AN IMPROVISED ENSEMBLE CNN ALGORITHM FOR DETECTING VIDEO STREAM IN MULTIMEDIA

C. Kiran Kumar[1], S. Chandra Sekaran[2], R. Gayathri[3] and S. Ramasamy[4]

[1]*Data Science, Codecraft Technologies, Bangalore, India*
[2]*Department of Computer Science and Engineering, PSV College of Engineering and Technology, India*
[3]*Department of Electronics and Communication Engineering, Rajalakshmi Engineering College, India*
[4]*Department of Computer Science and Engineering, Hindusthan Institute of Technology, India*

Abstract

The only criteria that are used to evaluate the various neural network-based object identification models that are currently in use are the inference times and accuracy levels. The issue is that in order to put these new classes and situations to use in smart cities, we need to train on them in real time. We were not successful in locating any research or comparisons that were centered on the length of time necessary to train these models. As a direct consequence of this, the initial reaction times of these object identification models will consistently be quite slow (maybe in days). As a consequence of this, we believe that models that put an emphasis on the speed of training rather than accuracy alone are in significant demand. Users are able to gather photos for use in training in the present by utilizing concept names in online data collection toolkits; however, these images are iconic and do not have bounding boundaries. Under these conditions, the implementation of semi-supervised or unsupervised models in a variety of smart city applications might be able to contribute to an improvement in the precision of data derived from IoMT. In this study, we categorize the video clips into their appropriate classes using an improved ensemble classification model.

Keywords:

CNN, Ensemble, Video Stream, IoT

1. INTRODUCTION

All of the previously made projections are currently being reexamined because of the meteoric rise in the number of internet-connected physical items. Even though the projection of 50 billion devices by 2020 will be widely discredited in the future, the next 10 billion Internet of Things devices coming online in the following year will result in trillions of connected things [1].

In addition, several recent studies [3, 4] and [5] offer unequivocal evidence of a growth in the proportion of Internet traffic that is comprised of multimedia. Sensors in smart cities are also producing a significant amount of multimedia data. This is happening as a direct result of the recent surge in the volume of multimedia traffic that is carried over the Internet. Unstructured (multimedia) occurrences include things such as traffic delays, automobile accidents, changes in the weather, problems with parking, security risks, persons being followed, and other such things.

As things stand, the Internet of Things was created to support the infrastructure of smart cities, which, in turn, improves the quality of life for the citizens who live in those cities. Researchers are primarily concerned with figuring out how to interpret scalar data in the context of smart cities. Examples of this are smart energy events that include readings from temperature sensors or energy sensors. An additional example of a scalar (structured) event is the data that is lost from RFID tags in the event that a packet is lost.

Traditional IoT worked just fine for us until we realized that smart cities generate massive amounts of multimedia data (i.e., images, videos, and audio) in addition to scalar data. This realization compelled us to switch to multimedia-based IoT (IoMT), which stands for the Internet of Multimedia Things [6]. Because integrating multimedia into the Internet of Things is still a relatively new concept, it has not yet been fully standardized, and research into its numerous potential applications—in fields such as traffic management, security, monitoring activities, terrorist attacks, natural disasters, and many more—needs to be carried out in depth. IoT middleware, which is responsible for supplying common services to apps and simplifying the development process, is still in its infant stages. This is true even for scalar events.

In this survey, we use the context of multimodal event processing in smart cities, which includes object detection, to highlight the research gaps that exist between this field and the current state of the art in IoT-based technologies. These gaps are highlighted using object detection as an example. In conclusion, we discuss the limitations of existing object identification models in the context of the implementation of smart cities, as well as the incapability of existing datasets to cover all varieties of smart cities. Additionally, we offer some possible solutions to the problems raised by these limitations.

Because the Internet of Things (IoT) industry has reached a mature stage, the focus of many studies [9] has shifted to concentrate on this sector. Event-based middleware and big data computing are utilized by an extremely small percentage of these organizations [10]. These studies primarily concentrate their attention on technologies that do not support or even explore the possibility of IoMT. Some examples of these technologies are those that do not incorporate multimedia components. The authors introduce IoMT and explore its goals and difficulties in a review [2], but they do not offer any real proposals for how to address the issue of IoMT application to multimedia event processing. This is because the authors believe that there is no clear solution to this problem.

Studies [2] that concentrate on the recognition of multimedia events tend to ignore the benefits of the internet of things. Numerous articles [11] have discussed the development of deep learning-based object recognition models with and without datasets, but these articles have neglected to evaluate how well these models perform in the context of smart city applications (in terms of accuracy, testing time, training time, number of classes, size, etc.).

