GSAFE - A Novel Feature Extraction Framework for Child Abuse and Violent Crime Detection

Paul McBrien
School of Computing, Dublin City University, Dublin, Ireland
paul.mcbrien2@mail.dcu.ie

***Abstract*—Skyrocketing levels of child sexual abuse material (CSAM) are being shared online which demands new and more robust technologies that can recognise violent crime in video. Many recently proposed solutions to this problem rely on computationally expensive, end-to-end deep learning feature extractors that lack interpretability and transparency and have been shown to exhibit bias. The CSAM detection system NeuralHash which has been proposed by Apple Inc, has been demonstrated to be vulnerable to simple evasion tactics such as flipping an image horizontally. These issues lead to a lack of trust from regulatory and privacy experts which stunts progress in this area of research. To combat these issues, this paper introduces GSAFE - a novel feature extraction method which supplements the use of deep learning feature extraction by using principles from Euclidean geometry. It offers a lightweight solution to these problems which is logical and interpretable, resilient to many evasion attacks and can be customised for various user privacy and information security requirements. The theoretical foundations of GSAFE are experimentally tested by applying it to the area of weakly-supervised criminal anomaly detection for abusive scenarios. These experiments further demonstrate the weaknesses of traditional feature extractors, and show that even a simple configuration of GSAFE running on lightweight hardware can achieve considerable performance in the detection of crime in video, and outperforms the feature extraction speed of 3D-ResNet-152 by 74%.**

1. Introduction

Europol has collected more than 51 million unique images and videos of abuse material [1] - a staggering figure that highlights the need for improved CSAM detection methods. Automated detection of archived and previously undetected videos is becoming an increasingly sought after technology, with many companies seeking to remove such videos from their platforms and law enforcement seeking to catch those who illegally possess and distribute these videos. Given the sensitive nature of this image and video content, the more general approach of violent crime detection is an active area of public research. For these methods, finding the exact time that a crime occurs in video is extremely useful as it allows for much faster manual review by humans. The release of UCF-Crime [2] has provided a difficult challenge for automated detection technologies, and is consistently the benchmark dataset which these technologies struggle most with. Currently, the best performing models utilise a 'weakly-supervised' deep-learning approach where the training videos are only labelled based on whether they contain anomalies or not - leaving the neural network to learn how to localise these anomalies. Given the large scale of the UCF-Crime dataset, feature extraction networks are used to reduce the size of the video data so that it can be processed efficiently [3] [4]. While these feature extractors currently produce the best performance with weakly-supervised models, research has shown that these models can exhibit representation bias in relation to the scene of video, rather than relying on the human activity to discriminate human activity [5]. These feature extraction networks can have hundreds of millions of parameters, making them computationally inefficient for smaller machines. Since these models rely solely on neural networks to form feature representations, their implementation results in a lack of interpretability, transparency and explainability. Given the serious nature of crime and abuse detection, these shortcomings lead to a lack of trust in end-to-end neural network detection methods. Regulatory agencies and law enforcement require an understanding of how these models work and the reasons why they sometimes fail, while security experts demand that these technologies respect the user privacy of the millions of people who use online platforms where these technologies are implemented.

The key contributions of this paper are as follows:

- Further experimental evidence is presented that many of the end-to-end deep learning feature extractors exhibit scene representation bias, and are too computationally expensive to run without heavyweight hardware.
- A novel feature extraction framework is introduced called Geometric Spatiotemporal Action Feature Extraction (GSAFE) which uses a mathematical approach to recover interpretability and explainability unavailable with end-to-end deep learning feature extractors.
- Considerable performance is achieved on a subset of the challenging UCF-Crime dataset that contains physically abusive crime categories.
- A processing speed increase of 74% is achieved with GSAFE when compared to the state-of-the-art feature extractor 3D-ResNet using only a CPU architecture.
- A thorough theoretical evaluation is presented which demonstrates the ability for GSAFE to solve serious vulnerabilities with Apple's NeuralHash detection system.

2. Related Work

Video anomaly detection is used to identify the presence of an anomaly for all frames of the video. This is different to typical video classification, where the entire video is labelled as anomalous or normal, and no prediction is made about the time at which an anomaly occurs. Early work on these problems primarily began with the use of unsupervised methods to train one-class classifiers with non-anomalous videos only [6] [7]. The method of included tracking object motion in frames were computationally efficient, but much less effective in crowded scenes [8], [9], . Other approaches included the use of deep auto-encoders to learn the underlying probability distribution of latent representations of the data [10], or the use of LSTMs for the same purpose [11], [12]. With these methods, a higher reconstruction error is typical for anomalous inputs and is therefore used as a criterion for identifying anomalies. These models can be supplemented with the use of generative adversarial networks to distort outlying samples and enhance the inlying samples [13]. Despite some success with many of many of these unsupervised learning approaches, many were prone to over-fitting and struggled with complex environments.

Using a fully supervised approach requires an excessive labelling effort for large datasets. A weakly-supervised anomaly detection approach provides an improvement in the performance and a smaller labelling effort. Current SOTA performance has been achieved using these methods. Sultani et al. [2] proposed the use of Multiple Instance Learning (MIL) to model each video as a bag of segments, with the instances of these bags represented as the video frames [14]. This method ranks each bag of frames using an anomaly score and the bag with the highest anomaly score is chosen as the anomaly. The use of *top-k* MIL ranking loss has been attempted to avoid noise which leads to a normal segment being mistaken for the most abnormal one. This is achieved by choosing a number of highly anomalous events [15], or using data augmentation methods [16].

As the focus of this paper is on detecting crime and abusive content in video, topics discussed also include related work in the area of child sexual abuse material detection [17] [18] [19]. These works focus on the detection of abuse-related material using lower-dimensional representations extracted using various feature extraction methods. Struppek et al. have found that even advanced neural network based hashing approaches are not robust to deal with evasion tactics such as image transformations and are convinced that many current systems are not robust enough to evasion attacks.

Evaluation of the mechanisms used by image feature extractors has been carried out by Choi et al. in the area of scene bias mitigation [5]. They found that 3D CNN feature extractors like I3D [3] and 3D Resnet (R3D) [4] can produce features which are influenced by the background scene of a video. Since these feature extractors are a standard method of weakly-supervised crime detection in previous works, the results using these models have likely been affected by this. Choi et al. develop a method which reduces the extent of this scene bias in video, and conclude that more work is needed in this area. Their methods have not been introduced in recent anomaly detection works [15] [16].

3. Weakly-Supervised MIL Model

A set of weakly-labelled training videos $\mathcal{V} = \{(\mathbf{F}_i, y_i)\}_{i=1}^{|\mathcal{V}|}$ where $F_i \in \mathcal{F} \subset \mathbb{R}^{M\times N}$ is a feature pre-computed using a video feature extractor (such as I3D [3], R3D [20] or GSAFE introduced in this work). There are M video snippets of dimension N, with the video-level annotation for video $\mathcal{V}_i$ denoted as $y_i \in \{0, 1\}$ ($y_i = 1$ for F_i as the feature representation of an anomalous video and $y_i = 0$ otherwise).

