computation

MDPI

Article

Detection of Shoplifting on Video Using a Hybrid Network

Lyudmyla Kirichenko [1,2], Tamara Radivilova [3], Bohdan Sydorenko [1] and Sergiy Yakovlev [4,5,*]

[1] Department of Applied Mathematics, Kharkiv National University of Radio Electronics, 61166 Kharkiv, Ukraine
[2] Applied Mathematics Department, Wroclaw University of Science and Technology, 50-370 Wroclaw, Poland
[3] Department of Infocommunication Engineering, Kharkiv National University of Radio Electronics, 61166 Kharkiv, Ukraine
[4] Mathematical Modelling and Artificial Intelligence Department, National Aerospace University "Kharkiv Aviation Institute", 61072 Kharkiv, Ukraine
[5] Institute of Information Technology, Lodz University of Technology, 90-924 Lodz, Poland
* Correspondence: s.yakovlev@khai.edu

Abstract: Shoplifting is a major problem for shop owners and many other parties, including the police. Video surveillance generates huge amounts of information that staff cannot process in real time. In this article, the problem of detecting shoplifting in video records was solved using a classifier, which was a hybrid neural network. The hybrid neural network included convolutional and recurrent ones. The convolutional network was used to extract features from the video frames. The recurrent network processed the time sequence of the video frames features and classified the video fragments. In this work, gated recurrent units were selected as the recurrent network. The well-known UCF-Crime dataset was used to form the training and test datasets. The classification results showed a high accuracy of 93%, which was higher than the accuracy of the classifiers considered in the review. Further research will focus on the practical implementation of the proposed hybrid neural network.

Keywords: human behavior; shoplifting; video surveillance; classification; features; neural network; gated recurrent units

Citation: Kirichenko, L.; Radivilova, T.; Sydorenko, B.; Yakovlev, S. Detection of Shoplifting on Video Using a Hybrid Network. *Computation* **2022**, *10*, 199. https://doi.org/10.3390/computation10110199

Academic Editors: Mykola Nechyporuk, Vladimir Pavlikov and Dmitriy Kritskiy

Received: 11 October 2022
Accepted: 1 November 2022
Published: 6 November 2022

1. Introduction

As one of the many crimes committed in stores, shoplifting attracts a lot of attention. Shoplifting is the theft of goods from an open retail establishment, usually by concealing items from the store in one's pockets, under clothing, or in a bag and leaving the store without paying. Shoplifting is a big problem for store owners and many other parties, including the police, government, and courts. According to a study by the National Association for Shoplifting Prevention, 1 in 11 people is a shoplifter. Moreover, it has been reported that thieves of this kind are arrested only once in every 48 thefts [1].

There are dozens of implementation methods for shoplifting; most of them usually involve hiding things by a person or an accomplice and leaving the store without paying. The key word is "hiding", accompanied by a characteristic sequence of actions. In terms of the behavior of the average shopper, it is abnormal behavior. Of course, when a person reviews a recorded incident of shoplifting, it is not difficult to trace this behavior and discover if the theft actually occurred. However, as the number of shoppers increases, so does the number of shoplifters, which creates the problem of not being able to monitor every one of them.

Every year, the number of video surveillance cameras in public places such as streets, banks, shopping malls, and retail stores grows to improve public safety. Video camera networks generate huge amounts of data and security personnel cannot process all the information in real time. The more recording devices become available, the more difficult the task of monitoring becomes [2].

Thus, there is a need for automatic video surveillance that detects shoplifting events [3]. In recent years, the solution to the problem of real-time monitoring and information processing has been the application of artificial intelligence methods; in particular, algorithms for anomaly detection and event classification [1,4–7].

2. Review of the Literature

The most common approaches to applying artificial intelligence to video monitoring data primarily include motion detection, face recognition, surveillance, inactivity detection, and anomaly behavior detection [8].

In [9], the authors conducted research on shoplifting classifications based on the Jubatus plug-in to extract the feature values from images to assess anomalous customer behavior. In the proposed application, the surveillance video data were classified using a linear classifier and a kNN classifier and the probability of shoplifting was determined.

Tsushita and Zin [10] presented an algorithm for video surveillance detection, violence, and theft. Their approach divided the frame into eight areas and looked for speed changes in the person being monitored.

In [11], the authors proposed a model of a genetic algorithm generating neural networks to classify human behavior in videos. The authors developed shallow and deep neural networks that used the posture changes of people in video sequences as the input.

To detect violent theft in video sequences, [12] proposed using a deep learning sequence model in which a feature extractor was trained. The features were then processed with two layers of long-term convolutional memory and passed through fully connected layers for the classification.

To accurately recognize human actions, the authors in [13] proposed to obtain features based on MobileNetV2 and Darknet53 deep learning models. The selected features, based on an improved particle swarm optimization algorithm, were used to classify actions in the video sequences using different classifiers. The effectiveness of the proposed approach was tested by the authors in experiments on six publicly available datasets.

One obvious and popular approach to detect theft is to classify the actions presented in the video using convolutional neural networks. Convolutional neural networks have shown superior performance in computer vision in recent years. In particular, 3D CNN networks, an extension of the convolutional neural networks of CNN, focus on extracting spatial and temporal features from videos. Programs that have been implemented with a 3D CNN include object recognition, human action recognition, and gesture recognition.

