data

MDPI

Article

Advances in Contextual Action Recognition: Automatic Cheating Detection Using Machine Learning Techniques

Fairouz Hussein [1,*], Ayat Al-Ahmad [2], Subhieh El-Salhi [1], Esra'a Alshdaifat [1] and Mo'taz Al-Hami [1]

[1] Department of Computer Information System, Faculty of Prince Al-Hussein Bin Abdallah II for Information Technology, The Hashemite University, P.O. Box 330127, Zarqa 13133, Jordan
[2] Department of Computer Science and Applications, Faculty of Prince Al-Hussein Bin Abdallah II for Information Technology, The Hashemite University, P.O. Box 330127, Zarqa 13133, Jordan
* Correspondence: fairouzf@hu.edu.jo; Tel.: +962-791329214

Abstract: Teaching and exam proctoring represent key pillars of the education system. Human proctoring, which involves visually monitoring examinees throughout exams, is an important part of assessing the academic process. The capacity to proctor examinations is a critical component of educational scalability. However, such approaches are time-consuming and expensive. In this paper, we present a new framework for the learning and classification of cheating video sequences. This kind of study aids in the early detection of students' cheating. Furthermore, we introduce a new dataset, "actions of student cheating in paper-based exams". The dataset consists of suspicious actions in an exam environment. Five classes of cheating were performed by eight different actors. Each pair of subjects conducted five distinct cheating activities. To evaluate the performance of the proposed framework, we conducted experiments on action recognition tasks at the frame level using five types of well-known features. The findings from the experiments on the framework were impressive and substantial.

Keywords: action recognition; machine learning; cheating; computer vision; feature extraction; video surveillance

Citation: Hussein, F.; Al-Ahmad, A.; El-Salhi, S.; Alshdaifat, E.; Al-Hami, M. Advances in Contextual Action Recognition: Automatic Cheating Detection Using Machine Learning Techniques. *Data* **2022**, *7*, 122. https://doi.org/10.3390/data7090122

Academic Editors: Antonio Sarasa Cabezuelo and Ramón González del Campo Rodríguez Barbero

Received: 1 August 2022
Accepted: 29 August 2022
Published: 31 August 2022

1. Introduction

Interest in monitoring examinations and their mechanisms is increasing. Universities and academic institutions around the world are racing to obtain the latest technologies to monitor cheating in exam halls and secure a cheat-free environment. Typically, to ensure the management of examinations and detect cheating in exams, professional proctors are employed to supervise the entire examination process. In conjunction with the change in the examination control system worldwide due to COVID-19, all universities and institutes are now seeking to work with an electronic mechanism to monitor paper and electronic exams in order to provide safe and secure exams. They are also keen to use the latest mechanisms to detect cheating methods in exams. This is what universities and academic institutes around the world have been planning in recent years, but COVID-19 has definitely sped up their schedule. There is no doubt that cheating is a dangerous phenomenon and disgraceful behavior. Exam cheating is a concern in the educational industry. For this purpose, we focus on automatic cheating detection in exams, as many teachers and educators complain about the spread of cheating and failure of detection methods. Cheating, in fact, has begun to spread not only at the university level, but also at the secondary and primary levels. Action recognition in videos has been a fruitful topic in computer vision in recent years. Its significance is demonstrated in many diverse applications, including remote sensing applications, video surveillance, video recovery, human–computer interactions, sports video analysis, home intelligence, and feature extraction. Action recognition is a challenging field due to the inherent noisy nature of interpretations captured by sensors,

which are frequently subject to viewpoint occlusion, scaling, illumination, cluttered background, camera motion, variation, and brightness. The importance of action recognition is substantiated in machine learning and data mining applications through the use of eligible metrics for choosing features and structure in these applications. The action recognition task is usually classified into two main categories: long-range recognition and short-range recognition. The former, long-range recognition, focuses on videos that span more than a minute. From this, it infers the future action based on the current action. The latter, short-range recognition, focuses on short-duration video sequences that consist of just a few seconds, such as video sequences in MSR DailyActivity and MSR-II [1]. The objective of this work is to infer the current action labels founded upon temporally unfinished video sequences. In this work, we present a comprehensive framework to detect and classify the strange actions and behaviors that occur in exam halls and lead to cheating. This is achieved by examining the exam by video and observing the students through the camera. The acquired model is optimized through renowned feature extraction. Another main contribution of this study is presenting a novel dataset on exam cheating. We generated and compiled the dataset ourselves because there is no open source dataset for identifying cheating in paper tests. The dataset was created to depict actions that students could take during a paper-based exam to allow them to cheat. It includes the most common cheating methods, such as exchanging exam papers, looking at another student's exam paper, using a cheat sheet, using a cellular device, and not cheating. The following is the order in which the manuscript was written. The sections "Introduction" and "Related Works" contain the introduction and literature review, respectively. The detailed description of the dataset and how the dataset was acquired is explained in Section 3. The key terms and the feature extractions of the proposed method are discussed in Section 4. Section 5 introduces the results and discusses the experiments in detail. Finally, the conclusion and an outlook are presented in Section 6.

2. Related Works

The significance of recognizing a human action from a video containing a complete action execution is dramatically increasing. The basic steps of action recognition are the preprocessing of raw data, feature extraction and training, and classification [2]. The work in [3] presented a survey of popular algorithms, existing models, popular action databases, technical difficulties, and evolution protocols for action recognition and prediction from videos, which represent the mainstay for real-world applications such as autonomous driving vehicles, video retrievals, etc. Deep learning algorithms and sensors embedded within smartphones and smartwatches were exploited in [4] to recognize eight human activities such as walking, jogging, sitting in a car, etc. The results of the study showed that a combination of data from wrist and pocket sensors can be used to accurately recognize many human activities. In [5], the authors developed techniques to control home appliances using multimodal interaction such as speech, gestures, and smartphone applications. The accuracy of control home appliances using gesture action was 79.25%. For few-shot action recognition, the researchers in [6] suggested a temporal-relation cross-transformation novel approach (TRX). The contribution was the construction of class prototypes using the CrossTransformer attention mechanism. The method proposed by [7] utilizes convolutional neural networks paired with temporal layers for video sequence classification tasks. The researchers in [8] introduced the Action for Cooking Eggs dataset (ACE). The ACE dataset contains activities that occurred in a kitchen, and action label and action recognition methods for analyzing scene contexts were provided for each frame. The use of Kinect devices improves the effectiveness of the application with an in-depth video for intelligent monitoring.