The model is defined by $r_{\theta,\phi}(F) = f_\phi(s_\theta(F))$ and returns a M-dimensional feature in $[0, 1]^M$ which represents the classification of each of the M video snippets. The parameters θ and ϕ relate to the temporal feature extractor and snippet classifier outlined in Equation 1 (respectively),

$$\min_{\theta,\phi} \sum_{i,j=1}^{|\mathcal{V}|} \ell_s\left(s_\theta\left(F_i\right), \left(s_\theta\left(F_j\right)\right), y_i, y_j\right) + \ell_f\left(f_\phi\left(s_\theta\left(F_i\right)\right), y_i\right) \tag{1}$$

where $s_\theta : \mathcal{F} \to \mathcal{X}$ is the temporal feature extractor which outputs $\mathcal{X} \subset \mathbb{R}^{M\times N}$, $f_\phi : \mathcal{X} \to [0, 1]^M$ is the video snippet classifier for each of the M snippets, l_s represents the loss function that maximises the separability between the distributions of the top-k snippet features for both normal and anomalous videos, and l_f is the loss function which trains the video snippet classifier f_ϕ using the top-k snippet features for normal and anomalous videos.

A temporal snippet feature extracted from a video is defined as $X = s_\theta(F)$, with a total M snippet features. Each snippet feature in X of a video is represented as a row x_m in X. The row for an anomalous snippet is denoted as x_m^+ and the row for a normal snippet is denoted as x_m^-. An anomalous video contains μ snippets x_m^+ drawn from the set of all positive snippets and contains $M-\mu$ snippets $\mathbf{x}_m^-$ drawn from the set of all negative snippets [21].

The summed feature magnitude of the top-k snippets is defined as

$$g_k(X) = \max_{\Omega_k(X)\subseteq\{x_m\}_{m=1}^M} \sum_{x_m\in\Omega_k(X)} x_m \tag{2}$$

where Ω_k is a subset from $\{x_m\}_{m=1}^M$, $x_m \in [0, 1]$ and the cardinality of Ω_k is k. Therefore, $g_k(X) \in [0, k]$.

The loss function for training the model s_θ in Equation 1 is a function maximises the separability between anomalous and normal videos,

$$l_s = \max\left\{0, m - g_k\left(X^+\right) + g_k\left(X^-\right)\right\} \tag{3}$$

where m is a predefined margin (set to the value of k in this work), $X^+ = s_\theta\left(F^+\right)$ is an anomalous video feature and the same quantity with a '$-$' symbol indicates a video feature which belongs to a normal video).

The term l_f in Equation 1 can be defined as

$$l_f = \lambda_1 \sum_{m=1}^{M} \left(f_\phi(x_m) - f_\phi\left(x_{m-1}\right)\right)^2 + \lambda_2 \sum_{i=1}^{M} \left|f_\phi\left(x_m\right)\right| \tag{4}$$

with λ_1 as a weight for the $l_{smoothness}$ term and λ_2 as a weight for the $l_{sparsity}$ term. The smoothness term is used to encourage similar anomaly scores for neighbouring

snippets. Since neighbouring snippets are closely related in time, anomaly scores should vary in a continuous fashion between them. The sparsity term is introduced to model the fact that only a small number of video snippets actually contain an anomaly.

3.1. Image Frame Feature Extraction

The I3D feature extraction method uses 3D convolutional neural network architecture with RGB frames and Optical-Flow streams as inputs [3]. This produces an output feature of dimension of 32×1024 for the individual input streams. The 3D-Resnet (R3D) model involves uses a similar 3D convolutional approach in conjunction with the ResNet architectures that have proved to be successful in image classification tasks [22]. These architectures use shortcut connections between layers which allow information to bypass intermediary hidden layers in a network during back-propagation. This allows very deep networks to be trained efficiently. The 3D element of the R3D network is implemented using kernels of dimension $1 \times 1 \times 1$ for the first and last convolutional layer and one of dimension $3 \times 3 \times 3$ between those two layers [4]. Other variations exist which use modified parameters for convolutions, batch normalisation and rectified linear unit order.

Features of dimension 32×1048 are extracted from last global pooling layer of the pre-trained 'Inception V1' variant of the I3D network, which uses data from the Kinetics-400 dataset [23] for training. Features of dimension 32×2048 are extracted with the ResNet-152 variant of the R3D model, using a combined dataset containing the Kinetics-700 dataset and the Moments in Time dataset for pre-training [20]. Beyond the initial baseline comparison of I3D and R3D, the R3D feature extraction model is used for all experiments.

3.2. You Only Look Once (YOLO)

The YOLOv5 framework consists of four main sizes, with small, medium, large and extra large variants of the model, each with increasing computational requirements. The model is similar to YOLOv4, but is implemented in Python rather than C [24], and uses the Darknet CNN backbone [25] trained on the MS-COCO dataset [26]. The algorithm works by dividing an image into N grids of equal dimensions. Predictions of the probability of a particular object class, as well as the coordinates of the bounding box of an object are made for each cell. A technique called Non Maximal Suppression removes bounding box candidates which have low probability scores. A process is then applied to analyse the ratio of bounding box area intersections and unions, before boxes are combined to enclose an entire object. The models involve many convolutional layers, with fully connected layers at the end. They can be used to detect persons, vehicles and other objects in a single image or in the many frames of a video.

3.3. Datasets

The UCF-Crime dataset [2] is a large anomaly detection dataset which contains 128 hours of video consisting of 1900 untrimmed videos from indoor and outdoor surveillance cameras. The footage is complicated as multiple anomalies can occur discontinuously throughout the videos and cuts to differing camera footage can occur. There are 13 anomaly classes including Abuse, Arrest, Assault, Arson, Road Accident, Shoplifting, Stealing, Shooting, Robbery, Fighting, Vandalism, Explosion and Burglary.

A subset of the UCF-Crime dataset is used for the main evaluations in this paper as they contain videos of a physically abusive nature. The subset contains all videos from the Abuse, Arrest, Assault, Fighting and Robbery classes. Classification of violent crimes into distinct groups is a difficult and subjective process. However, within the resources of this work, it is proposed that this subset provides a reasonable representation of physically abusive videos. The dataset train test split used is available on GitLab [27].

3.4. Scene and Object Masking

The primary goal of this paper is to develop a novel feature extraction method which preserves some level of interpretability, named GSAFE (see Section 3.5). To allow for a thorough comparison of R3D and GSAFE, it is important to evaluate the mechanism of R3D feature extraction.