In [14], the authors presented an approach to anomaly detection using a pretrained C3D model for the feature extraction and a full-link neural network to perform the regression. Using the UCF-Crime dataset [15], the authors trained their model on 11 classes and tested its results on videos of theft, fights, and traffic incidents.

The authors of [16] presented an approach to detect anomalies in real time; the algorithm was taught to classify 13 anomalous behaviors such as theft and burglary, fighting, shooting, and vandalism. They used a 3D CNN network to extract the features and label the samples into two categories: normal and abnormal. Their model included a rating loss function and trained a fully connected neural network to make decisions.

The authors of [17] used 3D convolutional neural networks to recognize suspicious actions. The 3D cuboid of the motion based on the frame difference method was used to detect and recognize real-time actions in video sequences. The effectiveness of the proposed methods was tested with the implementation of two sets of videos.

In [18], the authors analyzed the presence of violence in surveillance videos. For this purpose, the authors proposed the use of a deep learning model based on 3D convolutional neural networks without using manual functions or RNN architecture exclusively for encoding temporal information. To evaluate the effectiveness and efficiency of the model, the authors experimentally tested it on three benchmark datasets.

In [19], the authors investigated existing video classification procedures in order to recommend the most efficient and productive process. The authors showed that the

combined use of a CNN (convolutional neural network) and an RNN (recurrent neural network) performed better than CNN-dependent methods.

In the manuscript of [20], it was proposed to combine a 3D CNN and LSTM to predict human actions. In the approach, the 3D CNN was used for the feature extraction and LSTM for the classification. The result of the model depended on the pose, illumination, and environment. The reliability of these features allowed the prediction of human actions.

In [21], the authors also used a 3D CNN for the feature extraction and classification. They analyzed the effectiveness of this neural network model on a dataset of real-time shoplifting videos.

In [22], the authors presented an expert system for recognizing the actions of thieves. The system used a convolutional neural network to analyze the typical features of the motion of the thief and a deep learning module based on long-term memory was used to train the extracted features. This system alerted shoplifting based on an analysis of the appearance and motion features in a video sequence.

3. Problem Statement

Summarizing the research review, we proposed to solve the problem of shoplifting detection using a binary classification of customer behavior based on a video fragment in two classes: "shoplifting" and "non-shoplifting". For this purpose, it was desirable to develop a classifier that was a symbiosis of two neural networks: convolutional and recurrent. The convolutional neural network removed the features from each frame of the video and the recurrent network processed the time sequence of the processed frames and the further classification.

The input data were video sequences with the same duration and number of frames for which it was known whether a theft had occurred or not. A sequence of frames was formed from each video sequence. Each such video sequence was an object, which was labeled with one of two classes: 0—not shoplifting or 1—shoplifting. The labeled set of frame sequences was the training set. During the classifier training, each object passed the stage of feature acquisition through the hybrid neural network. The training resulted in a classifier, which was further used to classify new objects.

The aim of this work was to develop, train, and test a classifier based on a hybrid neural network to detect shoplifting from video monitoring data.

4. Materials and Methods

4.1. Forming the Input Dataset

The UCF-Crime dataset was used to form the training test dataset [15,16,23]. UCF-Crime is a dataset of 128 h of video. It consists of 1900 long and uncut videos of real-life criminal events, including violent incidents, arrests, arson, assault, traffic accidents, burglaries, explosions, fights, robberies, shootings, thefts, shoplifting, and vandalism.

The dataset with shoplifting had 28 videos recorded from surveillance cameras in retail stores, which had recorded incidents of shoplifting. Due to the fact that the dataset had a small number of videos, the dataset was artificially enlarged by splitting each video into 32 video episodes. Thus, 896 samples with a duration of 3 s each were obtained. The dataset was divided into 2 classes: "non-shoplifting" (Class 0) and "shoplifting" (Class 1). As a result, we obtained two classes of events with a division of the number of videos in each class: 741 videos of "not shoplifting" and 155 of "shoplifting".

Figure 1a shows a frame that corresponded with a video where shoppers simply chose products. The frame in Figure 1b corresponded with a video with the presence of "shoplifting".

(**a**)

(**b**)

Figure 1. Video frames: (**a**) not shoplifting; (**b**) shoplifting.

4.2. Choice of Neural Networks as a Classifier

The two most widely used deep learning architectures for video classifications are convolutional and recurrent neural networks. CNNs are mainly used to learn spatial information from a video whereas RNNs are used to learn temporal information from a video. These network architectures are used for very different purposes. However, the nature of video data with both spatial and temporal information requires the use of both of these network architectures [24,25].

Convolutional neural networks are a variation of neural networks, which are aimed at solving problems of image recognition and the detection of objects in them with the help of computer vision. The property that allows the solving of the above-mentioned problems is that convolutional neural networks form new information segments with reduced dimensionality but preserved features and combine them using several neural layers of different types [26]. Usually CNNs include convolutional layers, aggregated layers, fully connected layers, and normalization layers. A convolutional layer simulates the response of an individual neuron to a visual stimulus. The convolution layers apply a convolution operation to the input, passing the result to the next layer [27].