Image processing is still in its infancy, and requires many manual inputs to provide computers with the instructions they need to access the result. These computers were programmed to recognize images [9]. Many studies concentrate on tackling cheating action recognition and all aspects related to it [10]. Ref. [11] organized eight online

exam control procedures to detect cheating without employing human proctors or robotic proctors. The essential reasons for cheating actions were investigated by [12]. They found that the most influential factors are the papers exchanged and the environment in which the exam was held. However, Ref. [13] realized the danger of online exams with the tremendous development of technology, allowing for students to master cheating. Weka is used as a tool to identify student behavior that can be classified as cyber-cheating. Ref. [14] introduced computational methods involving a support vector machine (SVM) and text-mining to detect plagiarism. The used computational methods succeed with an accuracy and precision above 90% in determining the original author of the submitted document. Data-mining algorithms, hierarchical clustering, and dendrogram trees have been used to detect patterns in multiple-choice online exam responses that indicate cheating during an exam [15]. Human proctoring is the most prevalent methodology to control cheating in exams. The authors in [16] presented a multimedia analytical system for online exam proctoring. The system is composed of two inexpensive cameras and one microphone. The system's results hinted at future robust behavior-recognition educational applications. The work developed by [17] offered a system that functioned by capturing the data regarding head pose estimates and eye gaze using an internet connection and webcam. The visual focus of attention system (VFOA) was implemented using a hybrid classifier approach and machine learning to classify the students' actions as either malpractice or a momentary lapse in concentration. The COVID-19 pandemic imposed a rapidly invented system to prevent fraud during remote online exams [18]. This took advantage of CNN-based technologies and a new method to provide software that guaranteed more protection during e-exams. This technology was used during the COVID-19 pandemic and was recommended by the majority of governments around the world. Ref. [19] collected sensor data from the iPhone 7's accelerometer and gyroscope during movements, and machine learning was suggested as a candidate for detecting cheat behaviors in physical activities. The work offered by [20] proposed a framework based on deep learning to distinguish suspicious activities during exams held at halls. The proposed model was tested using the CIFAR-100 dataset. The developed system in [21] utilized 3D convolutional neural networks (3D CNN) for image recognition and processing. The system aims to monitor movements and gestures during exams. A recent study [22] reviewed 58 publications about online exams published from 2010 to 2021. The comprehensive review is a very useful resource to obtain an understanding of cheating mitigation, detection, and prevention for educators and academic workers. In the literature, the objectives for preventing and detecting cheating varied, including: (1) strengthening the morality and ethics of students; (2) limiting the possibilities of cheating, e.g., by assessment environment design optimization; and (3) detecting the students caught cheating. However, such approaches are time-consuming and expensive. To fill the gaps in the literature, this study proposes a new framework for the early detection of students' cheating practiced on exams.

3. Data Preparation and Acquisition

One of the main contribution of this study is providing a dataset that will soon be available for public use. Since there is no open source dataset related to detecting cheating in paper exams, we designed and prepared the dataset ourselves. We designed the dataset to contain actions that students may perform during the paper-based exam that will enable them to cheat. It covers most cheating techniques, including: exchanging exam papers, looking at another student's exam paper, using cheat sheets, using cellular devices, and not cheating. Figure 1 depicts several activity classes. A Canon 70D sensor camera was used to capture scenes. The scenes were captured in a classroom in the information technology faculty at the Hashemite University. The sensor recorded 24 frames per second, and the image size was 1920 $\times$ 1080 pixels. This period is very appropriate to determine the actions and not to ignore any movement, even if it is simple. The Canon sensor also captured the hand area of a subject. The distance between the sensor and the recorded scene was approximately 3 m. Video clips were grouped into five action types, as shown in Figure 1.

The presented dataset is a challenging one, as many activities appear very to be similar and offer actions that do not depend only on the movement of the body. For example, additional information such as "using cheat sheet" or "use of cellular device" should be taken into account to make a final decision on action recognition. Therefore, it is important to focus not only on the movement of the body but also the adjacent objects. Our dataset consisted of five classes. The total number of video sequences was 37, and the average number of images in each class was 1650. Table 1 shows the number of sequences and frames per class. For action recognition, not all frames are equally crucial; only a few are critical. Therefore, we asked annotators to select a subset of 300 images from each class such that they best depict the class. Overall, we recorded eight unique subjects: four female students and four male students. Each pair of subjects conducted five distinct cheating activities, that is, 1000 images for training were available for each class. In addition, 500 images were also captured as testing images for each class.

Table 1. Details of the actions of the student cheating dataset.

Action	No. of Sequences	No. of Frames
1 Use of cellular device	13	3192
2 Exchange exam paper	4	744
3 looking at another student's exam paper	8	1734
4 Using cheats sheet	8	1626
5 Not cheating	14	954