2. RELATED WORKS

Image recognition [2] has become one of the conventional spheres of smart cities in order to recognize multimedia events when a camera detects a certain object. Alarms are sent in real time via text or image communications to the nearest city officials in order to notify them of the situation. Congestion, accidents, shifts in the weather, terrorist attacks, parking troubles, security threats, and other events of a similar nature are only some of the many conditions that could potentially result in multimedia occurrences.

This type of question can be answered with the help of a camera that has been installed at the bus stop. The camera can record multimedia events that detail the current bus schedule. People would have a much easier time monitoring the situation if we had something like Public Transport Management to answer questions of this nature. This would make it possible for more people to stay informed. In a similar vein, if a user wished to subscribe to parking-lot-related events such as Is a parking slot empty? which the present systems that are based on public transit are unable to answer, then the user would want a system that is referred to as Car Parking Management.

Again, users who wish to sign up for additional services, such as those that demand a taxi or pedestrian membership, won't be able to do so if the system doesn't understand how to adapt to their needs and learn from their actions. Additionally, in the decentralized environment of smart cities, millions of known and unknown classes may be added to user subscriptions. Because none of the real-time apps are going to require exactly those generic classes, and because the object detection datasets that are now available only apply to generic classes, none of the real-time applications are going to require them.

As a direct consequence of this, there is an increasing trend toward the creation of an entirely new dataset for each and every original smart city use case. Combining image recognition technology with machine learning models is becoming increasingly important as the focus shifts toward the goal of allowing high-performance systems for the identification of events in smart cities. However, IoT-based solutions do not make the most of these improvements and instead focus exclusively on text-based actions. This is despite the fact that these developments are extremely beneficial.

When it comes to managing unstructured events, however, solutions that are based on middleware are still in their infancy, despite the fact that they are effective at abstracting domain-specific applications and distributed platforms. Research on multimedia event processing in smart cities, with an emphasis on event-based middleware solutions, is required because all of these issues, as well as the challenges and potential answers, make it necessary to do this research.

In recent years, there has been a proliferation of reviews covering a wide range of topics related to the Internet of Things. Some examples of these reviews include IoT middleware, event processing, multimedia big data, multimedia processing with deep learning, object identification models, comparisons of image processing datasets, and many more. In order for us to understand the relevance of the work that is being given, we must first conduct an in-depth analysis of the reviews that have come before. The research that has been conducted on the subject of the Internet of Things has resulted in a large body of published material that is continually having its breadth and depth broadened, investigated, and summarized in a number of surveys. The recognition of events in multimedia is another domain in which the Internet of Things is occasionally, but not always, utilized. In the deep learning-based surveys, many image recognition techniques, such as object detection models, were included along with the dimensions of their technical implementation; however, performance in real-time applications was not included.

A survey on deep learning for the Internet of Things, big data, and streaming analytics [13] reviews several deep neural network-based designs and investigates IoT-based applications that can benefit from DL methods. The purpose of this survey is to gain a better understanding of how DL algorithms can be applied to Internet of Things (IoT) data. This document serves as a reference for matching the appropriate deep learning models with the appropriate Internet of Things applications. However, it does not explore the efficacy of deep learning models (such as CNN in object recognition models) in the several smart city scenarios in which they are utilized. These situations include:

A further survey, this one centered on the function of the Internet of Things (IoT) in smart cities across all industries, provides a thorough review of IoT middleware. It investigates the compatibility of existing middleware solutions with the requirements of the Internet of Things. However, it does not cover the processing of multimedia data that is produced in smart cities, despite its extensive discussion of middleware for dealing with data from the Internet of Things (IoT). Event-based middleware is one of the design methodologies utilized by existing middleware solutions for the Internet of Things, such as those used in smart cities, banking, medical services, telecommunications, entertainment, etc. A classification approach for event-based programming environments is presented in the paper [14], which can be found here.

This taxonomy categorizes event-based programming systems according to their service architecture and the event model that they support, thereby illuminating the traits that are common to all of these systems. A more detailed classification of event services according to organizational and interaction models as well as other functional and non-functional aspects is presented. By utilizing this hierarchical collection of attributes, it is possible to specify the relationships that exist between event systems, event services, and event models. Even though it does not include more contemporary event processing models, the taxonomy that has been presented is extremely extensive.