The R3D feature extraction method uses a deep neural network to reduce the dimensionality of the video to form a feature representation. This makes it difficult to understand the mechanism of the R3D method. These end-to-end neural network feature extractors have been found to exhibit representation bias [5]. Since these networks are typically used for feature extraction of UCF-Crime videos, there is a responsibility to investigate this bias within the context of this work. Overlooking this bias could lead to devastating consequences for victims where crimes are not detected in video.

To illustrate this using an example, suppose there is a feature extractor which is trained on the Kinetics-700 dataset. The extractor learns to represent the image as bins containing certain ranges of RGB values. For example, it may learn to represent the action of skiing as a feature associated with blue and white pixels, since the activity of filming skiing typically depends on snow and clear weather. This is an extreme example of the feature extractor learning to produce features which are not related to the performed human action. This is known as scene representation bias. In reality, the learned feature will likely be more complex, but features may still encode scene-related information. Similarly, in a

Figure 1. A feature extractor may learn to classify these images as the same action because they are visually similar, despite there being two distinct actions of skiing and swimming.

context of crime detection, the feature extractor may learn to form a representation of a dark alleyway. The model may learn to output higher anomaly scores on average for videos with these dark scenes. This may increase its performance on a dataset with many crimes recorded on CCTV cameras

situated in these alleyways. The consequences are very severe if this model later fails to identify a violent crime because it occurred in a well-lit, open area.

This bias can be empirically evaluated for the UCF-Crime dataset by masking the regions of a frame using the object detection data from the YOLO framework for each video frame. Three types of masks are applied to each frame which systematically remove information from the image. The ShowAll mask blackens all pixels except the rectangles which described the detected region of the people in an image. The ShowOne mask blackens all pixels in an image except for those of the rectangle which describe the detected region for a randomly selected person in an image which is stored in the YOLO output feature. Finally, the ShowNone mask blackens all pixels of the rectangles which describe the regions for all of the detected people in an image. This is illustrated in Figure 2. The R3D feature extraction method is used to produce features for each of the three schemes of masked video.

(a) ShowAll (b) ShowOne (c) ShowNone

Figure 2. An example of the masking schemes applied to a frame.

3.5. Geometric Spatiotemporal Action Feature Extraction (GSAFE)

GSAFE is a novel method introduced in this work to produce feature representations using the YOLO network output of an input video. These feature describe the spatial and temporal information related to persons in a video using quantities derived from Euclidean geometry. The features can be applied to a wide range of video detection problems (such as the application outlined in Section 3.6). The GSAFE geometric quantities are tailored to the problem of weakly-supervised anomaly detection by transforming them into suitable feature embeddings for the MIL model. These feature embeddings are created by combining the geometric quantities using a theoretical framework introduced in this work, which is empirically tested in Section 4.

The set of people detected in a frame is denoted by $\{p_n\}_{n=1}^{|N|}$, where p_n is a person in a frame and $|N|$ is the total number of people in a frame. Each person in frame can be represented by a set of quantities, $p_n = \{x_n, y_n, w_n, h_n\}$, where x_n and y_n are the pixel coordinates of a person, and the width and height dimensions of the bounding box for that person as predicted by the YOLO algorithm are denoted by w_n and h_n, respectively.

The total distance between each of the persons is calculated as the sum of the pairwise Euclidean distances between each of the people detected in a frame, and is defined as

$$D = \sum_{i,j=1}^{|N|} \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \tag{5}$$

where (x_i, y_i) and (x_j, y_j) are the coordinates of the centroid of the enclosing rectangle for a detected person. The average distance between the people in a frame $\bar{D}$ is denoted as

$$\begin{cases} \bar{D} = \frac{D}{|N|} & |N| \neq 0 \\ \bar{D} = 0 & |N| = 0 \end{cases} \tag{6}$$

The sum of the bounding box areas A predicted for each person in a frame is denoted by

$$A = \sum_{n=1}^{|N|} w_n \cdot h_n \tag{7}$$

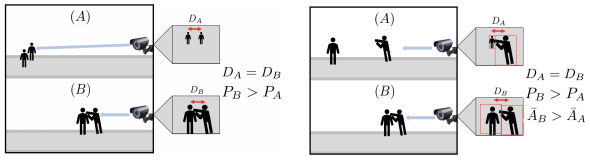
with the value for the average bounding box area $\bar{A}$ described by the same same function in Equation 6, replacing D with A.

The three geometric quantities N, D and A can then be used to form feature embeddings of a fixed size for input to the MIL model. This formulation uses a theoretical model of how these three geometric quantities relate to a probabilistic quantity P, with the relationship

$$P \propto \frac{1}{\bar{D}} \tag{8}$$

where P is the probability that a crime is occurring. This theoretical relationship relies on a simple assumption; given the close-quarters nature of the UCF-Crime data subset chosen, the probability of an image containing a crime will increase as the average distance between people decreases. Any function with a negative first derivative for all inputs may serve as a good model for the relationship between P and $\bar{D}$, but in this work an inverse square proportionality is assumed for simplicity.

Since the scale of the real-world environments will vary across the dataset videos, it is necessary to scale the value $\bar{D}$ by a scaling factor β. It is necessary to account for this since for different videos, the size of the people in image pixels of a frame is affected by many factors such as the distance from the camera to the people, the lens characteristics and how the camera is physically set up with respect to the scene.



(i) (ii)

Figure 3. For different video scenes, the scale of the humans can vary dramatically. In (i) and (ii), the image pixel distance between the persons in (A) and (B) is identical, but the real-world distance is much different.

To simplify the very complex nature of this scaling problem, the scaling factor β for $\bar{D}$ is chosen to be $\bar{A}^{-1}$, such that

$$\mathcal{G} = \frac{\bar{A}}{\bar{D}} \tag{9}$$

where $\mathcal{G}$ is a quantity which relates to the probability term P. These quantities can be parameterised in terms of ν, (where a video $\mathcal{V}$ has ν frames such that $\nu = \{\nu_i\}_{i=1}^{|\nu|}$) and also normalised to provide an approximation of P

$$P_{\nu_i} \sim \tilde{\mathcal{G}}_{\nu_i} = \frac{\mathcal{G}_{\nu_i} - \mathcal{G}_{min}}{\mathcal{G}_{max} - \mathcal{G}_{min}} \tag{10}$$

where P_{ν_i} is the probability that a crime is occurring in frame ν_i, and $\mathcal{G}_{min}$ and $\mathcal{G}_{max}$ are the minimum and maximum value of $\mathcal{G}$ across the set of all frames ν.

The value β is chosen to minimise the value of P_{ν_i} as $\bar{A} \to 0$, to account for the uncertainty in the distance between persons in a scene if they occupy fewer pixels in an image and are thus less resolved in an image. It is also assumed that the average heights and widths of people in an image are approximately equal (and have little variance). In reality, it is possible a tall person and short person at different distances from the camera may appear to be the same size.