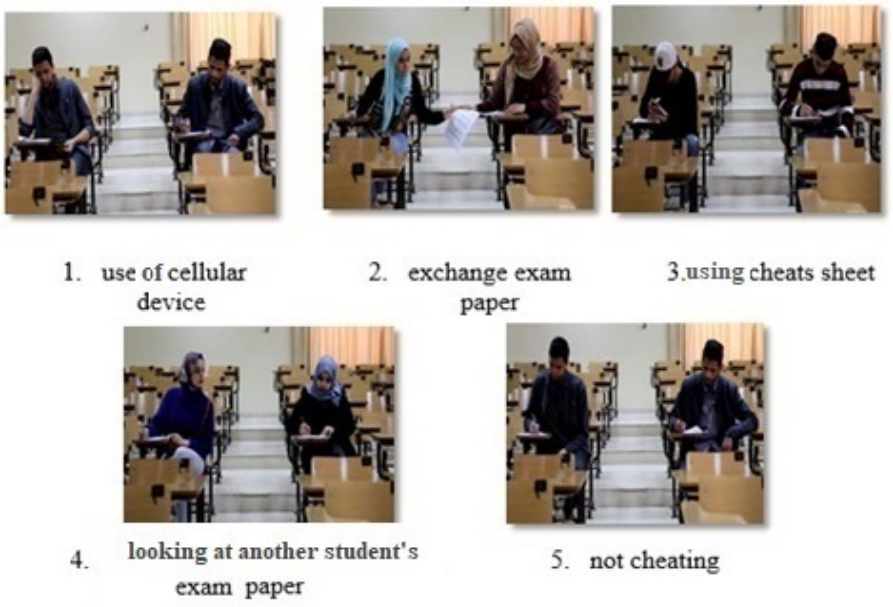

Figure 1. Example shots of each class.

Our task is to classify five kinds of exam cheating actions at frame-level, including: exchanging exam papers, looking at another student's exam paper, using cheat sheets, using cellular devices, and not cheating. It is an attractive dataset since most of the classes involve human–object interaction and share the same body movements.

4. Proposed Method

The proposed work is being developed for a computer vision-based system. The goal of this work is to create a multimedia analysis system that can detect and classify various actions indicative of cheating during an exam. The model includes scaling all of the frames in the dataset and the extraction of five renowned features. For each type of feature, a visual vocabulary codebook is created with different-sized words to encode the visual occurrences

in each frame. Finally, a support vector machine is used to classify the specified features. The proposed approach proves its effectiveness using the proposed dataset.

4.1. Definition of Key Terms

In our research, we want to infer the class label y for each frame in the video. More formally, a video V is represented by a set of frames $V = x_1, x_2, \ldots, x_T$, where x_t is an element from some input domain X (e.g., a video frame) and T is the length of a video sequence. Suppose we are given set of N samples (x_i, y_i), $i = 1, \ldots, N$, such that x_i is the feature vector of the i-th sample and y_i is its class label that falls from some discrete set of classes Y. The task is to produce a function F (classifier) that will work well on unseen samples. Mathematically, the frame label y is selected to maximize the scoring function F:

$$Y = argmax_y F(V) \tag{1}$$

Here, in Equation (1), let V represent the space of all possible inputs and y represent the set of identifiable actions such as "using cheats sheet", "use of cellular device", "no cheating", etc. $F(V)$ is a function that measures how well a sequence is presented. The task is to assign a class label Y at frame level. At test time, the maximizer function F : X->Y assigns a predictive label to the real vector space x. To find F, we used a multiclass Support Vector Machine (SVM) classifier [23]. This kind of classification is used in many action recognition applications. The formulation to solve multi-class SVM can be carried out by building (assuming Y classes) $Y(Y-1)/2$ multiple binary SVM classification problems. The objective of SVM is to learn the optimal separating hyperplane w, which can be found by:

$$argmin \frac{1}{2}||w||^2 + C \sum_{n=1}^{N} \xi_i \tag{2}$$

For each sample, one slack variable ξ_i is introduced to measure the loss of misclassification. The upper bound on the empirical risk is measured by the summation of the slack variables on the training set. For general purposes, a non-differentiable regularization parameter C is introduced to equilibrium the trade-off between complexity and loss. For example, we are given video sequences of "Exchange exam paper" and "no cheating"; each sequence is represented by frames that are considered our measurements and we want to correctly classify an unseen frame as either of these two classes. Each frame is digitized as 1920 $\times$ 1080 pixels, so we have measurement vectors $\xi_i \in R_d$, where d = 2,073,600. The positive label could indicate the "Exchange exam paper" class, and the negative label may indicate the "no cheating" class. Then, a new frame is given, which we want to classify: is it an "Exchange exam paper" or a "no cheating"?

4.2. Feature Extraction

Feature extraction is a kind of dimensionality reduction that professionally identifies informative parts of an image as a compressed feature vector. It is recommended to adapt this technique to large images to reduce processing time during tasks such as image retrieval and matching. In our experiments, to evaluate the effectiveness of the proposed method, we extracted five well-known features that are described as follows:

- BRISK: For each frame, we extracted the Binary Robust Invariant Scalable Key-points (BRISK) multi-scale corner features [24]. BRISK is a scale-invariant and rotation-invariant feature point detection and description technique. The BRISK features contain information about points and objects detected in a 2D gray-scale input image. An example of the detected key-points in the "use a cellular device" class is shown in Figure 2. Brisk accomplishes rotation in-variance by attempting to rotate the sample pattern by the measured orientation of the key-points. For clarity, the radials of the circles represent the orientation of the detected key-points while their size represents their scale. In our experiments, to extract BRISK features, we set the scale to 12 and

specified the minimum accepted quality of corners as 10% within the designated region of interest (rectangular region for the detected corner). The minimum accepted quality of corners denotes a fraction of the maximum corner measured value in the frame. Note that increasing this value will remove inaccurate corners.

