symmetry

MDPI

Article

Transfer Detection of YOLO to Focus CNN's Attention on Nude Regions for Adult Content Detection

Nouar AlDahoul [1,*], Hezerul Abdul Karim [1], Mohd Haris Lye Abdullah [1], Mohammad Faizal Ahmad Fauzi [1], Abdulaziz Saleh Ba Wazir [1], Sarina Mansor [1] and John See [2]

[1] Faculty of Engineering, Multimedia University, Cyberjaya 63100, Malaysia; hezerul@mmu.edu.my (H.A.K.); haris.lye@mmu.edu.my (M.H.L.A.); faizal1@mmu.edu.my (M.F.A.F.); 1191400100@student.mmu.edu.my (A.S.B.W.); sarina.mansor@mmu.edu.my (S.M.)

[2] Faculty of Computing and Informatics, Multimedia University, Cyberjaya 63100, Malaysia; johnsee@mmu.edu.my

* Correspondence: nouar.aldahoul@live.iium.edu.my

Abstract: Video pornography and nudity detection aim to detect and classify people in videos into nude or normal for censorship purposes. Recent literature has demonstrated pornography detection utilising the convolutional neural network (CNN) to extract features directly from the whole frames and support vector machine (SVM) to classify the extracted features into two categories. However, existing methods were not able to detect the small-scale content of pornography and nudity in frames with diverse backgrounds. This limitation has led to a high false-negative rate (FNR) and misclassification of nude frames as normal ones. In order to address this matter, this paper explores the limitation of the existing convolutional-only approaches focusing the visual attention of CNN on the expected nude regions inside the frames to reduce the FNR. The You Only Look Once (YOLO) object detector was transferred to the pornography and nudity detection application to detect persons as regions of interest (ROIs), which were applied to CNN and SVM for nude/normal classification. Several experiments were conducted to compare the performance of various CNNs and classifiers using our proposed dataset. It was found that ResNet101 with random forest outperformed other models concerning the F1-score of 90.03% and accuracy of 87.75%. Furthermore, an ablation study was performed to demonstrate the impact of adding the YOLO before the CNN. YOLO–CNN was shown to outperform CNN-only in terms of accuracy, which was increased from 85.5% to 89.5%. Additionally, a new benchmark dataset with challenging content, including various human sizes and backgrounds, was proposed.

Keywords: pornography detection; nudity detection; convolutional neural network; you only look once; feature extraction; visual attention; region of interest

Citation: AlDahoul, N.; Abdul Karim, H.; Lye Abdullah, M.H.; Ahmad Fauzi, M.F.; Ba Wazir, A.S.; Mansor, S.; See, J. Transfer Detection of YOLO to Focus CNN's Attention on Nude Regions for Adult Content Detection. *Symmetry* **2021**, *13*, 26. https://doi.org/10.3390/sym13010026

Received: 6 December 2020
Accepted: 21 December 2020
Published: 25 December 2020

1. Introduction

Given the vast growth and quantity of videos and images in all types of media nowadays, various content understanding methods have been developed and employed in real-world scenarios. Pornography and nudity detection in videos or a series of images have a significant impact on visual censorship applications (e.g., TV broadcasting and video-sharing platforms such as YouTube and TikTok). Nudity content can include one or more persons with explicit exposure of full, upper (for female) or lower body in various scales, positions, and backgrounds. Pornographic content refers to the exposure of specific sexual organs and sexual activity. Numerous films are broadcasted to the public 24 h a day and possibly contain pornography and nudity content which may be viewed by underage children and cause serious social problems. In addition, nudity is prohibited in TV broadcasting in some countries. Similarly, video-sharing platforms are also governed by having rules to censor nudity and sexual content. In addressing some of these issues, manual intervention is typically employed for censorship. However, the introduction of

an automatic nudity detection mechanism to assist in film censorship would be extremely helpful for TV broadcasting companies to avoid massive fines due to human negligence. As such, it is necessary to automate the censorship process and design an accurate nudity detection system.

Although, the task of pornography and nudity detection in a series of images is quite challenging due to variations in many factors including the human position (standing, sitting, lying, etc.), size of the nude people relative to the frame's size, clothes colour, skin colour, and background such as forest, snow, beach, supermarket, indoor, and streets scenes, etc.

Several solutions have been proposed to detect pornography utilising skin colour detection [1–3]. However, these techniques suffer from high false positives rate and the inability to consider illumination changes and objects with similar skin colours in the background such as wood, sand, and skin or fur of certain animals. Additionally, the skin detector often fails when there is no explicit exposure of the upper or lower body, such as people wearing swim attire and wrestling suits. Moreover, feature engineering methods for pornography detection, such as image zoning with colour and texture features [4], and random forest with skin features [5] have been explored and extensively examined, though this paper does not focus on these methods. On the contrary, feature learning methods for pornography and nudity detection still have room for improvement in order to minimise the false positive and negative rates.

Having said that, feature engineering and feature learning are two main methods used to extract or learn features that are applied to a classifier to categorise them into numerous classes. Feature engineering includes the manual design and selection of discriminative features to improve the classification model's performance. This 'handcrafted' feature approach is sensitive, designed by an expert, and based on domain knowledge. On the other hand, feature learning is an optimal solution when there is limited expert knowledge available. It is adaptable to various applications and able to learn high abstract features automatically and directly from raw pixels [6]. Deep models, which learn features in an end-to-end scheme, have demonstrated the proficiency in computer vision tasks and specifically in the application of nude/normal classification [7–13]. Convolutional neural networks (CNNs), which are well known deep neural networks, have several layers including convolutional, pooling, normalisation, activation, and fully connected. CNN's weights or parameters are fine tuned utilising a stochastic gradient descent (SGD) algorithm [14,15].

In order to enhance CNN's performance, large-scale datasets such as ImageNet [16] and COCO [17] are utilised for training the model. ImageNet has a subset of 1.2 million images which was used in 2010 in the Challenge of Large-Scale Visual Recognition for the classification of visual objects into 1000 categories [16]. Similarly, COCO 2017 contains 164k images and 80 classes [17]. However, it is difficult to collect a large-scale dataset for a specific application such as for nudity detection. Therefore, to train a model with a small or medium-scale dataset, transfer learning has been shown to utilise the weights of CNNs trained on ImageNet such as AlexNet [18], VGG16 [19], GoogleNet [20], Inception3 [21], ResNet50, and ResNet101 [22]. These weights of the first layers are usually frozen without tuning to utilise them for extracting features from a new small-scale dataset. In this paper, the same approach of transfer learning was used with our proposed challenging nudity dataset to extract features from the images. Additionally, the approach of transfer detection was also demonstrated with You Only Look Once (YOLO) as a COCO-based pre-trained detector in detecting a specific class such as a person and ignore other classes.

In the literature, several research works have demonstrated convolutional neural networks (CNNs only) for visual pornographic content in classifying them into two classes—porn and non-porn—and were found to have high detection accuracy [7,9]. Various pre-trained models such as AlexNet [18], GoogleNet [20], and ResNet [22] were also utilised to extract only spatial features from video frames and to classify them by support vector machine (SVM) [23]. On the other hand, to extract spatiotemporal features, the literature has proposed many methods such as long short-term memory (LSTM) recurrent

network [10,24], two CNNs in parallel: one for static (picture) and another for dynamic (motion) information utilising optical flow and MPEG motion vectors [9], and 3D CNN [25] to detect pornography utilising VGG-C3D [26] with a Linear SVM classifier and ResNet R (2+1) D CNN [27] with Softmax classifier. However, the previous methods mentioned depend on SGD to tune the model's parameters and consequently take a considerable time for training. However, to reduce training time, least-square solutions such as local receptive field-based extreme learning machine was proposed for adult content detection [13].