The research presented in this paper [12] offers a comprehensive analysis of the most recent developments in the field of event recognition, with a particular focus on deep learning architectures for multimedia. Multimedia event recognition takes the following forms: single photographs, personal photo collections, motion pictures, and audio recordings. In particular, it provides an in-depth review of systems that are based on deep learning for the purpose of event recognition. In addition, benchmark datasets are given a lot of importance in order to validate event identification methods. Therefore, it is the most relevant survey for image recognition; however, performance-based assessments of deep learning models are also required for real-time IoT applications.

3. DEEP LEARNING MODEL

It is difficult to obtain data for negative classes, a medical dataset may end up being unbalanced as a result. As a consequence of this, not all scores had access to the same amount of BBS motion data. On the other hand, if the model is trained with an imbalanced dataset, it is possible that it may be unduly sensitive to the class that is dominating, and it will perform poorly. In order to make the dataset more even, one method is to generate new data that is statistically equivalent to the one that was originally used.

In a manner analogous to that of downsampling, oversampling is a technique that may be utilized to improve BBS motion data. In study [11], an approach that was very similar to over-sampling was utilized in order to increase the overall number of observations and guarantee statistical equality between the two groups.

3.1 CLASSIFICATION MODEL

The BBS scoring technique was altered to incorporate the application of 1D-CNN and GRU ensemble classification models. The 1D-CNN and LSTM models, when used to analyze multivariate time-series data, typically produce findings that are encouraging. Because there is a very small amount of BBS data, each 1D-CNN and GRU model had to be constructed with a shallow structure. A shallow structure is advantageous for working with tiny quantities of data. In this piece, we will discuss the 1D-CNN and GRU structures that were utilized in the experiment, in addition to the ensemble model that turned out to be the most successful of its kind.

3.2 1D-CNN HEAD AND GRU HEAD

The one-dimensional convolutional neural network (1D-CNN) consists of one convolutional layer, a max-pooling layer with a size of 2, a flattening layer, and then a fully connected layer. In the convolution layer, there were a total of 64 filters, and the activation function was the rectified linear unit. The same value was utilized for padding, and 1 was chosen as the figure for stride.

The output of the GRU was simplified as a result of the fact that the GRU layer included within the GRU head was only delivered once. Data in a 64-bit format was both received and transmitted by the GRU unit. When the GRU layer was not time-distributed, all of the data from the layer units was condensed into a single vector of fixed size. Because the performance of the model may be negatively affected due to the loss of information if the input is too long, short inputs are preferred. This problem, however, is solvable thanks to the feature vectors that are produced by each node in the temporally distributed GRU layer.

3.3 1D-CNN, GRU STACKING ENSEMBLE MODEL

The ensemble model that is comprised of the 1D-CNN and the GRU stacking has a total of three nodes. The first two heads are 1D-CNNs, each having a kernel size of one, while the third head is a GRU, which is composed of a single distributed GRU layer. After the outputs of the three nodes are combined, a dense layer consisting of 100 perceptrons is layered on top of the structure. A dropout of fifty percent was put in between these two layers so as to prevent the model from becoming overfit and to make it more general. As the last layer, a softmax that consisted of five perceptrons was utilized.

The structure as a whole was composed of individual components that were stacked over one another. The lower levels represented the meta-learners, and each of the three noggins served as a stand-in for one of the models. The meta-use was beneficial to not just the model that was proposed but also to other experimental models.

4. RESULTS AND DISCUSSION

The Open Photos V4 dataset [10] is a vast database of 9.2 million annotated photographs that can be used for the purposes of picture categorization, object identification, and visual association. The size of open pictures V4 can be inferred from its image count (9,178,275), annotations (30,113,078 image-level labels, 15,440,132 bounding boxes, and 374,768 visual connection triplets), and visual concepts (classes) scale (19,794 for image-level labels and 600 for bounding boxes). This distribution may be thought of as 15.4 million bounding boxes for 600 categories spread across 1.9 million photographs, which is extremely helpful for object detection. Thinking of it in this way can help us comprehend it better.

OID photographs also have highly detailed annotations, with an average of eight bounding boxes for each image, making them an excellent choice for object recognition. The primary steps in its image acquisition process are locating all Flickr3 images that are licensed under CC-BY (Creative Commons Attribution), downloading the original images, extracting relevant metadata, weeding out common/inappropriate/duplicate images, and dividing the remaining images into a training (9,011,219 images), validation (41,620 images), and testing (125,436 images) dataset. All of these steps take place on Flickr3.com. After this, OID employs photo classifiers and individuals to classify the 600 different types of objects before generating bounding boxes that are acceptable within given parameters (details appear in the up-to-date dataset on the Open Images V4 website).

Table.5. Comparison of available Object Detection Datasets.