Recurrent neural networks, or RNNs, are a type of neural network specializing in value sequence processing [28]. Convolutional networks scale to images with a large width

and height; recurrent networks can scale to much larger sequences than would be practical for networks without a sequence-based specialization. Most recurrent networks can also handle sequences of a variable length [29].

As the output of a recurrent neuron at a time step is a function of all inputs at previous time steps, we can state that the neuron has a form of memory. The part of a neural network that stores the state through the time steps is called a memory cell.

The most popular long-term memory cell is the LSTM (long short-term memory) cell [30]. LSTM was created to avoid the problem of long-term dependency. An LSTM cell is a set of layers that interact with each other according to certain rules. Layers are connected using an element-by-element addition and multiplication operations.

Gated recurrent units are a type of recurrent neural network [31]. Similar to LSTM, they have been proposed to solve problems such as computing gradients in long-term memory. The two blocks work the same in many cases, but a GRU is trained faster. The block itself has a much lower number of parameters, which makes it much more efficient. Gated recurrence units are a new block and only started to be used in 2014, which explains its lower popularity [32].

In this paper, we used a video classification method that extracted the features from each frame by a CNN and passed the sequence to a separate GRU convolutional neural network. The convolutional neural network was used as a feature extractor so we obtained a sequence of feature vectors.

Feature extraction consists of determining the most relevant characteristics of images and assigning labels to them. In image classification, the decisive step is to analyze the properties of the image features and numerical features into classes. In other words, the image is classified according to its content. The efficiency of a classification model and the degree of classification accuracy mainly depend on the numerical properties of the different image features that represent these classification models. In recent years, many feature extraction methods have been developed; each method has advantages and disadvantages [33].

Transfer learning was used to extract the features especially pretrained from the ImageNet dataset CNN MobileNetV3Large from Keras [23,34]. As this neural network has been designed for low-resource use cases—namely, for phone central processors—it has a satisfactory execution speed, which guarantees its well-functioning performance in a real-time system.

As we used a pretrained model, its weights were loaded along with the model architecture; in this way, we used uploaded parameters. The feature extraction used a neural network without a fully connected layer at the top to obtain a set of features before they went into a prediction. The neural network model preprepared on the ImageNet dataset together with the weights could be loaded with Keras [34,35].

ImageNet is a project that aims to label and classify images into nearly 22,000 categories based on a specific set of words and phrases. At the time of writing, there were over 14 million images in the ImageNet project [34]. MobileNetV3Large received an image interpreted as an array with a shape of $224 \times 224 \times 3$ as the input and at the output, we got a feature vector with size of 960.

4.3. Assessment of the Classification Accuracy

To evaluate the obtained results during a binary classification, an error matrix with true positive (TP), false positive (FP), false negative (FN), and true negative (TN) values was most often obtained.

The following characteristics were chosen as the classification results in such a case.

Accuracy: the fraction of correct answers of the algorithm was found using the fractional expression:

$$\frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{FP} + \mathrm{FN} + \mathrm{TN}}. \quad (1)$$

However, we should keep in mind that this metric was not very informative in tasks with unbalanced classes.

Precision: the fraction of objects that the classifier called positive and were, in fact, positive was defined using the expression:

$$\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}}. \tag{2}$$

Recall: The proportion of objects of a positive class from all objects of a positive class was found using the expression:

$$\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}}. \tag{3}$$

This indicator demonstrated the ability of the algorithm to detect this class as a whole; the accuracy indicator helped to distinguish this class from other classes.

As with the precision measure, the recall measure was independent of the class correlation and, therefore, was applicable in unbalanced samples, unlike the accuracy measure.

F-measure (F-score): One of several ways to combine precision and recall into one aggregate criterion is to compute an F-measure. In this case, it was the harmonic mean of accuracy and recall.

The error curve or AUC–ROC curve (area under curve–receiver operating characteristic curve) is a graphical characteristic of the binary classifier quality. The dependence of the percentage of true positive classifications was the true positive rate (TPR):

$$\mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}} \tag{4}$$

from the false positive rate (FPR) of the false classifications (FPR):

$$\mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{FP}+\mathrm{TN}}. \tag{5}$$

In an ideal case, when the classifier does not make mistakes (FPR = 0, TPR = 1), the square under the curve is equal to one; if the classifier outputs the same number of TP and FP results, the AUC–ROC will approach 0.5. The area under the curve shows the quality of the algorithm so the bigger the square, the better.

5. Experiments

We built an algorithm to solve the shoplifting recognition problem as a classification problem. It is possible to point out the four main stages of the algorithm.

1: At the first stage, the data collection, a set of samples was collected for further use in the training of the model and its testing. In our study, it was a set of videos with recorded cases of shoplifting from surveillance cameras in retail stores. As the dataset was unbalanced, the predominant class was under sampled; i.e., a number of instances with the label "not shoplifting" were removed so that the number of video fragments in the two classes was equal (155).

As the dataset was somewhat small for such a non-trivial task as the classification of human actions (only 310 instances), it was decided to artificially enlarge it. Each video fragment was horizontally mirrored, after which we obtained 620 instances. From each of the 620 video fragments, 2 more copies were formed that were rotated 5 degrees to the left and 5 degrees to the right. Thus, we obtained a dataset of 1860 video fragments.