The NPDI pornography dataset [28,29] has been extensively used to train and evaluate various models to detect pornography in videos [7,9–13]. However, when the proposed models were used with real-life broadcasted movies for censorship purposes, the shortcomings of the existing methods and dataset have become evident. Although existing training strategies were able to classify pornographic videos into two classes—normal and nude—they were too simple to detect nudity with any scale and complex backgrounds in video frames. In other words, the NPDI dataset focuses mainly on pornographic content (sexual actions and exposure of genital organs) that covers the whole camera's field of view. Therefore, when we utilised the models trained on the NPDI dataset for real-world films, they were unable to detect nude frames, thus producing a high false negative rate. The main reason was that CNN loses the focus on nudity content and instead, considers other objects in the backgrounds. Therefore, to overcome the previous limitations of the dataset, this paper proposes a new nudity dataset with the challenging content of various nudity scales and backgrounds.

Although CNN-only methods have shown superior performance for pornography detection when the pornographic content covers the frames largely, their performance is still limited in detecting nudity when the nude or porn persons cover only small regions inside the frame and when the background is complex, such as nude people in a forest, snow, beach, supermarket, indoor, and streets. Nian et al. studied this problem and proposed a fast image scanning method which was based on the sliding window approach to detect the nude region [30]. However, the drawback associated with this approach was that the speed and number of sliding times depend on the width of the frame and the sliding stride. The authors claimed while mentioning to [31] that applying human detection is difficult as the frames may only show a small part of the human body in pornographic videos.

Human detection has been explored extensively in the literature. Fastest pedestrian detector of the west (FPDW) which is a feature engineering method utilises histogram of gradient (HOG) at different scales [32]. It balances the trade-off between accuracy and speed to detect humans faster than the state-of-the-art methods. Although FPDW requires a large number of pixels, only one frame is sufficient for detecting humans. Additionally, a combination of optical flow and AlexNet has been used to detect humans in various scales, viewpoints, positions, orientations, and cloth using varied altitudes of a camera attached to a moving airborne [33]. This combination requires two frames to find a set of candidate objects. Similarly, some works have demonstrated YOLO for human detection [34,35].

Some works have targeted nudity detection to remove the whole inappropriate frames from the video, while other works were proposed to filter out only sensitive regions utilising the approach of image-to-image translation that was based on adversarial training [36]. In addition, the detection of body parts with multi-labelled classification was also demonstrated [37].

Ensemble methods were also utilised in adult content recognition [38–40]. Here, a weighted sum of several deep neural networks (DNNs) was used to express the CNN's weights as a linear regression problem learned using ordinary least squares (OLS) [38]. Additionally, ensemble framework uncertain inference employed a Bayesian network. The prior global confidence of pornography for the candidate image was extracted using GoogleNet/ResNet-50 [39]. In addition, uncertain evidence was extracted using Single Shot MultiBox Detector to detect visual objects of the six sensitive semantic components [39].

Moreover, ensemble-based multi-instance learning has been proposed to utilise a group of extreme learning machine (ELM) classifiers with a different number of hidden nodes [40].

In this paper, we provide further insight into the nudity classification task that includes YOLO [41–43] to solve the problem of CNN-only methods. If the frame has more than one person, each person image patch is passed to the CNN to produce one category that is stored in a list. The list has labels of all person patches in one frame. If one element in the list has a nude label, the whole frame is considered nude, whereas if all labels in the list have normal elements, the whole frame is considered normal. Additionally, the proposed method detects nudity regions existing in different scales inside the frames with complex backgrounds. In other words, it can only edit and filter out only detected regions and keep other existing regions of the frame; thus, it reduces the interrupted period of the film and helps to understand the context.

This paper demonstrates a novel automated system in speeding up censorship and reduce the cost. The key contributions of this paper included the following:

- YOLO3 was utilised as a human detector. To the best of our knowledge, this is the first paper that uses YOLO for the nudity detection task.
- Pre-trained CNNs that have already been trained on the ImageNet dataset were demonstrated as feature extractors having fixed parameters of the first layers. The fully connected layers were removed. The objective was to find discriminative features for the normal/nude classification task.
- Various classifiers were used to replace the top layer of pre-trained CNNs to fit the proposed nudity dataset, in providing two distinct classes: normal and nude.
- An ablation study was undertaken to demonstrate the impact of adding YOLO before CNN.
- An ablation study was performed to provide further insight into the added advantage of data augmentation in nudity detection applications.
- An evaluation and comparison between various CNN-based feature extractors such as AlexNet [18], VGG16 [19], GoogleNet [20], Inception3 [21], ResNet50, and ResNet101 [22] are demonstrated.
- An evaluation and comparison between various classifiers such as K nearest neighbours (KNN) [44], random forest (RF) [45], extreme learning machine (ELM) [46], and SVM [23] with different kernels: linear (LSVM), Gaussian (GSVM), and polynomial (PSVM) were made.
- A new challenging nudity dataset was proposed containing humans in various scales with complex backgrounds such as forest, snowing, beach, supermarket, indoor, and streets.

The organisation of this paper is structured into four sections. Section 1 has provided a relevant background and aim of the study. Section 2 demonstrates the dataset, and three main blocks in the proposed system, including YOLO for human detection, various pre-trained CNNs for feature extraction, and various classifiers. In Section 3, experimental setup, and results are presented and discussed. An ablation study to gain insight on the effect of adding YOLO before CNN is also described. Section 4 summarises the outcome and significance of this work.

2. Materials and Methods

2.1. Datasets Overview

In this section, an overview of the training, validation, and testing images and videos, in both versions: augmented and non-augmented, is explored. Additionally, we demonstrate the experimental protocol that describes how to split the dataset into train, validation, and test.

2.1.1. ImageNet Dataset

ImageNet is a large-scale dataset consisting of 12 subtrees with a total of 3.2 million annotated images spread over 5247 categories, with an average of over 600 images for

each category [16]. ImageNet is considered a benchmark dataset to train and validate deep CNN models. Consequently, the models that were trained on ImageNet can be transferred to other applications when the new dataset is not sufficiently large enough to reduce the overfitting problem.

In this paper, various CNNs such as AlexNet, VGG16, GoogleNet, Inception3, ResNet50, and ResNet101, already trained on the ImageNet dataset, were used to extract features from the frames before being classified into nude or normal.

2.1.2. COCO Dataset

The COCO dataset contains photos of 91 object types with a total of 2.5 million labelled instances in 328k images [17]. In our experiments, the YOLOv3 deep learning-based human detector, already trained on the COCO dataset, was utilised to focus the attention on humans and ignore other objects in the backgrounds of the frames.

2.1.3. NDPI Dataset

The NPDI dataset includes about 80 h of 800 videos [28,29]. These videos were divided into 400 normal and 400 pornographic videos. The normal videos contain two subcategories: easy with random videos and difficult, which includes body skin content such as beach, wrestling, and swimming. An extended version of the NPDI dataset had 140 h of 2k videos, including 1000 pornographic and 1000 normal videos [47]. In the first experiment, we utilised the NPDI dataset in the training stage with a transfer learning approach to extract features from video's frames using pre-trained ResNet50. In addition, SVM was trained to classify the extracted features into two categories, namely, nude, and normal. We utilised the same CNN and SVM used in state-of-the-art solutions to compare the proposed method with existing ones. Figure 1. illustrates a few samples of the NPDI dataset. The majority of samples in NPDI have explicit pornographic content that covers the whole frames and includes the exposure of specific sexual organs and sexual activity.