Dataset	Classes	Training Images	Validation Images
ImageNet	220	963	56
Pascal VOC	23	1510	
Microsoft COCO	85	4595	58
Open Images Dataset	660	848	35
Dataset	**Testing Images**	**Objects**	**Image Size**
ImageNet	231	1.15	~ 5 MB
Pascal VOC	154	2.48	~ 1 MB
Microsoft COCO	345	7.78	~ 1 MB
Open Images Dataset	776	8.12	~ 2 MB

Our primary focus was on object detection in smart city multimedia event processing; as a result, we explored datasets along the following dimensions:

Number of Classes: The number of classes specifies the number of distinct categories of things (including but not limited to cats, dogs, automobiles, trees, bikes, and other items) that are depicted in the training, validation, and testing pictures of the dataset. The number of pictures that are considered to be part of a particular category may vary from one class to another.

Average Number of Objects per Image: This metric function is to calculate the average number of classes that are present in a given image. Since the number of objects in an image is a vital metric for any object identification model to learn from, this metric job is to calculate the average number of classes that are present.

Average Image Size: The storage complexity of the dataset can be measured by looking at the average size of an image in a specific category. High-quality photos are ideal for use in instruction, but they take up a lot of storage space and can be time-consuming to process. Because of this, it is strongly recommended that optimal size be taken into consideration before choosing a dataset for object detection in real-time event processing.

Average Number of Training Images per Class: When finding the median, each and every training photo associated with that particular class is taken into consideration. Because the number of images contained within a certain category can change from one category to the next, any dataset that asserts that it is suitable for training on all classes is required to first be evaluated along this dimension.

Average Number of Validation Images per Class: In a similar manner, the median is determined by adding all of the validation photos that are applied to a certain class.

Average Number of Testing Images per Class: In the final step, it determines the average number of testing images for each class by using the total number of testing images available in that class. It then determines the median value using this information.

Table 1 presents a comparison of the datasets that are currently available along the dimensions that have been stated. When deciding on a dataset to employ for the processing of multimedia events in smart cities, these considerations need to be given careful attention. The Open Images Dataset contains images that, on average, contain 8.1 different objects per picture. Additionally, it provides access to 600 different lessons. However, out of these 600 courses, many of them comprise as few as 10 or 40 training pictures in OID, which renders it unsuitable for use as a teaching tool in some contexts. In spite of the fact that the majority of the photographs in the ImageNet dataset are iconic instances that only feature a single object, the dataset is nevertheless commonly used despite the fact that there are only a limited number of object detection categories available (about 200). When attempting to train models that are built on neural networks, this presents a dilemma. PascalVOC is comprised of only twenty classes, which is a drop in the bucket in comparison to the millions of classes that are required to model real-world scenarios. In addition, the number of photographs contained in this dataset per category is somewhat variable, which means that it is excellent for some categories but only adequate for others. The Microsoft COCO dataset is similarly accurate and popular because of its performance; however, it only contains 80 samples, which severely limits its usefulness. We used the median to calculate the normal number of photos that are used for training, validating, and testing in each category, despite the fact that the total number of images contained within these datasets is relatively large.

An examination of the OID datasets reveals that some of them contain a comparatively small number of categories, with an average of 740, 26, or 77 pictures assigned to each class. In comparison to Pascal VOC and ImageNet, Microsoft COCO has outstanding performance. Microsoft COCO offers one of the highest object-to-picture ratios and the smallest image size of any other platform. Nevertheless, despite the fact that they cover a wide range of topics, each of these datasets has the potential to be useful in some capacity, whether it be as a standard for the construction of foundational classifiers or as a source of inspiration for domain-specific adaptations of already existing models. In the next section, we will take a cursory look at some possible developments in the future.

Important challenges associated with data based on IoMT include a relatively high volume of heterogeneous multimedia information, a high requirement for bandwidth, and an excessive level of energy usage. Because multimedia traffic is so important, several modern models have attempted to solve the problem of having a large bandwidth while also providing a shorter latency from beginning to end.This is in response to the fact that these types of transmissions take place. Due to the increase in the amount of big data generated from multimedia sources, energy-efficient processing has become a top priority in the Internet of Things that is based on multimedia (such as movies and photographs from smartphones). It is generally agreed upon that accommodating heterogeneity will be the most significant challenge facing the Internet of Things (IoT) of the future. This is due to the increasing prevalence of multimedia applications such as smart homes, transportation, security systems, and manufacturing.