- MSER: We extracted MSER features from the proposed dataset. The maximally stable extremal regions (MSER) technique was used to extract co-variant regions from images [25]. The word "extremal" means that all pixels within a certain region have a higher or lower intensity (brightness) than those outside their boundaries. This process is achieved by arranging the pixels in ascending order according to their intensity and then assigning pixels to regions. The region boundaries were specified by applying a series of thresholds, one for each gray-scale level. Almost all the producing regions resembled an ellipse shape. The resulting region descriptors are considered MSER features. For parameters, we set the step size between intensity threshold levels at 2. Increasing this value will return fewer regions. We also considered the vector [30, 14,000] for the size of the region in pixels. The vector $[minimum_areamaximum_area]$ allows for the selection of regions whose total pixels are within the vector. An example of the detected keypoints in the "exchange exam paper" class is shown in Figure 2. It depicts MSER regions, which are designated by pixel lists and are kept in the regions object. Figure 2 displays centroids and ellipses that fit into the MSER regions.
- HOG: The Histogram of Oriented Gradient is one of the most famous feature-extraction algorithms for object detection, proposed by [26]. It extracts features from a region of interest in the frame or from all locations in the frame. The shape of objects in the region is captured by collecting information about gradients. The image is divided into cells, and each group (grid) of adjacent cells forms spatial regions called blocks. The block is the foundation for the normalization and grouping of histograms. The cell is represented by angular bins according to the gradient orientation. Each pixel in the cell participates in a weighted gradient to its corresponding bin; this means that each cell's pixel polls for a gradient bin with a vote proportional to the gradient amount at that pixel (e.g., if a pixel has a gradient orientation of 85 degrees, it will poll with a weighted gradient of 0.9 for the 85-to-95 degree bin and a weighted gradient of 0.9 for the 75-to-85 degree bin). In the experiments, we extract HOG features from blocks specified by $[16, 16]$ cells and 9 orientation histogram bins to encode finer orientation details. However, an increasing number of bins increases the length of the feature vector, which then requires more time to access. A close-up of a HOG detection example is shown in Figure 2.
- SURF: Speeded-Up Robust Features (SURF) is a detector–descriptor scheme used in the fields of computer vision and image analysis [27]. The SURF detector finds distinctive interest points in the image (blobs, T-junctions, corners) based on the Hessian detector. The idea behind the Hessian detector is that it searches for strong derivatives in two orthogonal directions, thereby reducing the computational time. The Hessian detector also uses a multiple-scale iterative algorithm to localize the interest points. The SURF descriptor recaps the pixel information within a local neighborhood called "block". The block calculates directional derivatives of the frame's intensity. The SURF descriptor describes features unrelated to the positioning of the camera or the objects [28]. This rotational in-variance property allows for the objects to be accurately identified regardless of their perspectives or their different locations within the frame. The region of interest (ROI) is presented as a vector with the form [x y width height]. As parameters, we set the region size to $[1\ 1 size(\mathrm{I},\ 2)\ size(I, 1)]$, where the $[1\ 1]$ elements specify the left upper corner of the rectangular region of size $[size(I, 2) size(I, 1)]$. An example of the ROIs in the "using cheat sheet" class is shown in Figure 2.
- SURF&HOG: We used two of the aforementioned features, SURF and HOG, in the extraction process [29]. First, we used the SURF detector to obtain objects that contain information about the interest points in the images. We created a regular-spaced grid

of interest point locations over each image. This permitted dense feature extraction. Then, we computed the HOG descriptors centered on the point locations produced by the SURF detector. For clear visualization, we selected 100 points with the strongest metrics. Figure 2 shows the SURF interest points and the HOG descriptors in the "using cheat sheet" class. Bulleted lists look like this:

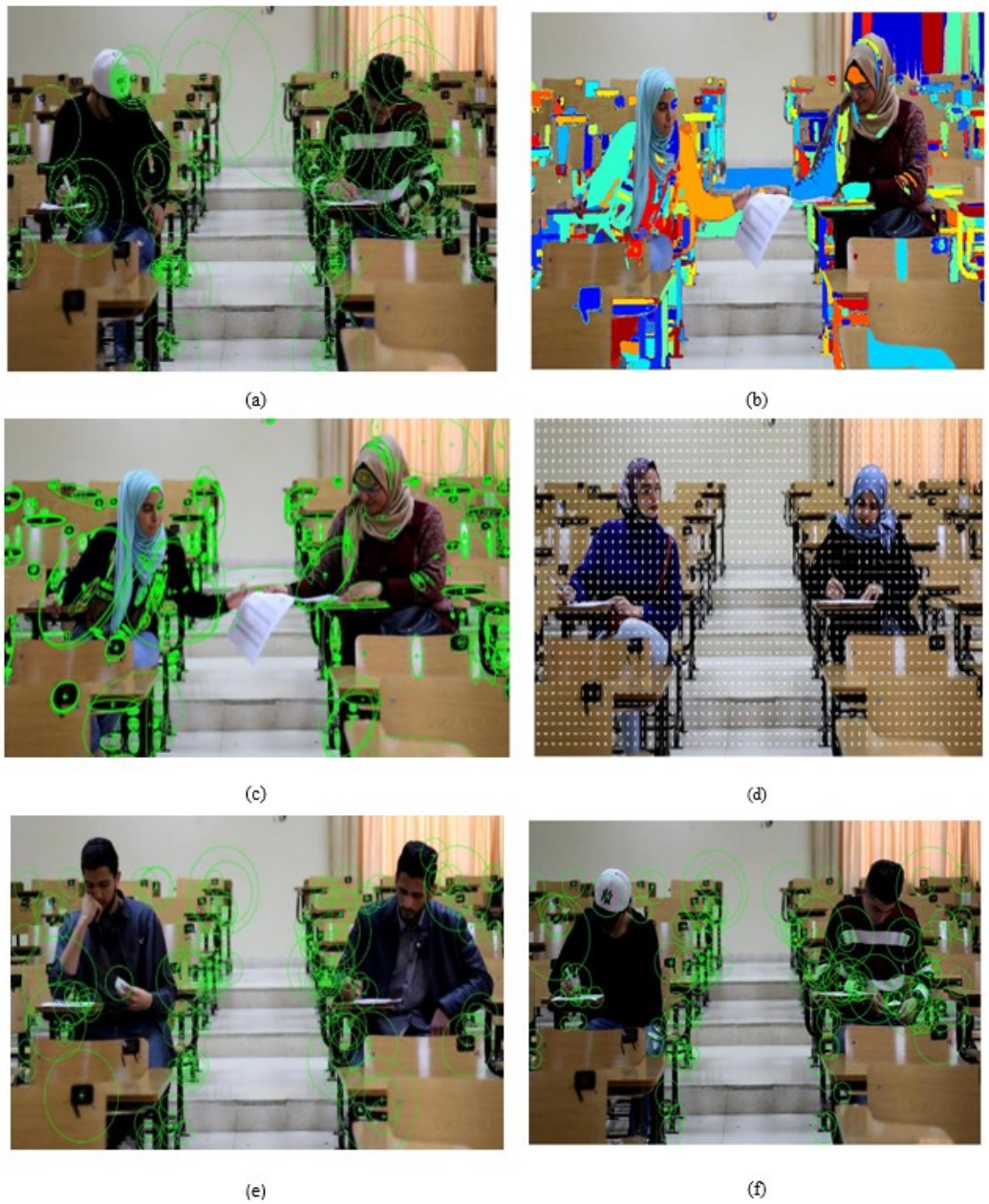

Figure 2. Here some sampling patterns of the (**a**) BRISK features, (**b**) MSER regions, (**c**) MSER ellipses and centroids, (**d**) HOG blocks around the strongest corners, (**e**) SURF locations of interest, and (**f**) SURF detectors and HOG descriptors.

5. Experiments

To evaluate the performance of the proposed method, we conducted experiments on action recognition tasks in the proposed dataset at the frame level using five kinds of well-known features. The dataset was made up of a set of short video sequences representing exam cheating actions. The dataset included a total of 37 sequences at a resolution of 1920 × 1080 pixels. Due to some issues with feature extraction in terms of the high dimensionality of the feature vector, we cropped the frames from the sides

to 960 × 540 pixels without affecting their contents or affecting the main objective of the classification task. We prepared training and validation frame sets. Since the frame sets contained an unequal number of frames per action, we adjusted this so that the number of frames in the training set was balanced. Note that each action set has exactly the same number of images. We separated the frames of classes into training and validation data. We chose 30% of the frames from each class for the training data and the remainder and 70%, for the validation data, and randomized the fragments to avoid biasing the results. Note that this ratio is not easy and is a challenge in the field of classification.

For each of the features listed in Section 4.2, we created a visual vocabulary code-book by using the bag of words technique. Bag of words (BOW) is a natural language processing technique adapted to computer vision. Additionally, the bag of words technique offers an encoded method to count the visual vocabulary occurrences in an image. BOW produces a histogram that becomes a reduced representation of an image. The vocabularies are constructed by reducing the number of features through a quantization of feature space using K-means clustering. In our experiments, to establish the code-book, an unsupervised learning clustering K-mean is used with $k = 400, 500, 600, 700$, where the clusters' centers are characterized as the video's vocabulary.

Tables 2 and 3 show the classification performance of the validation data for each class, with different types of features and different values for vocabulary. On the one hand, in these experiments, we used multiple vocabulary (k) values for each type of feature. It is good to note that the change in the number of vocabulary significantly affected the classification performance. Perhaps there are other vocabulary values that may increase accuracy, but this is beyond the scope of this research. In short, the experiments perform best in this classification task by leveraging SURF descriptors when the vocabulary size was 500. Additionally, Figure 3 displays a comparison of visual word occurrences using $k = 400$ and $k = 500$. On the other hand, based on the classification performance, we can categorize the accuracy of the cheat classes into four categories, and illustrate the results in Tables 2 and 3.

Table 2. Accuracy of classifying the validation dataset using BRISK and HOG features with multiple values for vocabularies.

Features	**BRISK**				**HOG**			
Vocabulary	**400**	**500**	**600**	**700**	**400**	**500**	**600**	**700**
1 Use of cellular device	65%	86%	69%	69%	80%	69%	51%	67%
2 Exchange exam paper	63%	84%	86%	92%	69%	94%	86%	88%
3 looking at another student's exam paper	57%	73%	78%	94%	75%	80%	92%	94%
4 Using cheats sheet	84%	55%	84%	80%	80%	80%	82%	88%
5 Not cheating	100%	98%	100%	96%	98%	86%	98%	100%
Average Accuracy	74%	79%	84%	86%	80%	82%	82%	87%

Table 3. Accuracy of classifying the validation dataset using MSER, SURF, and SURF&HOG features with multiple values for vocabularies.

Features	**MSER**				**SURF**				**SURF & HOG**			
Vocabulary	**400**	**500**	**600**	**700**	**400**	**500**	**600**	**700**	**400**	**500**	**600**	**700**
1 Use of cellular device	75%	73%	73%	92%	65%	90%	82%	69%	61%	75%	69%	67%
2 Exchange exam paper	82%	98%	94%	96%	75%	96%	73%	75%	92%	84%	94%	86%
3 looking at another student's exam paper	98%	78%	78%	84%	75%	86%	86%	82%	67%	98%	94%	92%
4 Using cheats sheet	94%	67%	78%	80%	73%	82%	94%	78%	82%	82%	90%	100%
5 Not cheating	98%	96%	100%	76%	94%	98%	100%	90%	100%	100%	96%	76%
Average Accuracy	89%	82%	85%	86%	76%	91%	87%	79%	80%	88%	89%	84%