Figure 1. A few samples of frames in NPDI dataset.

2.1.4. Our Challenging MMU Dataset

To address and overcome the limitation of the well known pornography dataset such as NPDI, this dataset was carefully collected from the internet to include various backgrounds such as forest, snowing, beach, supermarket, indoor, and streets. It is also considered challenging due to variations in many factors including human position (standing, sitting, lying, etc.), size of nude people relative to frame's size, cloth, skin colour, and backgrounds. The dataset contains 800 images: 400 normal and 400 nudes images. The images were divided into training and validation sets with 80% (640) and 20% (160), respectively, as shown in Table 1. The number of training samples with augmentation (AUG) and without Augmentation (no AUG) is illustrated in Table 1. Figures 2 and 3 illustrate a few samples from this challenging dataset.

In addition, the size of this dataset was augmented eight times to become 6400 images, as shown in Figure 4, as follows:

(1) Flip image horizontally.
(2) Convert the two versions of the image (original and flipped) to grey.
(3) Lighten (whiten) the two versions.
(4) Darken the two versions.

This dataset was used to train and validate the models in the ablation experiment that targets the study of adding YOLO3 in the pipeline before CNN. For CNN-only, the training samples were the whole images. On the other hand, YOLO–CNN used human-only patches that were extracted from the images by YOLO3. A few samples of the human-only dataset are shown in Figure 5.

Table 1. Number of samples for training and validation.

Frames	Training (no AUG)	Training (AUG)	Validation	Total (no AUG)
Total	640	5120	160	800
Negative	320	2560	80	400
Positive	320	2560	80	400

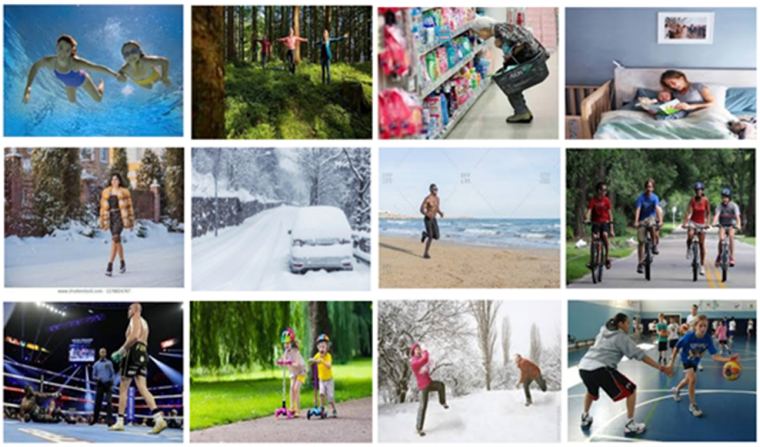

Figure 2. A few samples of normal frames with various activities, scales, positions, orientations, viewpoints, and cloth colours. The images have various resolutions, but they were resized to the same size.

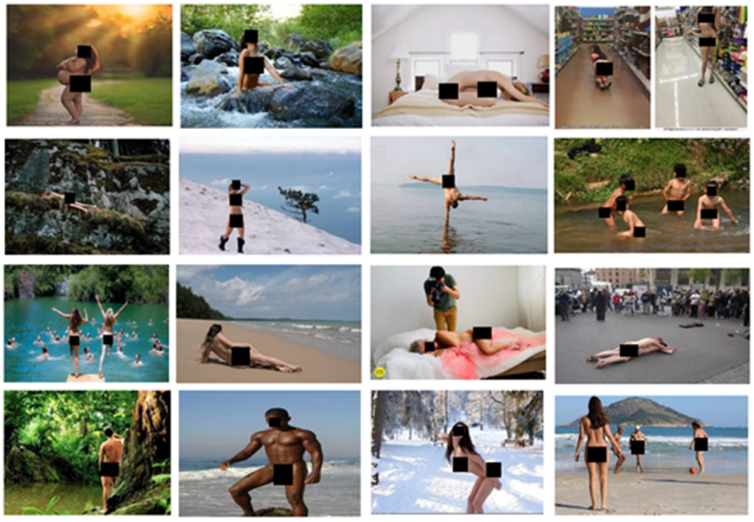

Figure 3. A few samples of nude frames with various backgrounds, activities, scales, positions, orientations, viewpoints, and skin colours. The images have various resolutions, but they were resized to the same size.

Figure 4. A few samples of images after data augmentation.

2.1.5. Testing Film Dataset

This dataset is also challenging and was only used for testing purposes. This consists of five real-world videos available on the internet, as described in Table 2. The total length of the testing videos is about one hour which can be divided into 360 videos with 10 s lengths for each video. Figure 6 shows a few samples of the frames. It is obvious that the scales or sizes of humans in the frames vary, and the backgrounds are complex, including green grass, indoor, streets, and beach. The dataset includes a total of 9891 frames that were applied to YOLO. As a result, YOLO discarded 1312 non-human frames and kept 8579 frames (5081 nudes and 3498 normal) that had at least one human. The dataset consists of 25,983 human images detected from 8579 frames selected from these five videos. The frames were selected using three frames per second (first, middle, and last frames).

In the first experiment, this dataset was used to test the model trained on NPDI. Additionally, these five videos were also used in the second experiment to test the model trained on human-only images. In the third experiment, we utilised these five videos to validate and compare various feature extractors and classifiers.

Figure 5. Few samples from the human-only images detected and extracted by YOLO3 using **MMU dataset.**

Table 2. Testing videos description.

Video ID	Video Title	Length (M:S)
1	world naked bike rider	02:03
2	Naked Parents	45:22
3	Il est de retour . . . uniforme et képi restauré	01:09
4	Naked European Walking Tour 2017	14:40
5	World naked bike ride London June 8th, 2013	02:17

Figure 6. A few samples of frames in the testing film dataset.

2.2. Methodology

In this section, the methodology adopted in this study is explored in detail. The proposed approach of model fusion represents a combination of YOLO for human detection and one of the CNNs including AlexNet [18], VGG16 [19], GoogleNet [20], Inception3 [21], and ResNet [22] for feature extraction. In addition, the last fully nconnected layers of CNNs were replaced by one of the various classifiers such as KNN [44], RF [45], ELM [46], and SVM with different kernels: linear (LSVM), Gaussian (GSVM), and polynomial (PSVM) [23] for nude/normal classification. The proposed system diagram is shown in Figure 7. The video's frame was applied to YOLO. The outcome of this stage was many image patches of humans available in the frame. These patches were applied to a pre-trained CNN to extract features that the classifier utilised to give two classes: nude and normal. The red and black rectangles on the naked regions in this paper were added manually for public consideration.

A brief review of each module used in the proposed detection system (YOLO, pre-trained CNN, and classifier) is summarised in the following subsections.

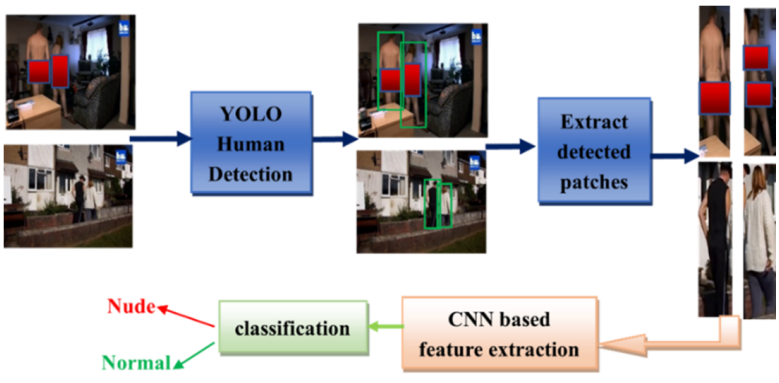

Figure 7. Illustration of the proposed pornography and nudity detection system.