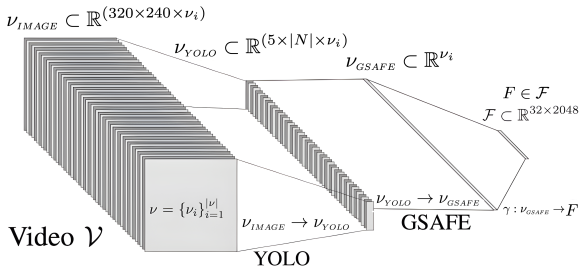


Figure 4. Structural overview of the GSAFE framework. For weakly-supervised model input, the input features are frames from video $\mathcal{V}$ which are reduced in dimensionality to produce the resultant feature F. The function γ is used to resample the data to the required size for use with the anomaly detection model.

Figure 4 illustrates the structure of the feature extraction process for a video $\mathcal{V}$. The collection of frames ν_{IMAGE} is composed of RGB images (with dimensions of 320×240 pixels) from a video $\mathcal{V}$. The YOLO algorithm extracts features for each person, producing a $5 \times |N|$ feature for each frame (a class ID and the four quantities in p_n for each person). Each of these YOLO features compose the collection of frame features ν_{YOLO}. GSAFE is then used to process the YOLO features for a particular frame into geometric quantities. These quantities are then transformed to create each value $\mathcal{G}(\nu_i)$. The collection of these single values across the video frames constitutes the feature ν_{GSAFE}.

To use GSAFE embeddings with the MIL model structure used in previous works, an extra feature processing step is used. A function γ is used to map the feature ν_{GSAFE} to a feature $\mathcal{F}$, which has dimensions $\mathbb{R}^{M\times N}$, where $M \times N$ is the dimension of the input layer of the MIL model. In the case of a video with more than $M \times N$ frames, γ will downsample the input feature, while upsampling a video with less than $M \times N$ frames. This can be achieved using a moving average function or interpolating function for γ, but in this work a Fourier method is used [28]. This involves transforming the values into the frequency domain to downsample or upsample. While this is a method usually applied to periodic data, it is assumed to be suitable for application to the weakly-supervised detection problem, since empirical evaluations are carried out to verify this assumption.

The evolution of $\mathcal{G}$ over the duration of a video is described in terms of ν, the frame number, to give $\mathcal{G}(\nu)$. This function is simply an approximation of the function $\mathcal{G}(t)$, which describes the temporal evolution of a real world scenario being captured in video format. The quantities D and A vary continuously as a function of t, and the quantity $|N|$ can be treated as a rational number rather than a natural number (e.g as a person walks out of a frame, only a fraction of their body remains in any frame). Therefore, it is assumed that the function $\mathcal{G}(t)$ and its frame-level approximation in ν sufficiently model the situations in the UCF-Crime data subset, and does not sufficiently model other types of situations (e.g shootings).

3.6. GSAFE for Child Sexual Abuse Material (CSAM) Detection

The empirical aspect of this work focuses on the publicly available dataset 'UCF-Crime' for evaluation of GSAFE when used for detecting criminal activity in video. In addition to this, a strong theoretical justification can be made for the application of GSAFE to CSAM detection.

Rather than scanning RGB images, comparison of image embeddings is the primary technology for CSAM detection [17]. Traditional functions used for creating these embeddings include standard image hashing [29] [30] and NeuralHash which has been introduced by Apple for scanning iCloud images [31]. These methods convert an image into a unique hash. Before the image is encrypted and uploaded, a client-side matching process compares the image hash against a locally available database of known CSAM hashes. This process requires that embeddings are relatively small so that they can be processed quickly and transmitted efficiently over networks. NeuralHash feature extraction method produces an abstract numerical representation of the image features, making diagnosis of classification failures difficult. As a neural network, the full mechanism of YOLO lacks interpretability. However, the output features and their resulting GSAFE embeddings are interpretable given that they represent physical objects detected in an image.

NeuralHash is more resilient to variations in image transcoding method, image resolution or colour filtering methods since it uses a convolutional neural network to extract features from an image instead of relying on exact pixel values. However, recent research has shown that the NeuralHash method is still vulnerable to simple transformations such as image rotation, flipping, resizing and centre cropping, as well as modifications to the compression method, brightness and contrast [19]. The GSAFE method is resistant to such detection evasion attacks.

Since the GSAFE method uses Euclidean distance between detected objects in an image, the embedding for any image will match that of the same image which has been rotated or flipped. This is mathematically provable by the fact that distance between points in a Euclidean space is a quantity that is invariant to rotations and reflections (also known as flips), since these transformations are isometries [32].

Since all distances are preserved during rotations and reflections of the image, the quantities A and D are preserved too, leading to the same values of $\mathcal{G}$ for all images in a video before and after such transformations. For scaling however, these values are not strictly preserved. A scaling factor α will scale all pixel distances in an image, resulting in different values of $\mathcal{G}$

$$A_{\text{SCALED}} = \alpha h \cdot \alpha w = \alpha^2 A \tag{11}$$

while the value D will be scaled by α, giving a new value for $\mathcal{G}$:

$$\mathcal{G}_{\text{SCALED}} = \frac{\alpha^2 A}{\alpha D} = \alpha \mathcal{G} \tag{12}$$

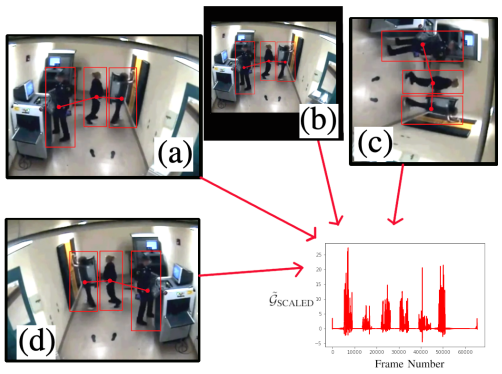


Figure 5. GSAFE applied to a particular frame. The image transformations of the original frame (a) are an image scale, an image rotation and horizontal image flip for frames (b), (c) and (d), respectively. All frames have the same GSAFE structure.

Therefore, the ratio of the scaled and unscaled GSAFE embeddings is a constant α, allowing for easy comparison with database embeddings. If the normalised value $\tilde{\mathcal{G}}$ is used to encode the GSAFE embedding, then both features will have the same values $\tilde{\mathcal{G}}_{\text{SCALED}}$=$\tilde{\mathcal{G}}$. This highlights the robust nature of the GSAFE method, which offers much greater resistance to detection evasion attacks.

The performance of GSAFE for CSAM detection may be impacted by more sophisticated video editing procedures, such as video trimming or removing frames periodically throughout a video. However, the structure of $\mathcal{G}(t)$ will still retain information. In this case, matching of GSAFE embeddings will require some level of error acceptance, which may lead to in increase in false positive matches. This has obvious negative impacts on expectations of user privacy, and as such, will require much further research. Nevertheless, this problem is not exclusive to GSAFE. Hashing functions suffer from a similar issue known as a hash collision. This is where hash functions can occasionally produce an image hash when an image that does not contain CSAM has the same hash output as a CSAM image [18].