2: At the second stage of data preprocessing, work was performed on the data. The video fragments were labelled into classes (Class 1: cases with shoplifting and Class 0: normal customer behavior). Processing was also performed before the feature extraction by resizing the image to a size of 224 × 224 pixels and dividing the video into frames. As each video fragment was 3 s long by 10 frames/second, this provided sequences of 30 frames. The dataset was divided into training and test sets (1302 training and 558 test sets).

3: During the third stage, the feature extraction was performed using the convolutional neural network MobileNetV3Large preprepared on the ImageNet-1k dataset.

4: We created, trained, and tested a recurrent neural network with layers of gated recurrent nodes. The features extracted by the convolutional network MobileNetV3Large from each image of the labeled sequences of frames of video fragments were delivered for training in a recurrent network with gated nodes.

Figure 2 shows the main stages of the classification algorithm under consideration.

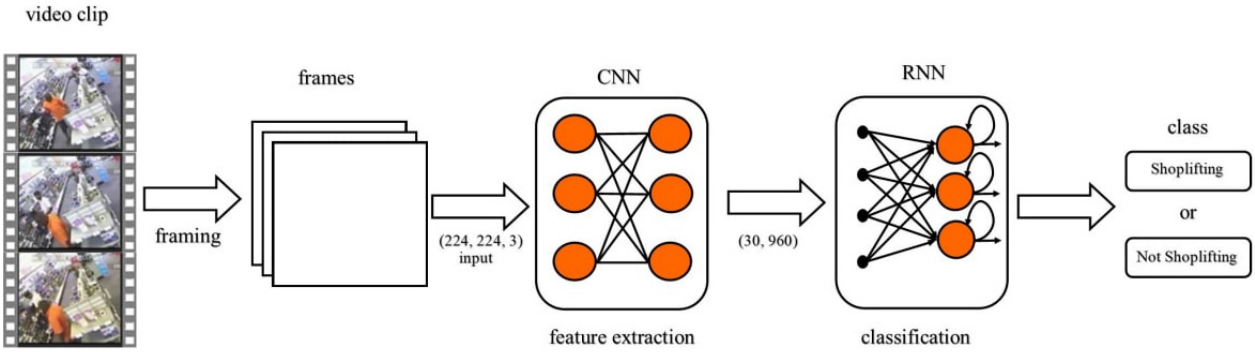

Figure 2. The main stages of the classification algorithm.

The program to solve the problem of video surveillance shoplifting recognition was written in the Python programming language. The web interactive computing environment Jupyter Notebook was used to process the dataset and the Colaboratory environment from Google Research was used to build the neural network architecture and train and test the model. This resource allowed the program to run in a cloud environment with Google Tensor Processors (TPU), which satisfied the need for a high speed when working with machine learning packages; namely, the TensorFlow software library that used the tensor calculations.

6. Results and Discussion

To achieve a sufficiently high accuracy of the classifier, a lot of research and experimentation was conducted in practice; in particular, the choice of video classification method, the search for a dataset, the search for optimal data processing and neural networks, and their configurations with parameters.

Data processing is an important factor in preparing model training. As noted above, we performed an artificial increase in the dataset as training on 310 samples achieved an accuracy of about 77% whereas in the case of a dataset increased to 1860 samples, the accuracy increased by about 5–7%.

The choice and configuration of the neural networks were also very important. The first experiments were performed on a combination of a convolutional InceptionV3 network [35] and a recurrent neural network with gated nodes. For such a set of neural networks, the speed of the program in real time was low because the extraction of the features took a long time, so we had to modify the convolutional neural network. Our task was to find a network that had fewer features in the output and would find the important information in the frame. After a number of experiments, we found an optimum neural network: MobileNetV3Large, which had a size of (1, 960) on the output in contrast to two times larger (1, 2048) in InceptionV3. As a result, the accuracy increased by approximately 10%.

One of the few hyperparameters of the convolutional neural network available for selection was the dimension of the video fragment. This was chosen based on the parameters of the input layer. We reduced all instances to $224 \times 224 \times 3$, meaning that the resolution was 224×224 and used the three-color model (RGB). We also removed the top layer in order to perform the feature extraction. At the output, we had a vector with a dimension of 1×960 (one frame)—that is, 960 features—which was a constant value for this neural

network. By trial and error, the optimal number of frames per sequence was obtained, which was 30.

An important step in the optimization was the selection of the batch size parameter, which meant the number of features per training iteration. Table 1 shows the accuracy and loss rates for each parameter value from 4 to 64 for the training and validation samples. The training sample, which occupied 70% of the dataset, was divided for the model training into a training sample and a validation sample at a ratio of 7:3. The validation sample was an intermediate training phase, which was used to select the best model and optimize it.

Table 1. Indicators for each investigated batch size value.