2.2.1. YOLO-Based Human Detection

YOLO is a unified simple model that is normally used when real-time detection is required, without losing too much accuracy [41–43]. The detection in YOLO is considered a regression task. YOLO has an end-to-end neural network to directly extract features from full images utilising convolutional layers and predicting bounding boxes and class probabilities using fully connected layers. Although YOLO produces more localisation errors, this makes less than half the number of background errors compared to fast region CNN (R-CNN) because it can see the full image during training [41]. The YOLO approach differs from classification given it was trained on a loss function that directly matches detection performance, and the whole model is trained jointly [41].

The YOLO network architecture was inspired by the GoogleNet model for image classification, comprising 24 convolutional layers followed by two fully connected layers instead of the inception modules used by GoogleNet, as shown in Figure 8. Convolutional layers were trained on the ImageNet classification with an input image of 224 × 224, and then the resolution was doubled for detection [41].

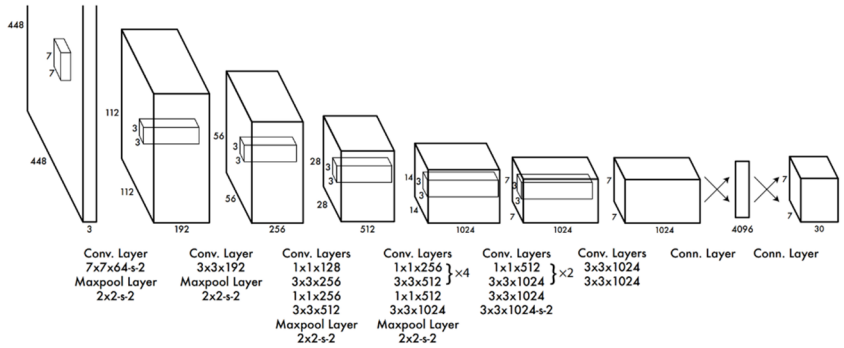

Figure 8. YOLO architecture [41].

In the YOLO model, the input image is divided into an S × S grid. The number of bounding boxes predicted for each grid cell (B) is 2, and the number of class probabilities (C) is 20. As a result, the size of the final output is S × S × (B × 5 + C) = 7 × 7 × 30 tensor of predictions [41].

The box confidence is formed as $P_r\ (Object) \times IOU_{pred}^{truth}$. If the object is not available in the cell, the confidence scores would be 0. Otherwise, it would be equal to the intersection over union (IOU) between the ground truth and predicted boxes [41]. Each bounding box consists of 5 predictions: x, y, w, h, and confidence. The coordinates x, y represent the box centre relative to the bounds of the grid cell. The height (h) and width (w) are predicted relative to the full image. The conditional class probabilities $P_r(Class_i \mid Object)$ for each grid cell are conditioned on the grid cell containing an object. The class confidence scores for each box $P_r\ (Class_i) \times IOU_{pred}^{truth}$ are given as follows [41]:

$$P_r\ (Class_i|Object) \times P_r\ (Object) \times IOU_{pred}^{truth} = P_r\ (Class_i) \times IOU_{pred}^{truth} \quad (1)$$

The problems associated with this model's instability and training divergence were raised when grid cells that did not have any object to push moved the confidence scores of these cells towards zero. Therefore, to address this problem, two parameters $\lambda coord = 5$ and $\lambda noobj = 0.5$ were used to increase the loss from the bounding box coordinate predictions and decrease the loss from confidence predictions for boxes that do not contain objects [41].

Although YOLO can identify objects in images, it fails to localise some objects, especially small ones, since its loss function treats errors equally in small and large bounding boxes [41]. The previous matter has an impact on the intersection over union (IOU) of small boxes. Therefore, to address this problem, an error metric should consider the small

deviations in small boxes that matter more than those in large boxes. As such, the square root of the bounding box width and height is predicted.

The multi-part loss function is optimised during training as follows [41]:

$$\begin{aligned}
&\lambda coord \sum_{i=0}^{s^2} \sum_{j=0}^{B} II_{ij}^{obj} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right] \\
&+\lambda coord \sum_{i=0}^{s^2} \sum_{j=0}^{B} II_{ij}^{obj} \left[\left(\sqrt{w_i} - \sqrt{\hat{w}_i} \right)^2 \right. \\
&\left. +(\sqrt{h_i} - \sqrt{\hat{h}_i})^2 \right] + \sum_{i=0}^{s^2} \sum_{j=0}^{B} II_{ij}^{obj} \left(C_i - \hat{C}_i \right)^2 \\
&+\lambda noobj \sum_{i=0}^{s^2} \sum_{j=0}^{B} II_{ij}^{noobj} \left(C_i - \hat{C}_i \right)^2 \\
&+ \sum_{i=0}^{s^2} II_i^{obj} \sum_{c \in classes} (p_i(c) - \hat{p}_i(c))^2
\end{aligned} \tag{2}$$

where II_i^{obj} refers to the existence of an object in cell i and II_{ij}^{obj} refers to the role of jth bounding box predictor in cell i.

It is obvious that the loss function only punishes the error of classification if an object exists in the grid cell [41]. In addition, it punishes an error of the box coordinate if the predictor is responsible for the ground truth.

YOLOv2 was found to improve the performance of its first version by adding several new features as follows [42]:

1. Batch normalisation to enhance convergence.
2. Fine tuning the classification network on ImageNet at a higher resolution 448 × 448 instead of 224 × 224.
3. Using anchor boxes to predict bounding boxes and removing the fully connected layers from YOLO. This plays an important role to enhance recall.
4. Shrinking the network to use 416 × 416 input images.
5. k-means clustering of the training set of the boxes' dimensions to find good priors automatically instead of by hand.
6. To overcome the model's instability resulting from anchor boxes, location prediction is directed in making the network more stable.
7. Concatenating the higher and lower resolution features by stacking them into different channels instead of spatial locations.
8. Training with a variety of input dimensions.

Most detection models are based on a VGG-16 network which requires 30.69 billion floating-point operations for forwarding only one 224 × 224 image. To increase the speed, YOLO [42] proposes a custom backbone, similar to GoogleNet [20], which requires only 8.52 billion operations, though its accuracy is slightly worse than VGG-16. On the other hand, YOLOv2 proposes Darknet19, which is like VGG having 19 convolutional layers and five max-pooling layers [42].

YOLOv3 transcends YOLOv2 as a good object detector [43], given it is both fast and accurate. Moreover, it is also very good for the detection metric of 0.5 IOUs. In addition, it is just as accurate as single shot detector (SSD), but three times faster. Though YOLOv3 is slower than YOLOv2 since it incorporates residual blocks, skips connections, and up-sampling to outperform YOLOv2 [43]. For detection performance, YOLOv3 predicts ten-fold the number of boxes predicted by YOLOv2 and predicts an abjectness score for each bounding box using logistic regression. The network in YOLOv3 is a hybrid approach between the network used in YOLOv2, Darknet-19, and residual network stuff.

Furthermore, it has 53 convolutional layers and is thus called Darknet-53 having a similar performance to ResNet-152 and two-fold faster [43].

In this paper, we proposed to use YOLOv3 with Darknet 19 network for human detection. The input image was resized to 416 × 416 before being applied to the detector. YOLO3 was selected to balance the trade-off between the accuracy of detection and speed. In addition, the pilot study and experiments performed with YOLO validated that YOLO was a good candidate to detect nude people in various scales and backgrounds. The model was tuned to filter and detect only persons and ignore other classes.