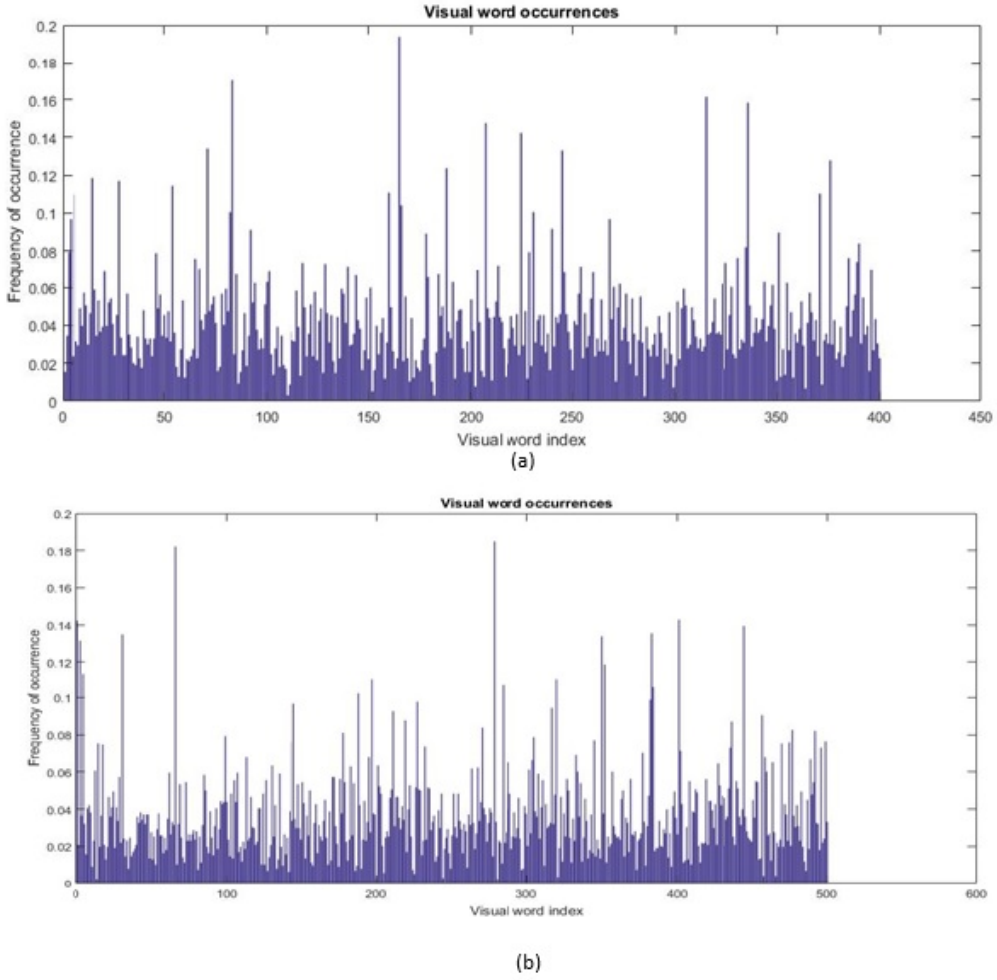

Figure 3. The comparison of visual word occurrences using SURF features at k = 400 (**a**) and k = 500 (**b**).

First, for the classification of the (looking at another student's exam paper, using cheat sheet) classes, the accuracy ranged from 86% to 98% for the "look at the student paper", and 84% to 100% for the "use cheat cheat" class. This is because the classes contain extremely varied kinds of cheating, which lead to huge variations in the feature space.

Second, for the classification of the "exchange exam paper," the accuracy ranged from 92% to 98%. The classifier maintained high accuracy despite choosing varying vocabulary values from different features. These accuracy values are considered reasonably high and are welcome in the classification world. Typically, the "exchange exam paper" class is triggered by specific object interactions in specific scene settings. As a result, it must include not only actions but also the interpretation of objects, situations, and their temporal arrangements with actions, as this knowledge might provide a valuable indication as to "what's going on now".

Third, when classifying the "using a cellular device" class, the accuracy varied between 75% and 92%. The results were not encouraging, and this could be for several reasons, including using a phone of a dark color, the same color as men's clothing; phones are also different shapes and sizes, which requires the system to be trained enough to be able to distinguish and classify them.

Fourth, in the classification of the "not cheating" class, we note that all the selected features were able to classify frames with a very encouraging accuracy of 100%. This is expected: classifying a class that contains very simple movements without interacting with objects is considered a difficult task in the classification process. From this, we conclude

that the results are better for the classification of non-cheating than for the classification of cheating.

Figure 4 highlights the best results. The results were achieved with different features. Note that choosing various features does not significantly reduce the recognition performance. Given the results shown in Figure 4, we were looking at the types of features from which the classifier was able to infer the best results. The results obtained from BRISK and HOG features were reasonable. For the BRISK features, the best was 94%, for the "looking at another student's exam paper" class, and the lowest was 69%, for the "use cellular device" class. For the HOG features, the best was 100%, for the "not cheating" class, while the lowest was, again, 67% for the "use cellular device" class. The average accuracy when classifying the validation dataset was 86% and 87% for BRISK and HOG, respectively. There may be a good opportunity to improve these results by increasing the number of detected keypoints in the descriptors, combining the BRISK and HOG descriptors with other detectors, or just tuning some of the parameters. There were encouraging results when using the MSER and SURF and HOG features. An identical average accuracy of 89% was obtained from both features. The MSER features distinguished "looking at another student's exam paper" and "not cheating" with an accuracy of 98%, and "use cheat sheet" with an accuracy of 94%. This is not strange, because the detected regions are well-defined by the intensity function. This leads to the regions having many key properties that make them valuable. Additionally, the significant results obtained by SURF and HOG features for the classification of "no cheating", "looking at another student's exam paper" and "exchange exam paper" cannot be avoided. HOG demonstrated its positive effects in detecting texture information and the edge of the image. However, SURF is the fastest, and comparable to SIFT in terms of performance.