Another vulnerability with hashing methods is that they still contain some information about the encoded image. NeuralHash uses 96 bits to encode hashes, while Microsoft's PhotoDNA uses hashes with 144-byte values [19]. Using larger embeddings increases the likelihood of information leak. In comparison, a GSAFE embedding uses 32 bits per image, with the possibility of using a half-precision floating-point format which uses 16 bits to store the image embedding.

Similar to a hash collision, the obvious case where the GSAFE method will produce a false positive is when the position and scales of people in one image are identical to those contained in a different image. For this reason, applying GSAFE to still images is likely to be an unviable option. However, the longer the duration of a video, the less likely that two differing videos will have exactly the same values for $\mathcal{G}$. This makes GSAFE more viable for video application.

For shorter videos and still images, GSAFE may still serve as an extra layer of security for both protecting user privacy and detection system function. In the case of user privacy preservation, using GSAFE as a auxiliary verification step will reduce the likelihood of a false positive detection. This extra verification step will harden the detection system to evasion attacks, as an attacker will need to invest resources targeting an extra detection system.

Very sophisticated attacks on NeuralHash include adversarial attacks which target the neural network architecture. Using the neural network gradient information, another neural network is trained to perturb the image pixels in a way that alters its likelihood of detection while the image itself remains visibly unaltered to humans. These attacks are thoroughly investigated on hashing functions by Struppek et al. [19]. GSAFE is also vulnerable to these attacks since it uses the the YOLO neural network for feature extraction. However, Struppek et al. explain that the modified images generated by an adversarial network can be used to augment the training data for detection networks like YOLO, which would make GSAFE more resistant to these attacks.

More robust GSAFE embeddings can be produced by increasing the complexity of the output features. This can be achieved by changing the scalar value $\mathcal{G}$ to a vector representation of multiple quantities. For example, the representation $\mathcal{G} = [A, D, N]$ could be used to encode the quantities related to people in the image. Additionally, since the YOLO network can recognise up to 80 classes of objects [26], a more detailed representation of a video can be extracted by extracting information about other object classes are encoded. This would increase the size of the embeddings, but the increase in robustness may be worth the information leak and processing time costs (see Section 5).

3.7. Evaluation Method

Following from previous papers [2], [33], [15], [16], the area under the curve (AUC) value for the Receiver Operating Characteristic (ROC) is used for all model evaluations (at the frame level). This is calculated by increasing the threshold at which an anomaly score is predicted to be an anomaly. This increases the rate that both true positives and false positives occur [34].

4. Experiments

4.1. Implementation Details

Each video is divided into $M = 32$ non-overlapping video snippets. The R3D features are generated using the pre-trained network 'Resnet-152' with Kinetics-700 and Moments In Time datasets [4]. Features are extracted from the global average pooling layer with dimensions 1024 in the case of I3D network, and from the global average pooling layer with dimensions 2048 in the case of the R3D network. As generating new optical flow features using the R3D network is very computationally expensive, the publicly available pre-trained I3D features are used for the multi-stream experiments [35].

Object class and pixel location data was extracted from the video frames using the small version of the YOLOv5 network (YOLOv5s). Only person and vehicle class features were extracted. This data was then used to mask the original RGB frames for all the videos in the data subset, and features were creating by passing these masked videos into the R3D feature extraction network. Both the vehicles and persons are masked to increase the likelihood that any persons interacting

with vehicles that are not directly detected by the YOLO algorithm are included in the masks.

Following from the structure of Sultani et al. [2], the model consists of three fully connected (FC) layers of each with 512, 32, and 1 neurons. A dropout value of 60% is used after each layer. A ReLU activation function is used for the first two FC layers and a Sigmoid function is used for the last.

During training, 30 randomly selected anomalous and normal videos are selected as a batch. The 'Adagrad' optimiser is used with an initial learning rate and weight decay of 0.001. The $\lambda_{smoothness}$ and $\lambda_{sparsity}$ terms are empirically set to 2×10^{-5}. The value $k = 4$ is used unless otherwise stated.

It was observed that AUC results were varying by up to 4% for runs with identical settings. This appeared to be caused by the random nature of batch video selection, but has not been evaluated or mentioned in previous works. To facilitate reproducibility of the results obtained in this work, the random seed has been set to the value '52345' for all experiments [36]. Fixing the random seed value was found to reduce AUC variability. It was found that the standard error across all experiments using the fixed seed value was 1.5% for the ROC AUC percentage (at a 95% confidence interval), unless otherwise stated.

The YOLO output data was used as input to the GSAFE feature extractor to generate the GSAFE embeddings with dimensions 32×2048. Vehicles are omitted from the data since the aim of GSAFE is to use a small amount of information to creating embeddings.

4.2. Previous Work Baselines

A comparison between previous implementations [2] [16] [15] and this implementation is shown in Table 1. This provides an overview of the differences in using single and multi-stream methods and also compares the performance of the I3D and R3D RGB features. The multi-stream features are produced by concatenating the RGB and Flow features before inputting them to the model. The value $k = 4$ is chosen for its good performance in [15]. The results show

TABLE 1. AUC SCORE RESULTS FOR I3D AND R3D FEATURES USING BOTH THE $k = 1$ [2] AND $k = 4$ [15] METHODS ON THE FULL UCF-CRIME DATASET. THE '*' INDICATES THAT I3D FLOW FEATURES WERE USED WITH THE R3D RGB FEATURES.

Input Features	AUC (%) (k=1)	AUC (%) (k=4)
I3D RGB	79.3	80.3
I3D Flow	80.8	80.6
R3D RGB	76.9	80.0
I3D RGB+Flow	82.9	82.8
R3D RGB+Flow*	81.0	83.4

that using the multi-stream input produced the highest performance. For both R3D RGB and multi-stream methods, using $k = 4$ produced higher AUC score results. For I3D, the choice of k led to statistically insignificant score differences. The difference in performance between R3D and I3D was statistically insignificant for RGB and multi-stream methods for $k = 4$. On this basis, the R3D method with $k = 4$ is chosen for its performance for further experiments.

4.3. Scene Masking

The AUC results for each of the video masking schemes are presented in Table 2.

TABLE 2. THE ROC AUC PERCENTAGES FOR EACH OF THE DIFFERENT MASKING SCHEMES. THESE RESULTS REPRESENT THE MEANS OF NUMEROUS TESTS WITH A STANDARD ERROR OF $\pm 0.4\%$.