Batch Size	Iterations	Training Accuracy	Training Loss	Validation Accuracy	Validation Loss	Test Accuracy
4	228	99.34	0.026	90.28	0.407	90.14
8	114	98.02	0.066	92.84	0.247	91.27
16	57	99.34	0.027	90.79	0.326	92.11
32	29	98.57	0.051	90.79	0.302	92.83
64	15	99.45	0.015	93.09	0.338	93.19

Such hyperparameters were selected by a large number of experiments: batch size: 64; epochs: 60; the ratio of the training and test data: 70:30; and the quality metric in terms of training: accuracy. As we only had two classes, binary cross-entropy loss was chosen to perform in the recurrent neural network.

We concluded that this configuration (including all the factors listed above) was optimal. Below, we present the results of this particular case.

An important and necessary part of the process of selecting the hyperparameters was the construction of the architecture of the sequence model (recurrent classifier). The set of layers was as follows: (1) at the input, we had a GRU layer with 32 units; (2) the GRU layer was next, with 16 units; (3) a dropout layer was used with a rate of 0.4 to reduce overfitting; (4) a fully connected layer was used with 8 neurons with a Relu activation function; and (5) the output was a fully connected layer with 1 neuron and a sigmoid activation function, which yielded the probability of a theft.

Using the Adam optimizer (Adaptive Momentum) from Keras, we obtained a leaning rate of 0.001 by default and the descent stochastic gradient momentum was adaptively adjusted as determined by the Adam optimizer.

The graph of changes in the values of the training and validation accuracy depending on the epoch of the training is shown in Figure 3.

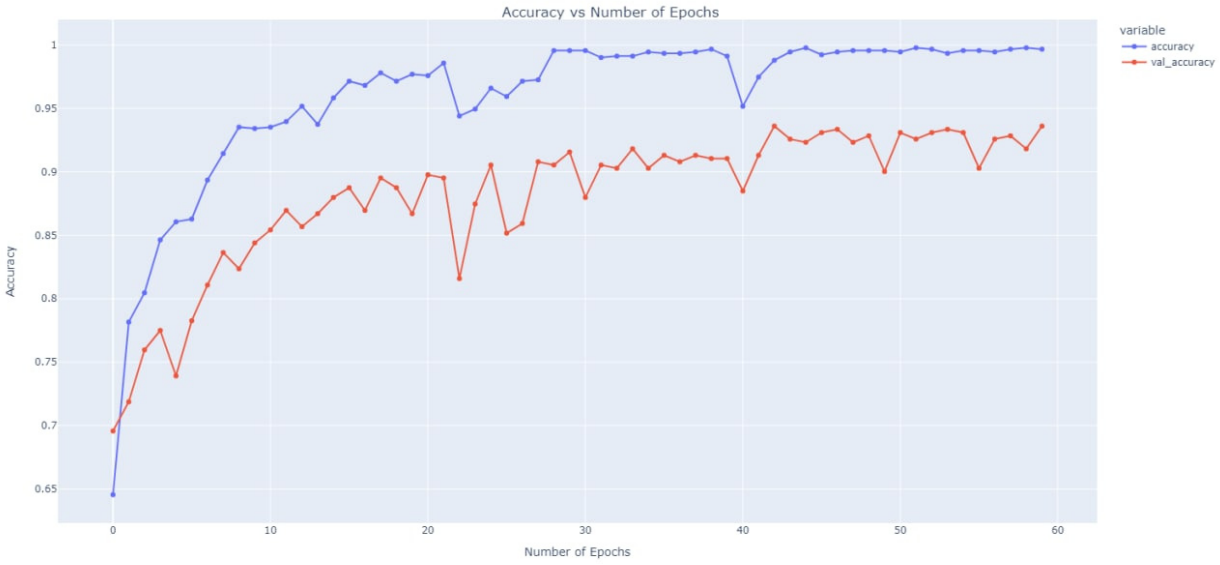

Figure 3. Training and validation accuracy depending on the epoch.

The graph of changes in the training and validation loss values is shown in Figure 4.