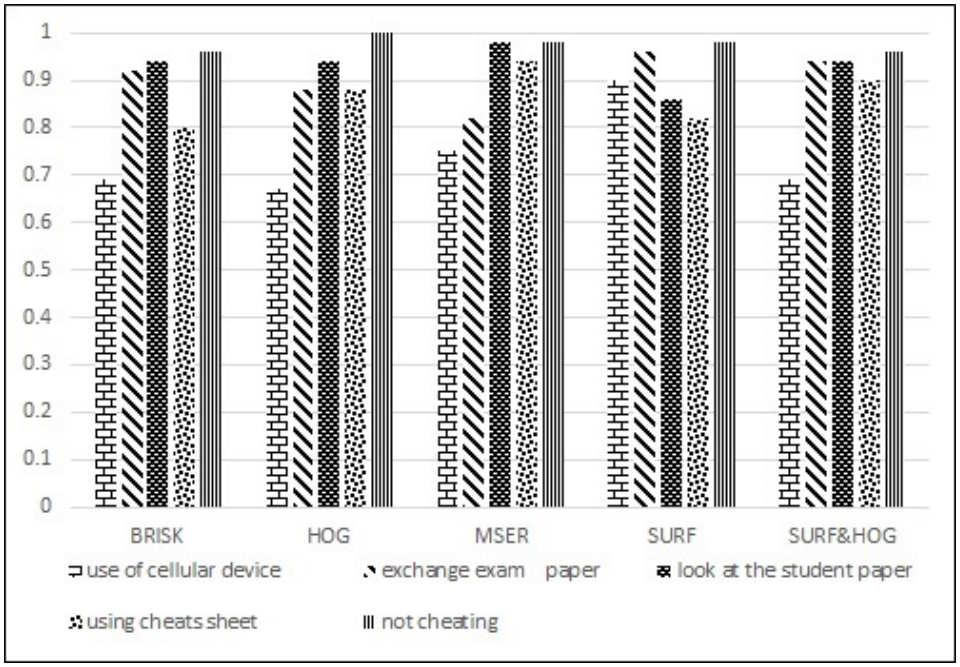

Figure 4. The accuracy obtained when classifying the validation dataset using different features.

Typically, the results obtained from the SURF features are remarkable. The average accuracy was 91%; see Table 4. It had the distinct ability to distinguish between the features of "using a cellular device" with an accuracy of up to 90%. This notable accuracy could not be reached by the other features most of the time. The SURF technique is well-known for its quick computation of operators utilizing box filters. Figure 5 shows a comparison of the different correlations between the five cheating classes using SURF features.

Table 4. The average accuracy of classifying the validation dataset.

Features	Accuracy
BRISK	86%
HOG	87%
MSER	89%
SURF	91%
SURF&HOG	89%

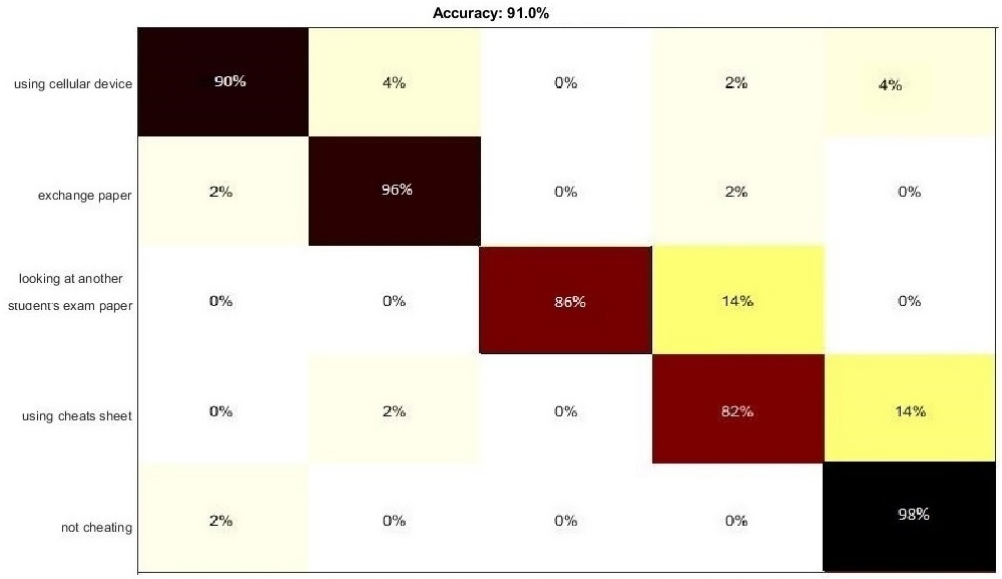

Figure 5. Comparing the accuracy of different correlations between the five cheating classes using SURF features.

6. Conclusions

In this research, we created a cheating video sequence dataset that detects cheating actions in paper-based exams. The dataset contains very challenging video sequences, since many activities appear to be quite similar and include actions that are not solely dependent on body movement. The results from the experiments on the framework were impressive and substantial. The cheating recognition model correctly recognized the cheating actions with an accuracy of 91%. As the results of the work were encouraging and distinct, there are several ways in which our work might be enhanced. For example, more complex algorithms could be used, such as deep learning for learning and more appropriate features and classifiers for classification. The system can also be expanded in the future to detect cheating in online exams with more than one subject. Moreover, the proposed dataset was captured in one country, and the examination environment is different in every country. Therefore, the dataset can be expanded by recording more videos and taking more dynamic factors such as: different environments; lighting (dim, normal, bright); camera angle (low angle, face-level, on-looking, top-down); presence of various motions; blurriness; resolution (SD, HD, 4K); etc.

Author Contributions: F.H. proposed the research framework, conceptualization, methodology, formal analysis and data curation. A.A.-A. worked on formal analysis and writing—original draft preparation. S.E.-S., E.A. and M.A.-H. worked on the writing—review, supervision and editing. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Informed consent was obtained from all individual participants included in this study.

Data Availability Statement: Data are available from the authors upon request.

Acknowledgments: The authors would like to thank the Hashemite University for its encouragement.