2.2.2. CNN-Based Feature Extraction

In this section, we demonstrate the transfer learning approach, which is summarised by training CNNs with a large-scale dataset and utilising the trained network with the proposed nudity dataset. In addition, various CNNs were demonstrated to provide further insight into the functionality of each network and its advantages and drawbacks. Two main types of CNN learning were sequentially performed as follows:

- Supervised CNN Model

This CNN is an end-to-end learning model. During training, the images, and labels (classes) are available and used to fine-tune the parameters of the whole network. The network consists of convolutional, pooling, batch normalisation, dropout, and fully connected (top) layers. The objective is to fit the large-scale dataset of ImageNet. The SGD was used to tune the parameters. This scenario is called feature learning. At the end of the training, the optimal parameters, which best fit the ImageNet images, and mapping them to 1000 categories, were generated and ready to be transferred to a new dataset.

- Pre-trained CNN Model

The previous CNNs, already trained on ImageNet dataset, were utilised after removing the top layers. The objective is to use the parameters of the first layers to extract features from the proposed nudity dataset. This scenario is called feature extraction. Various classifiers replaced the removed top layers in order to tune their parameters to fit the nude images and map them into two categories, including nude and normal.

In this paper, CNN-based learning was transferred to the nudity detection task. We explored various architectures of CNN such as AlexNet with its fewer layers [18], GoogleNet [20], and VGG16 [19] with their deeper layers, ResNet50 and ResNet101 with their very deep layers [22]. The comparison between these architectures was then carried out. In this work, the previously mentioned pre-trained CNNs were utilised to extract features from image patches that had only persons. The input images were converted from RGB to BGR, then each colour channel was zero-centred with respect to the ImageNet dataset, without scaling. After that, images were resized to 227 × 227 in AlexNet, 299 × 299 in Inception3, and 224 × 224 in Vgg16, GoogleNet, ResNet50, and ResNet101. The dimensions of the extracted features differ from CNN to another as follows: 4096 in AlexNet and VGG16, 1024 in GoogleNet, and 2048 in Inception3, ResNet50, and ResNet101.

2.2.3. Various CNN Architectures

- AlexNet

This CNN network has five convolutional layers, few max-pooling layers, and three fully connected layers. The last classification layer has Softmax activation with 1000 categories. The network includes 60 million parameters [18]. In order to reduce overfitting, the dropout layer was added to the top layers as a regularisation technique. In this paper, AlexNet was utilised as a feature extractor without tuning the network's parameters to extract 4096 features from each patch of image patches. These patches have only persons and were extracted from the whole frame using YOLO3. Each patch was resized to 227 × 227 pixels.

- VGG

This network has 16 weight layers and is scaled up to 19 layers. It demonstrates the advantage of representation depth to enhance classification accuracy. It was found under this architecture that increasing depth, using a very small (3×3) convolution filters, showed a significant improvement compared to previous configurations [19]. The depth was achieved by pushing the depth to 16–19 weight layers. VGGNet uses about three-fold more parameters than AlexNet. In this paper, VGG16 was utilised as a feature extractor without tuning the network's parameters to extract 4096 features from each patch of image patches. Each patch was resized to 224×224 pixels.

- GoogleNet

This network is a 22-layer-deep network with increased depth and width of the network [20]. GoogleNet used about 12 times fewer parameters than AlexNet. The inception model contains a series of fixed Gabor filters of various sizes to work with several scales. In addition, all filters in the inception model are learned. Furthermore, inception layers are repeated many times to get 22 layers of the GoogleNet [20]. A computational cost of 1.5 billion multiply–adds at inference time. In this paper, we utilised GoogleNet as a feature extractor without tuning the network's parameters to extract 1024 features from each patch of image patches. Each patch was resized to 224×224 pixels.

- Inception3

The network aims to balance the width and depth of the network. Inception includes less than 25 million parameters and has a computational budget of 5 billion multiply–adds per inference [21]. The computation of Inception is less than VGGNet [19]. Therefore, Inception networks can be used in big-data scenarios where large-scale data are needed to be processed at a reasonable, if not an affordable cost. It is also efficient in the scenario of limited memory or computational resources such as mobile vision. Inception3 combines fewer parameters, batch-normalised regularisation, and label-smoothing for training high-quality networks on modest-sized training sets [21]. Although Inception3 has 42 layers, the computation cost is only about 2.5 higher compared to GoogleNet and is still more efficient than VGGNet [21]. In this paper, Inception3 was utilised as a feature extractor without tuning the network's parameters to extract 2048 features from each patch of image patches. Each patch was resized to 299×299 pixels.

- ResNet

ResNet is a residual learning framework to ease the training of very deep networks [22]. The layers are reformulated as learning residual functions with reference to the layer inputs. With 152 layers, ResNet is eight times deeper than that of VGG nets and less complex. It has many versions, such as ResNet50, 101, and 152 [22]. ResNet is a supervised CNN model that has already been trained on large-scale datasets such as ImageNet.

In this paper, ResNet50 and ResNet101 were utilised as feature extractors without tuning the network's parameters to extract 2048 features from each patch of image patches. Each patch was resized to 224×224 pixels.

2.2.4. Classification

In this paper, CNN's top layers were replaced by various classifiers to determine the best classifier having the highest accuracy and F1 score. The classifiers were trained to fit the features extracted from the proposed nudity dataset and map them into two categories: nude and normal. The following classifiers replaced the last fully connected layers of pre-trained CNNs:

1. Support vector machine (SVM) with different kernel functions such as linear, Gaussian, and polynomials [23].
2. Extreme learning machine (ELM) [46].
3. K nearest neighbour (KNN) [44].
4. Random forest (RF) [45].

- Support Vector Machine (SVM)

SVM is a supervised learning model that is normally utilised for classification purposes. The SVM has many versions, of which the simplest version has a linear kernel and is linearly separable. It is used when the relationship between data points is linear. On the other hand, various non-linear kernel functions, such as Gaussian and polynomial, are available [23].

SVM takes multiple feature vectors as inputs and produces numerous hyperplanes in many dimensions. The best or optimal hyperplane, called the decision boundary, separates the feature vectors to maximise the margins from both vectors. In other words, the model is trained to select the hyperplane whose distance to the nearest vector is the largest. The loss function should be minimised as follows [48]:

$$min_{w,b} \frac{1}{2} w^T w + C \sum_{i=1} \max\left(0, y_i \left(w^T \phi(x_i) + b\right)\right) \quad (3)$$

where W is a weight vector, b is a bias vector, ϕ is the identity function, and C is a regularisation constant.

Usually, the kernel is linear and gives a linear classifier: $K(x, x') = x^T x'$. However, non-linear kernels produce non-linear classifiers without data transformation. The dot product is applied to the map, using the ϕ function, and the current space to a higher dimensional space for non-linear data classification [48]. The kernel computes the inner product between two functions as follows [48]:

$$K(x_i, x_j) = \phi(x_i)^T \phi(x_j) \quad (4)$$

The output of the decision function after solving the optimisation problem has a sign that determines the predicted class, which is calculated by the sum of all support vectors for samples within the margin; where x is a given sample, α is the dual coefficient and equals zero for the samples outside the margin as follows [48]:

$$\sum_{i \in SV} y_i \alpha_i K(x_i, x) + b \quad (5)$$

Radial basis function (RBF) or Gaussian kernel was used as a non-linear kernel. Training an SVM requires tuning C and gamma. When C impacts on the decision surface, high C makes correct classification, whereas low C makes the decision surface smooth. In addition, gamma determines the impact of a single training example. A small value of gamma makes the model constrained and unable to capture the complexity of the data. The kernel is calculated as follows [48]:

$$K(x, x') = \exp\left(- \| x - x' \|^2 / 2\sigma^2\right) \quad (6)$$

where σ is the standard deviation.