Input Features	AUC (%)
RGB	79.4
ShowAll	78.5
ShowOne	76.5
ShowNone	75.3

In theory, if the R3D network can perfectly mitigate scene bias, then it is expected that the model performance will be equivalent for both the RGB features and the ShowAll features. It is also expected that the ShowNone features will produce an AUC score of 50%, since no information about human actions is present in the image. In practicality, however, the performance of the YOLO algorithm will also affect the quality of the masked features. As the quality of many of the UCF-Crime videos is poor, the YOLO algorithm occasionally fails to recognise a person in an image.

The performance of the model is reduced by masking the scene using the ShowAll mask in a small but statistically significant way. This suggests two possibilities; (a) that the R3D model may be using scene information to extract features, or; (b) that the small number of failed detections reduces AUC performance since not all persons are visible to the model.

Astonishingly, the model performs considerably well when the persons are masked out using the ShowNone mask, achieving an AUC score 15% higher than theoretically expected. This result provides strong evidence for (a) since; to accept (b), it must the case that the model performs robustly with sparse information for missed detections with the ShowNone masked, but is not robust to sparse loss of information with the ShowAll mask. These results and those of Choi et. al [5] suggest that (a) is a more consistent explanation.

4.4. GSAFE and RGB Comparison

The overall AUC performance for both the GSAFE embeddings and the R3D RGB features using the MIL model were evaluated for the UCF-Crime data subset. These results are displayed in Table 3.

TABLE 3. A COMPARISON OF THE AUC SCORES FOR THE MODEL USING GSAFE AND RGB FEATURES.

Input Features	AUC (%)
R3D RGB	79.4
GSAFE ($\mathcal{G}$)	68.6

The results in Table 3 show that the GSAFE embeddings perform considerably well compared to the R3D RGB method, with only a 10.8% difference in AUC scores. To further evaluate these results, an ablation study was conducted on the components of the GSAFE. The anomaly score predictions for various videos are visualised in Figure 6.

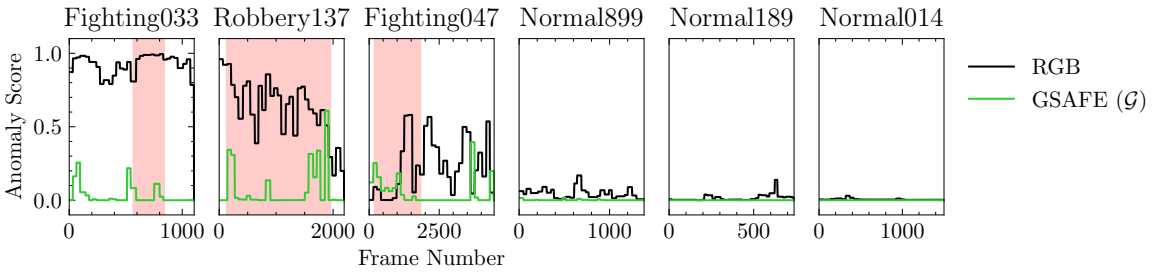


Figure 6. A visualisation of anomaly score predictions for the R3D RGB features and the GSAFE embeddings $\mathcal{G}$.

4.5. GSAFE Ablation Study

To conduct the ablation study, the stages of GSAFE were decomposed to allow evaluation of the geometric quantities as features without transforming them using Equation 9. For these decomposed features, the resampling function γ was applied to each of three quantities to create new 'raw' input features. These raw input features consisting of the values in N, D and A were used as model inputs individually, with the results presented in Table 4. The quantities $\bar{D}$ and $\bar{A}$ were also used as individual feature inputs. For $\bar{D}$ and $\bar{A}$, the representation learned during training resulted in many false positives and negatives, leading to an AUC score less than 50%. This suggests that the training and test distributions of these features are not similar. The model was unable to discriminate between anomalous and normal snippets for the features N and D. The individual features N, D and A were concatenated to produce a raw input 'trio' feature (in the form [N,D,A]). This 'trio' feature and the A feature produced similar results that were statistically equivalent.

When the raw input features were transformed using Equation 9 to produce the GSAFE embeddings $\mathcal{G}$, the AUC performance improved by a further 7-8%. The anomaly scores for these features are visualised in Figure 7.

TABLE 4. ABLATION STUDY FOR GSAFE USING AN EVALUATION OF THE OVERALL AUC PERFORMANCE FOR THE VARIANTS OF THE GEOMETRIC QUANTITIES.

Input Features	AUC (%)
G	68.6
Trio Input [N,D,A]	60.7
N	51.0
D	50.0
A	61.9
$\bar{D}$	36.8
$\bar{A}$	37.2

These ablation study results further demonstrate that the GSAFE theoretical formulation described in Equation 9 can sufficiently model the likelihood that a crime is occurring in a video from the chosen data subset. Similarly to Tian et al. [15], an evaluation of the AUC performance on each individual class was also carried out. For this experiment, the entire training data was used for all of the UCF-Crime data subset, and testing was carried out only on the videos for a single class. The results are displayed in Figure 8. This evaluation of the model performance on individual classes of crime indicate that GSAFE is competitive with the results achieved by Sultani et al. for all the classes tested, and outperforms their results for Arrest, Fighting and Robbery.

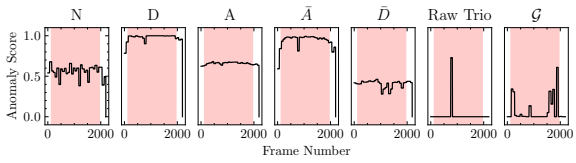


Figure 7. A comparison of the anomaly score prediction values for the various component of GSAFE. Scores shown for the video 'Robbery137'.

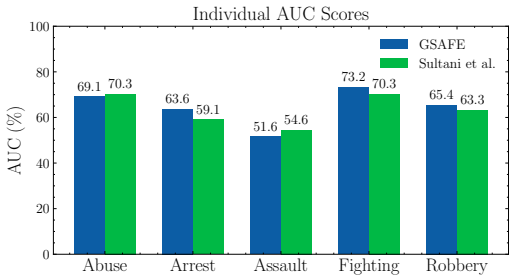


Figure 8. Comparison of single class GSAFE AUC performance with the Sultani et al. [2] results.

4.6. Computational Efficiency Analysis

The R3D network was too large to load with a 2GB Quadro P260 GPU, as there was insufficient memory even using a batch size of 1. Conversely, the YOLOv5s network can be run at a speed of 41.7FPS on this GPU.

Using a 2.4GHz Intel i7-2750QM with 8 cores (a laptop CPU introduced in 2011), the feature extraction speeds were measured. This serves as a measure of how well these methods can be applied to small devices. The R3D model was found to process video frames at a rate of 7.2FPS. In comparison, the entire GSAFE method was found to process video frames at a rate of 12.5FPS. These results exclude loading times, although the R3D network is likely to take longer to load given that it has 117 million parameters, while the YOLOv5s network only has 7.2 million.

5. Discussion

The results in Section 4 show that GSAFE embeddings perform considerably well with the weakly-supervised deep learning model.