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

References

1. Oreifej, O.; Liu, Z. Hon4d: Histogram of oriented 4d normals for activity recognition from depth sequences. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, 23–28 June 2013; pp. 716–723.
2. Alshdaifat, E.; Alshdaifat, D.; Alsarhan, A.; Hussein, F.; El-Salhi, S.M.F.S. The effect of preprocessing techniques, applied to numeric features, on classification algorithms' performance. *Data* **2021**, *6*, 11. [CrossRef]
3. Kong, Y.; Fu, Y. Human action recognition and prediction: A survey. *Int. J. Comput. Vis.* **2022**, *130*, 1366–1401. [CrossRef]
4. Alam, A.; Das, A.; Tasjid, M.; Al Marouf, A. Leveraging Sensor Fusion and Sensor-Body Position for Activity Recognition for Wearable Mobile Technologies. *Int. J. Interact. Mob. Technol.* **2021**, *15*, 141–155. [CrossRef]
5. Fakhrurroja, H.; Machbub, C.; Prihatmanto, A.S. Multimodal Interaction System for Home Appliances Control. *Int. J. Interact. Mob. Technol.* **2020**, *14*, 44. [CrossRef]
6. Perrett, T.; Masullo, A.; Burghardt, T.; Mirmehdi, M.; Damen, D. Temporal-relational crosstransformers for few-shot action recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 475–484.
7. Fernando, B.; Gould, S. Learning end-to-end video classification with rank-pooling. In Proceedings of the International Conference on Machine Learning, PMLR, New York, NY, USA, 19–24 June 2016; pp. 1187–1196.
8. Shimada, A.; Kondo, K.; Deguchi, D.; Morin, G.; Stern, H. Kitchen scene context based gesture recognition: A contest in ICPR2012. In *International Workshop on Depth Image Analysis and Applications*; Springer: Berlin/Heidelberg, Germany, 2012; pp. 168–185.
9. Hussein, F.; Piccardi, M. V-JAUNE: A framework for joint action recognition and video summarization. *ACM Trans. Multimed. Comput. Commun. Appl. (TOMM)* **2017**, *13*, 1–19. [CrossRef]
10. Liu, X.; Li, Y.; Li, Y.; Yu, S.; Tian, C. The study on human action recognition with depth video for intelligent monitoring. In Proceedings of the 2019 Chinese Control And Decision Conference (CCDC), Nanchang, China, 3–5 June 2019; pp. 5702–5706.
11. Cluskey, G., Jr.; Ehlen, C.R.; Raiborn, M.H. Thwarting online exam cheating without proctor supervision. *J. Acad. Bus. Ethics* **2011**, *4*, 1–7.
12. Wang, J.; Tong, Y.; Ling, M.; Zhang, A.; Hao, L.; Li, X. Analysis on test cheating and its solutions based on extenics and information technology. *Procedia Comput. Sci.* **2015**, *55*, 1009–1014. [CrossRef]
13. Hernándeza, J.A.; Ochoab, A.; Muñozd, J.; Burlaka, G. Detecting cheats in online student assessments using Data Mining. In Proceedings of the Conference on Data Mining | DMIN, Las Vegas, NV, USA, 26–29 June 2006; Volume 6, p. 205.
14. Diederich, J. Computational methods to detect plagiarism in assessment. In Proceedings of the 2006 7th International Conference on Information Technology Based Higher Education and Training, Ultimo, Australia, 10–13 July 2006; pp. 147–154.
15. Chen, M. Detect multiple choice exam cheating pattern by applying multivariate statistics. In Proceedings of the International Conference on Industrial Engineering and Operations Management, Bogota, Colombia, 25–26 October 2017; Volume 2017, pp. 173–181.
16. Atoum, Y.; Chen, L.; Liu, A.X.; Hsu, S.D.; Liu, X. Automated online exam proctoring. *IEEE Trans. Multimed.* **2017**, *19*, 1609–1624. [CrossRef]
17. Indi, C.S.; Pritham, K.; Acharya, V.; Prakasha, K. Detection of Malpractice in E-exams by Head Pose and Gaze Estimation. *Int. J. Emerg. Technol. Learn.* **2021**, *16*, 47. [CrossRef]
18. Sharma, N.K.; Gautam, D.K.; Rathore, S.; Khan, M. CNN implementation for detect cheating in online exams during COVID-19 pandemic: A CVRU perspective. *Mater. Today Proc.* **2021**. [CrossRef]
19. Kock, E.; Sarwari, Y.; Russo, N.; Johnsson, M. Identifying cheating behaviour with machine learning. In Proceedings of the 2021 Swedish Artificial Intelligence Society Workshop (SAIS), Stockholm, Sweden, 14–15 June 2021; pp. 1–4.
20. Genemo, M.D. Suspicious activity recognition for monitoring cheating in exams. *Proc. Indian Natl. Sci. Acad.* **2022**, *88*, 1–10. [CrossRef]
21. El Kohli, S.; Jannaj, Y.; Maanan, M.; Rhinane, H. Deep Learning: New Approach for Detecting Scholar Exams Fraud. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2022**, *46*, 103–107. [CrossRef]
22. Noorbehbahani, F.; Mohammadi, A.; Aminazadeh, M. A systematic review of research on cheating in online exams from 2010 to 2021. *Educ. Inf. Technol.* **2022**, *27*, 8413–8460. [CrossRef] [PubMed]
23. Hsu, C.W.; Lin, C.J. A comparison of methods for multiclass support vector machines. *IEEE Trans. Neural Netw.* **2002**, *13*, 415–425. [PubMed]
24. Leutenegger, S.; Chli, M.; Siegwart, R.Y. BRISK: Binary robust invariant scalable keypoints. In Proceedings of the 2011 International Conference on Computer Vision, Barcelona, Spain, 6–13 November 2011; pp. 2548–2555.

25. Matas, J.; Chum, O.; Urban, M.; Pajdla, T. Robust wide-baseline stereo from maximally stable extremal regions. *Image Vis. Comput.* **2004**, *22*, 761–767. [CrossRef]
26. Dalal, N.; Triggs, B. Histograms of oriented gradients for human detection. In Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), San Diego, CA, USA, 20–25 June 2005; Volume 1, pp. 886–893.
27. Bay, H.; Tuytelaars, T.; Gool, L.V. Surf: Speeded up robust features. In *European Conference on Computer Vision*; Springer: Berlin/Heidelberg, Germany, 2006; pp. 404–417.
28. Jegham, I.; Khalifa, A.B.; Alouani, I.; Mahjoub, M.A. Safe driving: Driver action recognition using surf keypoints. In Proceedings of the 2018 30th International Conference on Microelectronics (ICM), Sousse, Tunisia, 16–19 December 2018; pp. 60–63.
29. Madan, R.; Agrawal, D.; Kowshik, S.; Maheshwari, H.; Agarwal, S.; Chakravarty, D. Traffic Sign Classification using Hybrid HOG-SURF Features and Convolutional Neural Networks. In Proceedings of the ICPRAM, Prague, Czech Republic, 19–21 February 2019; pp. 613–620.