The polynomial kernel was used as another non-linear kernel as follows [48]:

$$K(x, x') = \left(1 + x^T x'\right)^d \quad (7)$$

where d signifies the degree.

- Extreme Learning Machine (ELM)

ELM is a fast and single hidden layer feedforward neural network that has good generalisation. The number of hidden nodes is considered a hyperparameter to be selected manually [46]. In this architecture, the input weights and biases are generated randomly.

On the other hand, the output weights, that link the output layers to the hidden layers, are calculated analytically as follows [46]:

$$f(x) = \sum_{i=1}^{L} F_i(x, w_i, b_i).\ \beta_i, \quad w_i \in R^d,\ b_i,\ \beta_i \in \mathrm{R} \tag{8}$$

where $F_i(.)$ is an activation function of the ith hidden node, w_i is an input weight, b_i is a bias, L neurons are used in the hidden layer, and β_i is the weight applied to the output as follows [46]:

$$\beta = U^{+}\ T \tag{9}$$

$$\beta = U^T \left(\frac{1}{\lambda} + U\ U^T\right)^{-1}\ T \tag{10}$$

U is a hidden layer output matrix, U^+ is the Moore–Penrose generalised inverse of U, T is a target matrix that has one hot encoded label for ELM binary classifier, and λ is a regulation coefficient.

ELM reduces the overfitting problem and enhances the learning speed more so compared to gradient-based methods [46]. However, the drawback of ELM is the randomness of input parameters.

- Random Forest (RF)

RF is supervised ensemble learning that combines multiple tree predictors for classification [45]. RF is used to reduce the overfitting problem. All trees in the forest have the same distribution, and each one is based on the values of a random vector sampled independently. The generalisation performance of forests converges better when the number of trees becomes larger [45]. The generalisation is also based on the power of each tree individually and the correlation between trees. Figure 9 shows the architecture of the RF.

Many hyperparameters are selected carefully to improve RF performance. The number of estimators or trees has an important impact; when the number of trees is increased, the performance is enhanced and makes the predictions more stable. However, more trees reduce computation time. Max features is another hyperparameter that refers to the maximum number of features to consider splitting a node [48].

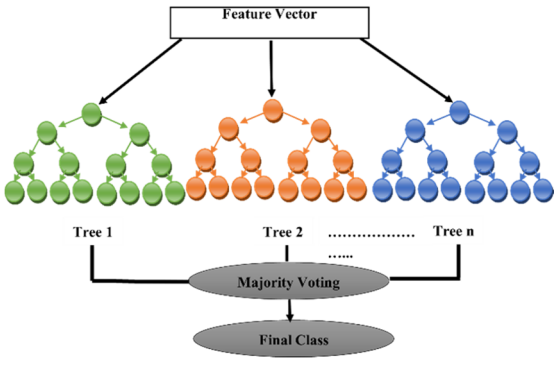

Figure 9. Random forest architecture.

- K Nearest Neighbour (KNN)

KNN is a non-parametric technique and a simple classifier based on the similarity measure such as distance functions between data points. KNN depends on the approach that considers similar data points are closed to each other in order to capture the similarity or closeness of the data points. Simple mathematics is used to calculate the distance between the data points on a graph. K, which refers to the number of neighbours, is a

hyperparameter to be selected because it impacts on the errors and accurate predictions. The advantages of KNN are its simplicity and ease of implementation.

In addition, there are no hyperparameters to be tuned in advance [44,48]. On the other hand, KNN speeds down significantly when the number of samples and independent variables is increased.

KNN algorithm has many steps [44]:

1. Initialise the number of neighbours K.
2. The distance between the unseen sample and each sample in the dataset is calculated.
3. Store the distance and the index of the sample in a buffer.
4. Order the distances and indices inside the buffer in ascending order.
5. Pick the first K sample from the sorted buffer.
6. Check the labels of the K first samples.
7. For the classification task, the mode of the K labels is found and is considered as the predicted category.

2.2.5. The Proposed YOLO–CNN–SVM Method

Several traditional CNN-only methods were utilised in the state-of-the-art methods for pornography detection [7,9]. In these methods, the whole frames, including multiple objects, were resized, and applied directly to one of the pre-trained CNNs to extract the features and classify them by the attached classifier. The classifier gives a normal or nude category at the output. On the other hand, the proposed method consists of three main blocks, including pre-trained YOLO3, pre-trained CNN, and a classifier. The first block of YOLO3 was used for human detection in order to check if persons are available in the frames or not. The frames may have one, two, or a group of persons. The persons appear in various positions, sizes, and backgrounds. Some frames include persons wearing cloth with various colours, while other frames have nude persons with various skin colours. Another challenge is the overlapping of persons in the frames. After the persons were detected, and boundary boxes were drawn around them, the patches of images surrounded by boxes were extracted and resized to fit the input dimensions of the pre-trained CNN. The CNN-extracted features from patches have persons (regions of interest (ROIs)) to be classified into normal and nude. The flow chart of the proposed method and the CNN-only method is shown in Figure 10.

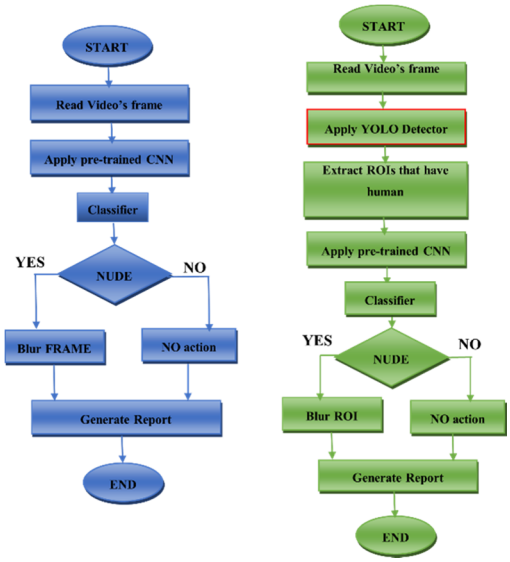

Figure 10. Flow chart of the proposed method (on the right) and the state-of-the-art CNN-only (on the left) for nudity and pornography detection.

If the frame comprises more than one person, each person image patch is passed to CNN to produce one category that was stored in a list. The list has labels of all person patches in one frame. If one element in the list has a nude label, the whole frame is considered nude. However, if all labels in the list have normal elements, the whole frame is considered normal. The advantage of this method is the ability to censor (edit or blur) specific regions in the frames instead of blurring or cutting the whole frames. This technique, in contrast to CNN-only, keeps the frames in the movie videos and reduces the interrupted period of the film, which helps to understand the context.

3. Experimental Setup and Results

In this section, an ablation study to validate the impact of adding YOLO3 before CNN is elaborated. Furthermore, we evaluate and compare various pre-trained CNNs and classifiers for the nudity classification task. Several performance metrics such as accuracy, recall, precision, false negative rate (FNR), false positive rate (FPR), F1 score, and area under curve (AUC) were utilised to evaluate the models. The experiments were performed on a desktop computer installed with Windows 10 (64-bit OS), 64.0 GB RAM, Nvidia GeForce GTX 1080 Ti, 12 GB GPU.

3.1. Performance Metrics

Multiple performance metrics were used to validate the performance of the machine learning model. Accuracy, F1 score, AUC, FPR, and FNR are important factors that should be considered to validate the nude/normal classification model.