The derived quantity $\mathcal{G}$ is analogous to a person density function describing the density of persons in a scene. Despite the considerable performance of GSAFE on the MIL

problem, there is likely to be a considerable overlap between the distribution of videos with densely positioned persons that contain crime and the distribution of those that do not contain crime. If a dataset truly represents these density distributions, the AUC performance should be closer to 50% if the density is is balanced between both classes of activity. For this reason, GSAFE is not generally applicable to detection of previously unseen criminal activity in video. However, these results also highlight that other more complex feature extractors, such as R3D, are also not generally applicable for this purpose given that scene representation bias produced statistically significant performance differences. Other models such as I3D are trained in a similar manner, so they may also exhibit this bias.

Furthermore, testing on other available benchmark datasets is a typical method used to demonstrate the generalisability of a model. For example, the ShanghaiTech and UCSD Ped2 video datasets are typically used for evaluation along with UCF-Crime. Given that both of these datasets are also based on CCTV footage of a similar nature, it is the case that the same idiosyncrasies related to the density of people in video could quite easily exist in these other datasets. While evaluation of multiple datasets was beyond the resources of this work, it remains a very irresponsible assumption that using these other datasets for evaluation serves as evidence that that a model is capable of generalising to new unseen data.

It is quite remarkable that the ShowNone R3D features outperformed GSAFE by 6.7%, despite the fact that the vast majority of the RGB information related to the persons in the frames had been removed using the mask. The considerable model performance using the ShowNone masked videos can be attributed to two plausible factors - (a) the R3D model uses information about the scene in a video which results in scene representation bias - or (b) R3D is exceptionally capable of encoding sparse information about human activity from the occasional frames in which people remain undetected (with ShowNone missed detections) but is impacted when sparse information is removed (with ShowAll missed detections). Both conditions for (b) appear inconsistent. Choi et al. [5] demonstrate that scene representation bias occurs with models pre-trained using the Kinetics dataset. The results in this paper provide further evidence of bias with the small but statistically significant difference in the ShowAll mask and original RGB R3D features. In summary, there is strong evidence for (a) and no evidence for (b). In future work, using a larger and more robust YOLO network may reduce these missed detections and provide even stronger evidence.

For specific applications, GSAFE is a favourable method, and can even be implemented on low-resource systems that cannot run R3D and other large feature extractors. For example, with systems that have low memory GPUs or CPU-only architectures, GSAFE can be implemented if R3D is too large. In terms of processing speed, GSAFE outperforms R3D on the tested CPU by 74%. Examples of these low-resource systems include physically small or budget devices, such as surveillance cameras and smartphones.

If more robust embeddings are required for an application, the size of the data type used to represent an image with $\mathcal{G}$ can be increased in size. For example, encoding the three values N, D and A individually (without transformation to $\mathcal{G}$) would provide more granularity in the feature representation. This approach may be useful for CSAM detection, where a more detailed representation of the features in a video are needed. Shorter videos will have a narrower feature distribution, and so a more detailed representation of each image may prevent false positives for videos that do not contain previously identified abusive content.

In all cases, GSAFE offers interpretability when compared to other deep learning feature extraction methods. Since the features are derived using geometry, expert knowledge and human reasoning can be used in computer vision system design rather than relying on a neural network to learn the desired feature representation. The mathematical basis of GSAFE makes it extremely favourable for applications where regulatory approval success relies upon the interpretability of the system.

GSAFE is particularly powerful for systems where preservation of privacy is paramount. Transmission of RGB data may allow a receiver or interceptor to extract information about persons in the image, the location of the image, and possibly other data that may be contained in the EXIF data of transmitted frames. It is even possible to reconstruct information using extracted features from a neural network or hashed image [19]. With GSAFE, even perfect reconstruction of extracted features only provides information about geometries of detected persons. If privacy and information security is preferred over data representation complexity, then removing any aspect of the feature representation that may pose a risk is a system design option. For example, it may be desired not to include the detected classes in an image in a feature that encodes p_n, and so only the coordinates of the detected objects can be transmitted.

6. Conclusion

This work develops a new method named GSAFE for extracting features from video. GSAFE is based on a mathematical formulation which preserves feature representations across a number of image transformations, such as image rotation, scaling and pixel colour modifications. This makes it resilient to attacks which Apple's CSAM detection technology 'NeuralHash' is vulnerable to. GSAFE has also be experimentally tested on the challenging problem of MIL anomaly detection. To responsibly develop GSAFE, this work also demonstrated that traditional deep learning image feature extraction methods are vulnerable to scene representation bias and lack interpretability or transparency. The mathematical basis of GSAFE provides a interpretable and logical solution to these problems. This work outlines the applications and limitations of GSAFE in the context of CSAM and violent crime detection. Motivation has been provided for why further research is needed to understand the distribution of anomalies represented by all benchmark datasets. The lightweight nature of GSAFE has been experimentally verified using computational efficiency evaluations. Finally, the customisable nature of GSAFE that it is suitable for a variety of scenarios where client-side processing is needed and system/user privacy and security is valued. In future work, evaluation of GSAFE on larger datasets could provide more insight. Additionally, the YOLO network used with GSAFE could be trained to identify other object features which can be used as identifiers for previously identified CSAM on platforms.

Acknowledgments

I would like to acknowledge and thank my supervisor Dr. Suzanne Little for the helpful discussions we had throughout my research process.

References

[1] "Experts meet to identify victims of child sexual abuse," https://www.europol.europa.eu/media-press/newsroom/news/experts-meet-to-identify-victims-of-child-sexual-abuse.

[2] W. Sultani, C. Chen, and M. Shah, "Real-World Anomaly Detection in Surveillance Videos," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Jun. 2018, pp. 6479–6488.

[3] J. Carreira and A. Zisserman, "Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul. 2017, pp. 4724–4733.

[4] K. Hara, H. Kataoka, and Y. Satoh, "Can Spatiotemporal 3D CNNs Retrace the History of 2D CNNs and ImageNet?" in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Jun. 2018, pp. 6546–6555.

[5] J. Choi, C. Gao, J. C. E. Messou, and J.-B. Huang, "Why can't I dance in a mall? learning to mitigate scene bias in action recognition," in *Proceedings of the 33rd International Conference on Neural Information Processing Systems*. Red Hook, NY, USA: Curran Associates Inc., Dec. 2019, no. 77, pp. 853–865.

[6] M. Hasan, J. Choi, J. Neumann, A. K. Roy-Chowdhury, and L. S. Davis, "Learning Temporal Regularity in Video Sequences," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2016, pp. 733–742.

[7] W. Luo, W. Liu, and S. Gao, "A Revisit of Sparse Coding Based Anomaly Detection in Stacked RNN Framework," in *2017 IEEE International Conference on Computer Vision (ICCV)*, Oct. 2017, pp. 341–349.

[8] A. Basharat, A. Gritai, and M. Shah, "Learning object motion patterns for anomaly detection and improved object detection," in *2008 IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2008, pp. 1–8.