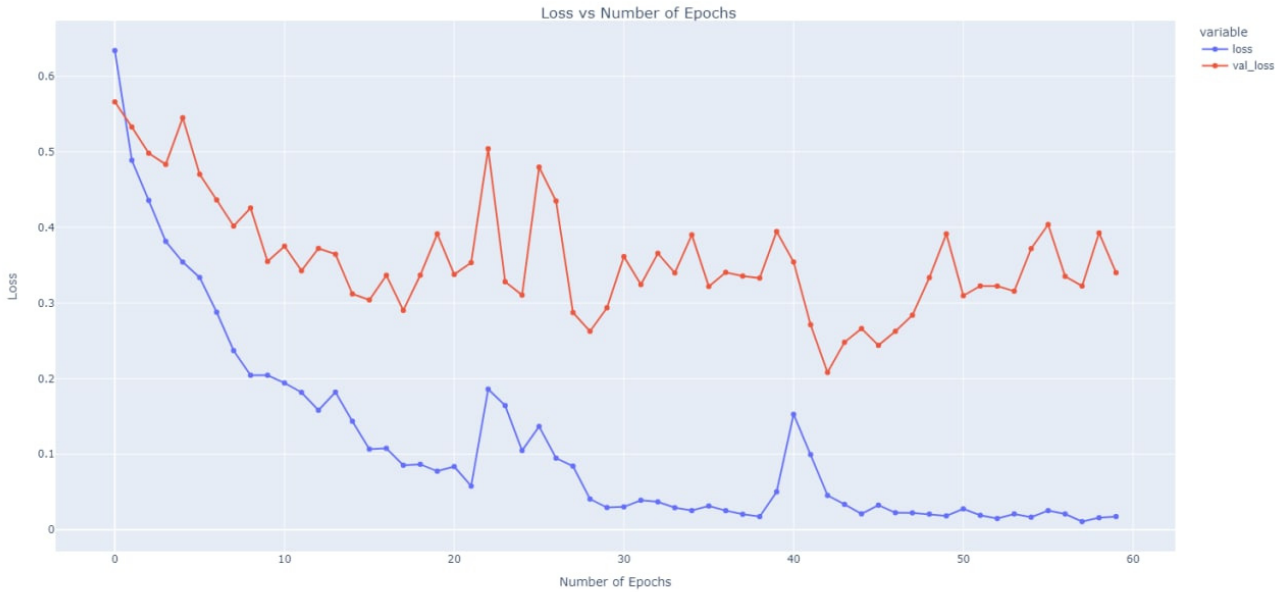

Figure 4. Training and validation loss values depending on the epoch.

The classification results for the test sample are displayed in the confusion matrix in Table 2; the rows show the true positive (TP) and false negative (FN) classification results and the columns show the false positive (FP) and true negative (TN) values.

Table 2. Confusion matrix.

	Normal Behavior (Class 0)	**Shoplifting (Class 1)**
Predicted Class 0	261	19
Predicted Class 1	22	256

Table 3 shows the values of the calculated metrics of the precision, recall, and F1-score in relation to each of the classes.

Table 3. Values of precision, recall, and F1-score.

	Precision	**Recall**	**F1-Score**	**Support**
Normal Behavior (Class 0)	0.92	0.93	0.93	280
Shoplifting (Class 1)	0.93	0.92	0.93	278
Accuracy	–	–	0.93	558

Thus, the classification accuracy was 0.93. As the sample was balanced and given the presented values of the precision, recall, and F1-score, this value fully characterized the classification result.

The quality of the prompted classifier was also evaluated using an additional ROC curve (Figure 5). The area under it (AUC value) was 0.97.

It should be noted that the classification accuracy of 93% was several percent higher than the accuracy obtained in similar studies. In [21], the classification accuracy was 75%. The researchers in [22] obtained an accuracy of 90.26%, but did not use real video surveillance data, only a synthesized set of videos.

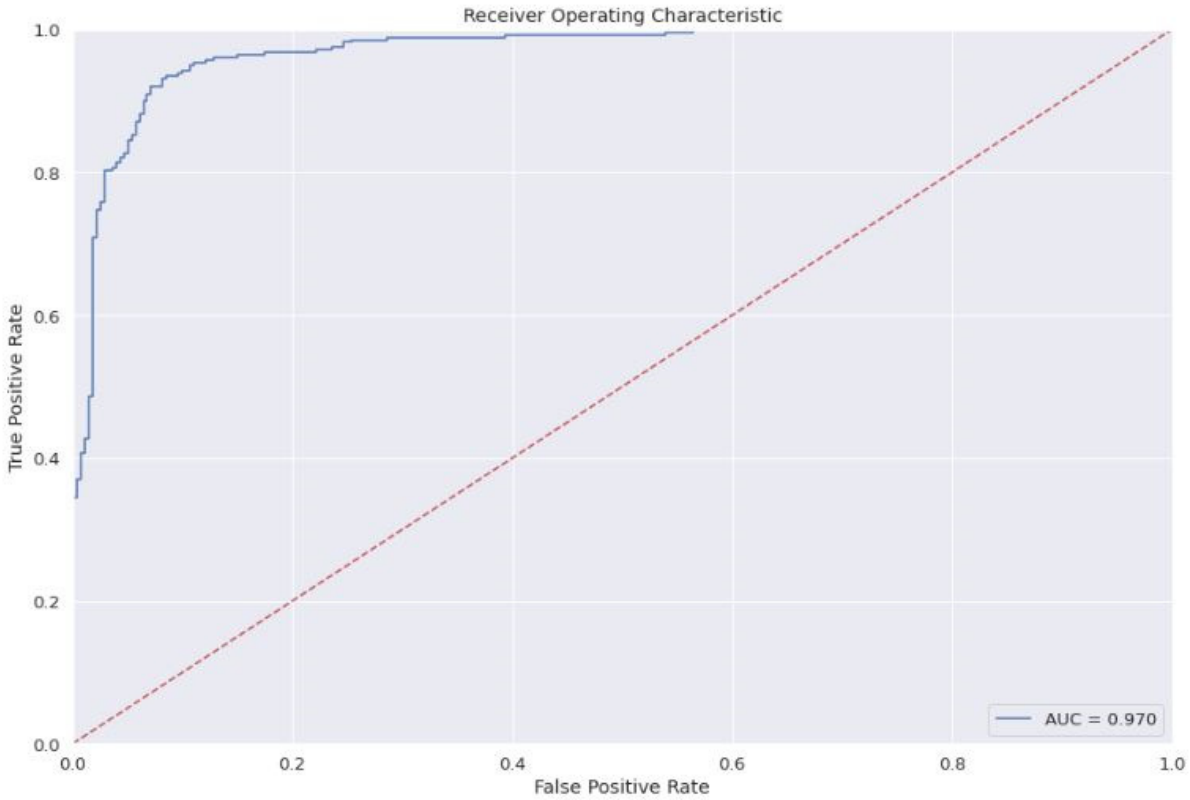

Figure 5. ROC curve and AUC value.