The summary of performance mercies is as follows:

1. Accuracy is a measure that calculates the number of samples predicted correctly over all available samples.

$$\text{Accuracy } = \frac{TP + TN}{TP + TN + FP + FN}$$

where *TP*: True positive, *TN*: True negative, *FP*: False positive, *FN*: False negative.

2. Precision is a measure that calculates the number of samples predicted correctly as nude over all the samples predicted as nude:

$$\text{Precision} = \frac{TP}{TP + FP}$$

3. Recall (Sensitivity) is a measure that calculates the number of samples predicted correctly as nude over all actual nude samples:

$$\text{Recall } = \frac{TP}{TP + FN}$$

4. Negative predictive value (NPV) is a measure that calculates the number of samples predicted correctly as normal over all samples predicted as normal:

$$\text{NPV} = \frac{TN}{TN + FN}$$

5. False negative rate (FNR) (1-Recall) is a measure that calculates number of samples predicted wrongly as normal over all the actual nude samples:

$$\text{FNR} = \frac{FN}{TP + FN}$$

6. False positive rate (FPR) (1-specificity) is a measure that calculates the number of samples predicted incorrectly as nude over all actual normal samples:

$$\text{FPR} = \frac{FP}{TN + FP}$$

7. F1 score: this metric summarizes *recall* and *precision* in one term:

$$\text{F1 score} = \frac{2 \times precision \times recall}{precision + recall}$$

8. Area under curve (AUC): This metric is calculated after plotting receiver operating characteristic (ROC) curve given false positive rate as a horizontal axis and true positive rate as a vertical axis. It reflects how much a binary classifier is robust when the threshold is varied.

3.2. Experiments

3.2.1. The First Experiment

We began by comparing the model of ResNet50-average pooling strategy [10] with ACORDE-50 [10] on the 2K extended version of NPDI, which was already divided into five-fold for training and validation. It was found that the ResNet50-average pooling strategy gave an average accuracy of 95.04%, which was higher than 92.43% average accuracy of ACORDE-50. Therefore, we selected ResNet50-average pooling to train the model that would be used to infer the frames in the testing film dataset.

To evaluate the exiting CNN-only methods [7,10] with the testing film dataset, we conducted the experiment using the same model's architecture (ResNet50+SVM) that was used in [10]. We used the ResNet50 CNN-only method, which was already trained on the ImageNet dataset [16], to extract the features directly from the frames of the extended version of the NPDI Pornography dataset [47]. Additionally, SVM was trained to classify extracted features as normal and nude. The previous stage is called the training stage. On the other hand, in the testing stage, the frames of the testing film dataset were applied to ResNet50 to extract the features and utilise the already trained SVM to categorise them. The performance metrics were then calculated.

Unfortunately, the existing methods of CNN-only, that were trained on NPDI, were unable to detect nudity or pornography in the video's frames that had specific content such as nude people with various positions, scales, and backgrounds. Table 3 shows the experimental results. The high FNR was caused by the misclassification of nude frames since they have nudity regions in small-scale with complex backgrounds. In other words, most of the frames were classified as normal.

Table 3. Performance metrics for ResNet50 CNN-only [10] with a testing film dataset.

Video ID	Precision%	FNR%	FPR%	NPV%	F1-Score%
1	97	65.8	68.2	11.2	50.57
2	62.8	80.8	7.9	62.3	29.41
3	-	100	0	68.3	-
4	88.7	90.4	5.5	18.8	17.32
5	90.5	37.9	35.8	23.6	73.66

3.2.2. The Second Experiment

In this experiment, the existing method [10] of ResNet50-only, which was transferred to the NPDI dataset and used SVM instead of top layers, was compared with the proposed method of YOLO-ResNet50-SVM.

The second experiment explored the limitations of the existing convolutional-only approaches [10] that apply CNN directly on the whole frames of videos and the limitations of the NPDI dataset [47] in the task of small-scale nudity detection. Furthermore,

it proposes to utilise an object detector such as YOLO to focus CNN's attention on the nudity region inside the frame. The outcome is a model fusion that combines YOLO and ResNet50 CNN. In this proposed method, in the training stage, a collected set of human-only images was used to train the ResNet50+SVM [10]. In the testing stage, the testing film dataset was applied to YOLO that was trained on the COCO dataset [17] to detect humans inside the frames. After that, the detected patches, that contain humans, were applied to the already trained ResNet50+SVM to decide if this frame is nude or not. If the frame has at least one person labelled as nude, the whole frame is considered nude, whereas if all persons have normal labels, the whole frame is considered normal. Table 4 shows the performance metrics for YOLO+ResNet50. Finally, we compared the proposed method of YOLO+ResNet50 with ResNet50-only, as shown in Figure 11. The proposed YOLO+ResNet50 was found to outperform ResNet50-only regarding accuracy and the F1-score in all five videos. Moreover, the FNR in YOLO+ResNet50 was much lower than ResNet50-only as shown in Tables 3 and 4.

Table 4. Performance metrics for YOLO + ResNet50 CNN with testing film dataset.

Video ID	Precision%	FNR%	FPR%	NPV%	F1-Score%
1	94.6	0	90.9	100	97.23
2	73	10	23.1	91.8	80.61
3	86.9	0	7	100	92.99
4	87.5	4.1	61.7	67.4	91.51
5	87.7	2.1	75.5	68.4	92.52

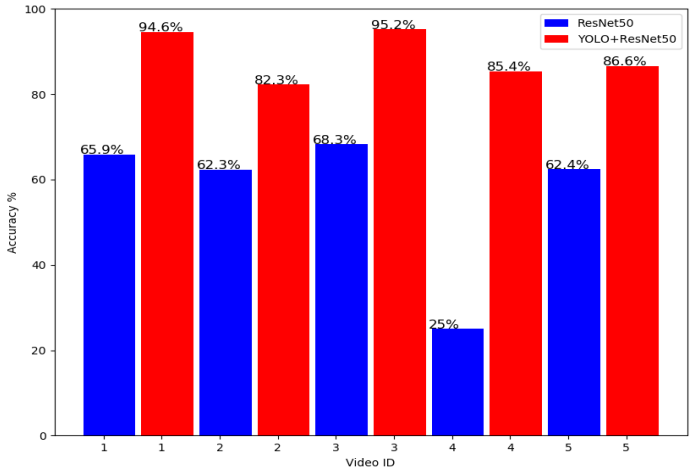

Figure 11. Comparison between the proposed YOLO-ResNet50 and ResNet50-only [10] in terms of accuracy using the testing film dataset.

The big difference between the performance of the proposed method and the ResNet50-only [10] is that the latter was trained on pornographic frames that have big scales of nudity with fewer backgrounds. On the other hand, the real-world video contents that are common to be broadcasted on TV often contain more nudity with different scales and various backgrounds. Therefore, our proposed method plays an important role in this type of visual content to focus the attention on the expected nude regions before classifying them as nude or not.

3.2.3. Ablation Study

In this study, we gained insight into the effect of adding the YOLO model before pre-trained CNN in the pipeline. The comparison was performed between ResNet50-only [10], and YOLO-ResNet50 using the MMU collected dataset which includes variations in human position, the size relative to frame's size, cloth, skin colour, and backgrounds such as forest,

snowing, beach, supermarket, indoor, and streets. The data were divided into train and validation sets, as mentioned in Section 2.1.4.

K = 5 cross-validation was performed to validate the improvement achieved by adding YOLO in the pipeline. The accuracy for each fold and average accuracy were calculated for both methods, as shown in Table 5. YOLO–CNN was found to outperform CNN-only regarding the accuracy, which increased from 85.5% to 89.5%. In addition, Table 5 demonstrates the impact of augmentation, as described in Section 2.1.4 to improve accuracy by 2%. The confusion matrix of each fold for both methods is shown in Figure 12.

