive to the domain.

BIBLIOGRAPHICAL REFERENCES

1. Arik, S., Kliegl, M., Child, R., Hestness, J., Gibiansky, A., Fougner, C., Prenger, R., Coates, A. (2020). Convolutional recurrent neural networks for small-footprint keyword spotting. US Patent 10,540,961.
2. Cho, K., Van Merrienboer, B., Gulcehre, C., ¨ Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y. (2014). Learning phrase representations using rnn encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078.
3. Coucke, A., Chlieh, M., Gisselbrecht, T., Leroy, D., Poumeyrol, M., Lavril, T. (2019). Efficient keyword spotting using dilated convolutions and gating. ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, pp. 6351–6355. Computación y Sistemas, Vol. 25, No. 2, 2021, pp. 317–328 doi: 10.13053/CyS-25-2-3867 Deep Learning for Language and Vision Tasks in Surveillance Applications 327 ISSN 2007-9737
4. He, K., Zhang, X., Ren, S., Sun, J. (2016). Deep residual learning for image recognition. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
5. Hochreiter, S., Schmidhuber, J. (1997). Long Short-Term Memory. Neural Computation, Vol. 9, No. 8, pp. 1735–1780.

6. Jocher, G., Stoken, A., Borovec, J., NanoCode012, ChristopherSTAN, Changyu, L., Laughing, tkianai, yxNONG, Hogan, A., lorenzomammana, AlexWang1900, Chaurasia, A., Diaconu, L., Marc, wanghaoyang0106, ml5ah, Doug, Durgesh, Ingham, F., Frederik, Guilhen, Colmagro, A., Ye, H., Jacobsolawetz, Poznanski, J., Fang, J., Kim, J., Doan, K., Yu, L. (2021). ultralytics/yolov5: v4.0 - nn.SiLU() activations, Weights & Biases logging, PyTorch Hub integration.

7. McFee, B., Raffel, C., Liang, D., Ellis, D. P., McVicar, M., Battenberg, E., Nieto, O. (2015). Librosa: Audio and music signal analysis in python. Proceedings of the 14th python in science conference, volume 8, pp. 18–25.

8. Mittermaier, S., K ¨urzinger, L., Waschneck, B., Rigoll, G. (2020). Small-footprint keyword spotting on raw audio data with sinc-convolutions. ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, pp. 7454–7458.

9. Olmos, R., Tabik, S., Herrera, F. (2018). Automatic handgun detection alarm in videos using deep learning. Neurocomputing, Vol. 275, pp. 66–72.

10. Ravanelli, M., Bengio, Y. (2018). Speaker recognition from raw waveform with sincnet. 2018 IEEE Spoken Language Technology Workshop (SLT), IEEE, pp. 1021–1028.

11. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.-C. (2018). Mobilenetv2: Inverted residuals and linear bottlenecks. Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 4510–4520.

12. Tan, M., Pang, R., Le, Q. V. (2020). Efficientdet: Scalable and efficient object detection. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10781–10790.

13. Warden, P. (2018). Speech commands: A dataset for limited-vocabulary speech recognition. arXiv preprint arXiv:1804.03209.

14. Xie, S., Girshick, R., Dollar, P., Tu, Z., He, K. (2017). Aggregated residual transformations for deep neural networks. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

15. Zhang, Y., Suda, N., Lai, L., Chandra, V. (2017). Hello edge: Keyword spotting on microcontrollers. arXiv preprint arXiv:1711.07128.