7. Conclusions

In this article, a classifier of video data from surveillance cameras was proposed to identify fragments with cases of shoplifting. The classifier represented a symbiosis of convolutional and recurrent neural networks. With this approach, a convolutional neural network was used to extract the features from each frame of a video recording and a recurrent network was applied to process the time sequence of the video frames and classify them. As a recurrent network, a variety of gated recurrent units was selected.

To teach the hybrid neural network, the popular UCF-Crime dataset was used. This contained video recordings of consumer behavior, including shoplifting. The original dataset contained data that were class-unbalanced; therefore, the sample of the predominant class was reduced. The resulting balanced sample of video data was artificially increased, which made it possible to improve the network training. The experiments were carried out with a pretrained convolutional neural network.

A neural network with ventilated recurrent nodes was used to classify the sequence of the video fragments. The classification results showed a high accuracy of 93%, which was several percent higher than the accuracy of the classifiers considered in the review of similar studies. The trained classifier had a high performance sufficient for a real-time operation. Further research will focus on the practical implementation of the proposed hybrid neural network in shopping malls.

Author Contributions: Conceptualization, L.K. and S.Y.; methodology, L.K. and T.R.; software, B.S.; validation, T.R. and B.S.; formal analysis, L.K. and S.Y.; investigation, L.K. and S.Y.; resources, T.R.; writing—original draft preparation, L.K. and S.Y.; writing—review and editing, L.K. and T.R.; visualization, B.S.; supervision, L.K. and S.Y.; project administration, L.K. and T.R. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Beethoven Grant No. DFG-NCN 2016/23/G/ST1/04083 and by a Grant of the Ministry of Education and Science of Ukraine "Technologies, tools for mathematical modeling, optimization and system analysis of coverage problems in space monitoring systems".

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Generated data and test tasks were used.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Chemere, D.S. Real-time Shoplifting Detection from Surveillance Video. Master Thesis, Addis Ababa University, Addis Ababa, Ethiopia, 2018; p. 94.
2. Kirichenko, L.; Radivilova, T. Analyzes of the distributed system load with multifractal input data flows. In Proceedings of the 2017 14th International Conference The Experience of Designing and Application of CAD Systems in Microelectronics, CADSM 2017, Lviv, Ukraine, 21–25 February 2017; pp. 260–264.
3. Gim, U.J.; Lee, J.J.; Kim, J.H.; Park, Y.H.; Nasridinov, A. An Automatic Shoplifting Detection from Surveillance Videos. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY USA, 7–12 2020; Apress: Berkeley, CA, USA, 2020; Volume 34, pp. 13795–13796.
4. Ivanisenko, I.; Kirichenko, L.; Radivilova, T. Investigation of multifractal properties of additive data stream. In Proceedings of the 2016 IEEE 1st International Conference on Data Stream Mining and Processing, Lviv, Ukraine, 23–27 August 2016; pp. 305–308.
5. Kirichenko, L.; Radivilova, T.; Bulakh, V. Machine learning in classification time series with fractal properties. *Data* **2019**, *4*, 5. [CrossRef]
6. Pang, G.; Shen, C.; Cao, L.; van den Hengel, A. Deep Learning for Anomaly Detection: A Review. *ACM Comput. Surv.* **2020**, *1*, 36. [CrossRef]
7. Radivilova, T.; Kirichenko, L.; Ageiev, D.; Bulakh, V. Classification methods of machine learning to detect DDoS attacks. In Proceedings of the 2019 10th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, IDAACS, Metz, France, 18–21 September 2019; pp. 207–210.
8. Rehman, A.; Belhaouari, S.B. Deep Learning for Video Classification: A Review. *TechRxiv* **2021**, *preprint*.
9. Yamato, Y.; Fukumoto, Y.; Kumazaki, H. Security camera movie and ERP data matching system to prevent theft. In Proceedings of the 2017 14th IEEE Annual Consumer Communications & Networking Conference (CCNC), Las Vegas, NV, USA, 8–11 January 2017; pp. 1014–1015.
10. Tsushita, H.; Zin, T.T. A Study on Detection of Abnormal Behavior by a Surveillance Camera Image. In *Big Data Analysis and Deep Learning Applications*; Zin, T.T., Lin, J.C.W., Eds.; Springer: Singapore, 2019; pp. 284–291.
11. Flores-Munguia, C.; Ortiz-Bayliss, J.C.; Terashima-Marin, H. Leveraging a Neuroevolutionary Approach for Classifying Violent Behavior in Video. *Comput. Intell. Neurosci.* **2022**, *2022*, 1279945. [CrossRef] [PubMed]
12. Morales, G.; Salazar-Reque, I.; Telles, J.; Diaz, D. Detecting violent robberies in cctv videos using deep learning, IFIP advances in information and communication technology. In *Artificial Intelligence Applications and Innovations*; Springer International Publishing: Cham, Switzerland, 2019; pp. 282–291.
13. Akbar, M.N.; Riaz, F.; Awan, A.B.; Khan, M.A.; Tariq, U.; Rehman, S. A Hybrid Duo-Deep Learning and Best Features Based Framework for Action Recognition. *Comput. Mater. Contin.* **2022**, *73*, 2555–2576.
14. Nasaruddin, N.; Muchtar, K.; Afdhal, A.; Dwiyantoro, A.P.J. Deep anomaly detection through visual attention in surveillance videos. *Big Data* **2020**, *7*, 87. [CrossRef]
15. University of Central Florida. UCF-Crime Dataset. Available online: https://www.v7labs.com/open-datasets/ucf-crime-dataset (accessed on 17 September 2022).
16. Sultani, W.; Chen, C.; Shah, M. Real-world anomaly detection in surveillance videos. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 6479–6488.
17. Arunnehru, J.; Chamundeeswari, G.; Bharathi, S.P. Human action recognition using 3D convolutional neural networks with 3D motion cuboids in surveillance videos. *Procedia Comput. Sci.* **2018**, *133*, 471–477. [CrossRef]
18. Li, J.; Jiang, X.; Sun, T.; Xu, K. Efficient Violence Detection Using 3D Convolutional Neural Networks. In Proceedings of the 2019 16th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), Taipei, Taiwan, 18–21 September 2019; pp. 1–8.
19. Islam, M.S.; Sultana, S.; Kumar Roy, U.; Al Mahmud, J. A review on video classification with methods, findings, performance, challenges, limitations and future work. *J. Ilm. Tek. Elektro Komput. Dan Inform. (JITEKI)* **2020**, *6*, 47–57. [CrossRef]
20. Alfaifi, R.; Artoli, A.M. Human action prediction with 3D-CNN. *SN Comput. Sci.* **2020**, *1*, 286. [CrossRef]
21. Martinez-Mascorro, G.A.; Abreu-Pederzini, J.R.; Ortiz-Bayliss, J.C.; Garcia-Collantes, A.; Terashima-Marin, H. Criminal Intention Detection at Early Stages of Shoplifting Cases by Using 3D Convolutional Neural Networks. *Computation* **2021**, *9*, 24. [CrossRef]
22. Ansari, M.A.; Singh, D.K. ESAR, An Expert Shoplifting Activity Recognition System. *Cybern. Inf. Technol.* **2022**, *22*, 190–200. [CrossRef]
23. Harvey, M. Five Video Classification Methods Implemented in Keras and TensorFlow: Exploring the UCF101 Video Action Dataset. 2017. Available online: https://blog.coast.ai/five-video-classification-methods-implemented-in-keras-and-tensorflow-99cad29cc0b5. (accessed on 14 September 2022).