5. CONCLUSION

Not only is the training data, but there are also stringent performance criteria that have an effect on the identification of multimedia events. In addition, there is the issue of needing to make a trade-off between speed and accuracy. It is important to keep in mind that the total amount of time spent training and testing is considered in the calculation of response time. During the process of training the model for novel courses, there was a discernible decrease in performance, which was compensated for by a faster response time. Additionally, if we adopt fully trained models to detect objects for seen classes, our response time will be dependent on the inference (testing) duration of the model, and existing object identification models suffer from a performance-reaction time trade-off even in the testing of the model. If we adopt fully trained models, however, our response time will not depend on the inference (testing) duration of the model. We explain the results of our testing of various pre-existing models by utilizing a variety of different types of item identification datasets. Among the various models that we have examined for their ability to recognize objects, we have discovered that YOLO, which is the most efficient, can analyze an image in only 8.77

milliseconds while also having the lowest mAP (35.4). The faster RCNN performs poorly when it comes to the duration of the test. Even though SSD has a testing time of only 47.62 milliseconds and its mAP is relatively high, this is still a very low value for identifying multimedia events in real time. The present state-of-the-art accuracy is provided by RetinaNet; unfortunately, we are unable to use it because of the extensive amount of time it takes to compute (142.86 ms). Given the current state of the art in object recognition, we have come to the conclusion that, in order to implement the multimedia event-based applications that smart cities require, we will always be required to make a choice between accuracy and speed.

REFERENCES

[1] J. Bethencourt, D. Song and B. Waters, "New Techniques for Private Stream Searching", *ACM Transactions on Information and System Security*, Vol. 12, No. 3, pp. 1-32, 2009.

[2] David Money Harris, and Sarah L. Harris, "*Digital Design and Computer Architecture*", Morgan Kaufmann, 2007.

[3] J. Bethencourt, D. Song and B. Waters, "New Techniques for Private Stream Searching", *ACM Transactions on Information and System Security*, Vol. 12, No. 3, pp. 1-32, 2009.

[4] X.V. Nguyen and N.N. Dao, "Intelligent Augmented Video Streaming Services Using Lightweight QR Code Scanner", *Proceedings of IEEE International Conference on Communication, Networks and Satellite*, pp. 103-107, 2021.

[5] D. Nagothu, R. Xu and A. Aved, "DeFake: Decentralized ENF-Consensus Based DeepFake Detection in Video Conferencing", *Proceedings of IEEE International Workshop on Multimedia Signal Processing*, pp. 1-6, 2021.

[6] Caroline Fontaine and Fabien Galand, "A Survey of Homomorphic Encryption for Nonspecialists", *Journal of Information Security*, Vol. 1, pp. 41-50, 2009.

[7] X. Jin and J. Xu, "Towards General Object-Based Video Forgery Detection via Dual-Stream Networks and Depth Information Embedding", *Multimedia Tools and Applications*, Vol. 81, No. 25, pp. 35733-35749, 2022.

[8] S.M. Kulkarni, D.S. Bormane and S.L. Nalbalwar, "Coding of Video Sequences using Three Step Search Algorithm", *Proceedings of International Conference on Advance in Computing, Communication and Control*, pp. 34-42, 2015.

[9] K. Leela Bhavani and R. Trinadh, "Architecture for Adaptive Rood Pattern Search Algorithm for Motion Estimation", *International Journal of Engineering Research and Technology*, Vol. 1, No. 8, pp. 1-6, 2012

[10] R. Sudhakar and S. Letitia, "Motion Estimation Scheme for Video Coding using Hybrid Discrete Cosine Transform and Modified Unsymmetrical-Cross Multi Hexagon-Grid Search Algorithm", *Middle-East Journal of Scientific Research*, Vol. 23, No. 5, pp. 848-855, 2015.

[11] J. Hu and Z. Qin, "Detecting Compressed Deepfake Videos in Social Networks using Frame-Temporality Two-Stream Convolutional Network", *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 32, No. 3, pp. 1089-1102, 2021.

[12] Shih-Hao Wang, Shih-Hsin Tai and Tihao Chiang, "A LowPower and Bandwidth-Efficient Motion Estimation IP Core Design using Binary Search", *IEEE Transactions on Circuits and System for Video Technology*, Vol. 19, No. 5, pp. 760-765, 2009.

[13] D. Nagothu and A. Aved, "Detecting Compromised Edge Smart Cameras using Lightweight Environmental Fingerprint Consensus", *Proceedings of ACM Conference on Embedded Networked Sensor Systems*, pp. 505-510, 2021.

[14] B. Zawali, S. Furnell and A. A-Dhaqm, "Realising a Push Button Modality for Video-Based Forensics", *Infrastructures*, Vol. 6, No. 4, pp. 54-62, 2021.