[9] R. Bensch, N. Scherf, J. Huisken, T. Brox, and O. Ronneberger, "Spatiotemporal Deformable Prototypes for Motion Anomaly Detection," *International Journal of Computer Vision*, vol. 122, no. 3, pp. 502–523, May 2017.

[10] D. Abati, A. Porrello, S. Calderara, and R. Cucchiara, "Latent Space Autoregression for Novelty Detection," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2019, pp. 481–490.

[11] H. Park, J. Noh, and B. Ham, "Learning Memory-Guided Normality for Anomaly Detection," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2020, pp. 14 360–14 369.

[12] D. Gong, L. Liu, V. Le, B. Saha, M. R. Mansour, S. Venkatesh, and A. Van Den Hengel, "Memorizing Normality to Detect Anomaly: Memory-Augmented Deep Autoencoder for Unsupervised Anomaly Detection," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, Oct. 2019, pp. 1705–1714.

[13] M. Sabokrou, M. Khalooei, M. Fathy, and E. Adeli, "Adversarially Learned One-Class Classifier for Novelty Detection," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Jun. 2018, pp. 3379–3388.

[14] S. Andrews, I. Tsochantaridis, and T. Hofmann, "Support Vector Machines for Multiple-Instance Learning," *Advances in Neural Information Processing Systems*, vol. 15, pp. 561–568, Jan. 2002.

[15] Y. Tian, G. Pang, Y. Chen, R. Singh, J. W. Verjans, and G. Carneiro, "Weakly-supervised Video Anomaly Detection with Robust Temporal Feature Magnitude Learning," *arXiv:2101.10030 [cs]*, Aug. 2021.

[16] H. Lv, C. Zhou, Z. Cui, C. Xu, Y. Li, and J. Yang, "Localizing Anomalies From Weakly-Labeled Videos," *IEEE Transactions on Image Processing*, vol. 30, pp. 4505–4515, 2021.

[17] K. Joffres, M. Bouchard, R. Frank, and B. Westlake, "Strategies to Disrupt Online Child Pornography Networks," in *Proceedings of the 2011 European Intelligence and Security Informatics Conference*, ser. EISIC '11. USA: IEEE Computer Society, Sep. 2011, pp. 163–170.

[18] C. Peersman, C. Schulze, A. Rashid, M. Brennan, and C. Fischer, "iCOP: Automatically Identifying New Child Abuse Media in P2P Networks," in *2014 IEEE Security and Privacy Workshops*, May 2014, pp. 124–131.

[19] L. Struppek, D. Hintersdorf, D. Neider, and K. Kersting, "Learning to Break Deep Perceptual Hashing: The Use Case NeuralHash," in *2022 ACM Conference on Fairness, Accountability, and Transparency*, ser. FAccT '22. New York, NY, USA: Association for Computing Machinery, Jun. 2022, pp. 58–69.

[20] H. Kataoka, T. Wakamiya, K. Hara, and Y. Satoh, "Would Mega-scale Datasets Further Enhance Spatiotemporal 3D CNNs?" *ArXiv*, 2020.

[21] M. Lapin, M. Hein, and B. Schiele, "Loss Functions for Top-k Error: Analysis and Insights," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2016, pp. 1468–1477.

[22] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2016, pp. 770–778.

[23] A. Zisserman, J. Carreira, K. Simonyan, W. Kay, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, and M. Suleyman, "The Kinetics Human Action Video Dataset," 2017.

[24] C. Gao, Q. Cai, and S. Ming, "YOLOv4 Object Detection Algorithm with Efficient Channel Attention Mechanism," in *2020 5th International Conference on Mechanical, Control and Computer Engineering (ICMCCE)*, Dec. 2020, pp. 1764–1770.

[25] C.-Y. Wang, H.-Y. Mark Liao, Y.-H. Wu, P.-Y. Chen, J.-W. Hsieh, and I.-H. Yeh, "CSPNet: A New Backbone that can Enhance Learning Capability of CNN," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Jun. 2020, pp. 1571–1580.

[26] "Microsoft COCO: Common Objects in Context — SpringerLink," https://link.springer.com/chapter/10.1007/978-3-319-10602-1_48.

[27] "Paul Mcbrien / 2022-mcm-GSAFE," https://gitlab.computing.dcu.ie/mcbriep2/2022-mcm-GSAFE.

[28] SciPy 1.0 Contributors, "SciPy 1.0: Fundamental algorithms for scientific computing in python," *Nature Methods*, vol. 17, pp. 261–272, 2020.

[29] "Detecting child sexual abuse material: A comprehensive survey," *Forensic Science International: Digital Investigation*, vol. 34, p. 301022, Sep. 2020.

[30] P. Eleuterio and M. Polastro, "An adaptive sampling strategy for automatic detection of child pornographic videos," 2012.

[31] Apple, "Apple CSAM Detection Technical Summary," Technical Summary.

[32] F. S. Beckman and D. Quarles, "On isometries of euclidean spaces," 1953.

[33] Y. Lin, W. Chi, W. Sun, S. Liu, and D. Fan, "Human Action Recognition Algorithm Based on Improved ResNet and Skeletal Keypoints in Single Image," *Mathematical Problems in Engineering*, vol. 2020, p. e6954174, Jun. 2020.

[34] A. P. Bradley, "The use of the area under the ROC curve in the evaluation of machine learning algorithms," *Pattern Recognition*, vol. 30, no. 7, pp. 1145–1159, Jul. 1997.

[35] J. Park, J. Kim, and B. Han, "Learning to Adapt to Unseen Abnormal Activities Under Weak Supervision," in *Computer Vision – ACCV 2020*, ser. Lecture Notes in Computer Science, H. Ishikawa, C.-L. Liu, T. Pajdla, and J. Shi, Eds. Cham: Springer International Publishing, 2021, pp. 514–529.

[36] "Reproducibility — PyTorch 1.12 documentation," https://pytorch.org/docs/stable/notes/randomness.html.

Appendix

I declare that this material, which I now submit for assessment, is entirely my own work and has not been taken from the work of others, save and to the extent that such work has been cited and acknowledged within the text of my work. I understand that plagiarism, collusion, and copying are grave and serious offences in the university and accept the penalties that would be imposed should I engage in plagiarism, collusion or copying. I have read and understood the Assignment Regulations. I have identified and included the source of all facts, ideas, opinions, and viewpoints of others in the assignment references. Direct quotations from books, journal articles, internet sources, module text, or any other source whatsoever are acknowledged and the source cited are identified in the assignment references. This assignment, or any part of it, has not been previously submitted by me or any other person for assessment on this or any other course of study.

I have read and understood the referencing guidelines found at http://www.dcu.ie/info/regulations/plagiarism.shtml , https://www4.dcu.ie/students/az/plagiarism and/or recommended in the assignment guidelines.