24. Kirichenko, L.; Alghawli, A.S.A.; Radivilova, T. Generalized approach to analysis of multifractal properties from short time series. *Int. J. Adv. Comput. Sci. Appl.* **2020**, *11*, 183–198. [CrossRef]
25. Wang, J.; Yang, Y.; Mao, J.; Huang, Z.; Huang, C.; Xu, W. Cnn-rnn: A unified framework for multi-label image classification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 2285–2294.
26. Gollapudi, S. *Learn Computer Vision Using OpenCV: With Deep Learning CNNs and RNNs*, 1st ed.; Apress: Berkeley, CA, USA, 2019; p. 171.
27. Nebauer, C. Evaluation of convolutional neural networks for visual recognition. *Neural Netw. IEEE Trans.* **1998**, *9*, 685. [CrossRef] [PubMed]
28. Medsker, L.; Jain, L.C. (Eds.) *Recurrent Neural Networks: Design and Applications (International Series on Computational Intelligence)*, 1st ed.; CRC Press: Boca Raton, FL, USA, 1999; p. 416.
29. Time Series Classification. Welcome to the UEA & UCR Time Series Classification Repository. Available online: http://www.timeseriesclassification.com (accessed on 9 May 2022).
30. Segall, R.S.; Niu, G. *Biomedical and Business Applications Using Artificial Neural Networks and Machine Learning*; IGI Global: Hershey, PA, USA, 2022; p. 394.
31. Shah, S. *Implementation and Evaluation of Gated Recurrent Unit for Speech Separation and Speech Enhancement*; Northern Illinois University: DeKalb, IL, USA, 2019; p. 91.
32. LazyProgrammer. *Deep Learning: Recurrent Neural Networks in Python: LSTM, GRU, and More RNN Machine Learning Architectures in Python and Theano*; Machine Learning in Python; LazyProgrammer: Apress Berkeley, CA, USA, 2021; p. 93.
33. Medjahed, S.A. A Comparative Study of Feature Extraction Methods in Images Classification. *Int. J. Image Graph. Signal Process.* **2015**, *7*, 16. [CrossRef]
34. ImageNet Database. Available online: https://image-net.org/index.php (accessed on 8 July 2020).
35. Keras API Reference/Keras Applications/MobileNet, MobileNetV2, and MobileNetV3. Available online: https://keras.io/api/applications/ (accessed on 14 September 2022).