Table 5. K = 5 cross-validation to compare ResNet50-only and YOLO + ResNet50 in terms of accuracy.

5 Cross-Validations	ResNet50 (No AUG) Accuracy%	ResNet50 (AUG) Accuracy%	YOLO-ResNet50 (AUG) Accuracy%
K = 1	82.500	82.500	91.250
K = 2	89.375	91.250	91.875
K = 3	84.375	87.500	85.625
K = 4	78.125	81.875	89.375
K = 5	84.375	84.375	89.375
Average %	83.500	85.500	89.500

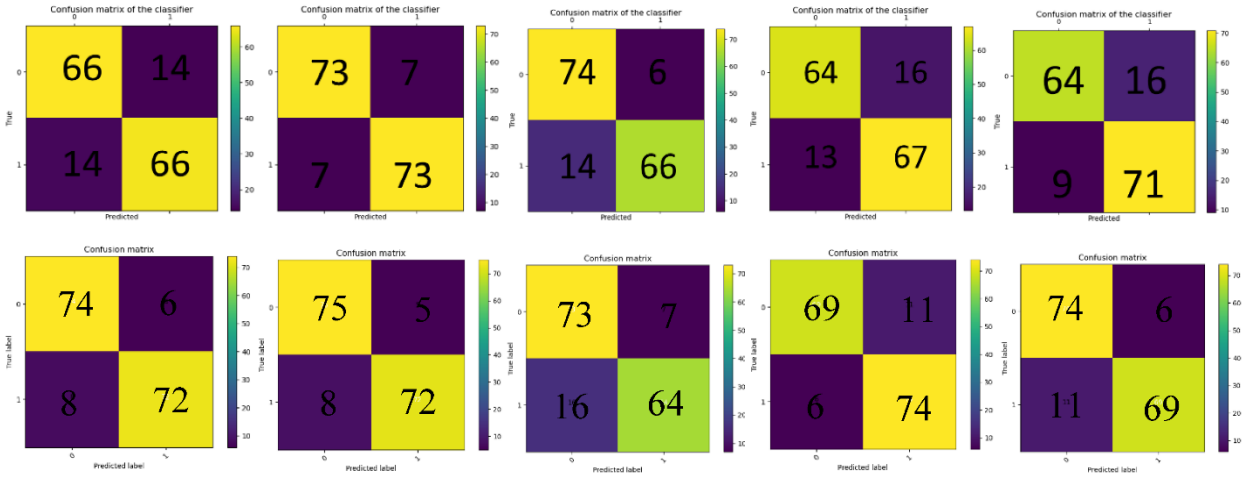

Figure 12. Confusion matrix of CNN-only (first row) and YOLO–CNN (second row) for each fold.

The accuracy metric is not adequate to validate the results. Therefore, the ROC curve was drawn to measure the performance of the two methods. It was found that the mean AUC of YOLO–CNN is 97 % which outperforms CNN-only by 4%. ROC is illustrated in Figure 13. The blue curve represents YOLO–CNN, whereas the green curve represents CNN-only. In addition, performance metrics were calculated for both ResNet50-only and YOLO + ResNet50 as shown in Table 6.

The previous results in the ablation study validated the correctness of the hypothesis that was given in this paper. This concludes that attracting the attention of CNN-only on expected nudity regions in frames helps to classify these regions better and improve the performance of nudity classification. On the other hand, the results highlighted the limitation of other methods that make CNN take all various content and objects in the frame.

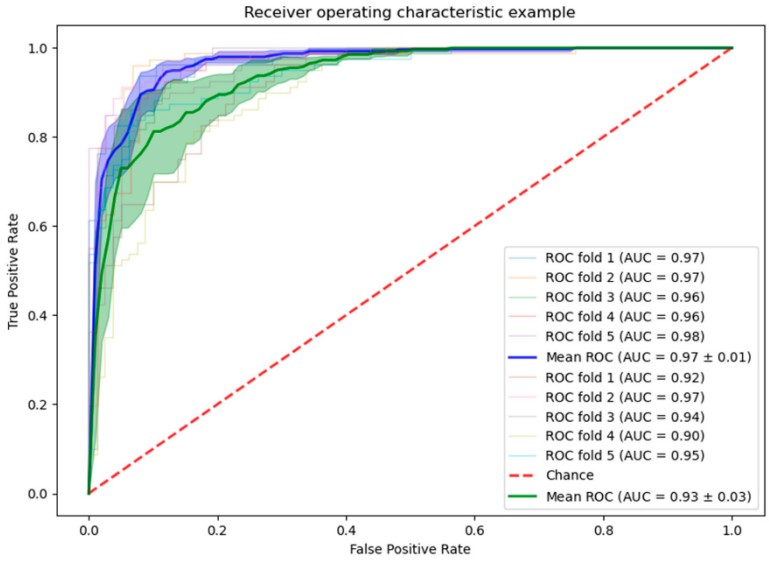

Figure 13. Receiver operating characteristic (ROC) curves of YOLO–CNN and CNN-only using the MMU dataset.

Table 6. Performance metrics to compare ResNet50-only and YOLO + ResNet50.

Performance Metric	ResNet50 (AUG)	YOLO-ResNet50 (AUG)
Average Precision	85.5498	91.0028
Average Recall	85.7500	87.7500
Average FNR	14.2500	12.2500
Average FPR	14.7500	8.7500
Average NPV	85.7258	88.3373
Average F1 score	**85.5661**	89.2713

3.2.4. The Third Experiment

In the previous section, an ablation study was conducted to validate the efficiency of adding YOLO3 before ResNet50 CNN to have three blocks in the pipeline, namely YOLO3, ResNet50, and SVM. In this section, the objective is to combine YOLO3 with other architectures of CNNs and classifiers. Therefore, an experiment was carried to utilise a combination of one of six feature extractors and one of six classifiers to have in total 6×6 models. In order to evaluate the performance of the 36 models, each feature extractor and classifier was tested on 25,983 human images detected from 8579 testing images, as mentioned in Section 2.1.5. The image is considered nude if any part of it contains nudity. Table 7. shows the performance metrics for all 36 models.

The results were compared with similar work presented in [7,10]. The authors in [7] employed AlexNet and GoogleNet with SVM for the pornographic dataset. In [10], they utilised ResNet50 and ResNet101 with Linear SVM. Therefore, in this paper, we replicated their models on our dataset to compare with the proposed feature extractors and classifiers. The best of the proposed models based on the F1-score were in the following order: ResNet101-RF (90.03%), ResNet101-GSVM (89.97%), and ResNet50-KNN (89.96%), as shown in Figure 14. The top three accuracies of 87.75%, 87.42%, and 87.66% respectively, which were underlined in Table 7, outperform the accuracies of state-of-the-art models including AlexNet-LSVM [7] (83.81%), GoogleNet-LSVM [7] (84.99%), ResNet50-LSVM [10] (84.94%), and ResNet101-LSVM [10] (86.53%).

The settings of the classifiers were as follows: SVM was trained with three different kernels: Linear, Gaussian, and Polynomial. For RF, a different number of trees was tested. It was found that 100 trees produced the best performance with the testing dataset. On

the other hand, to validate the impact of the distance function in KNN, different functions were evaluated, and Euclidean distance was nominated to be utilised in this experiment. In addition, a different number of neighbours was tested in selecting the best classifier, which was 30 nn to be employed in the experiments. Furthermore, the hyperparameters of ELM were also investigated to optimise the results. ELM, with 9000 hidden nodes and 2^{15} regulation factors, was the best candidate.

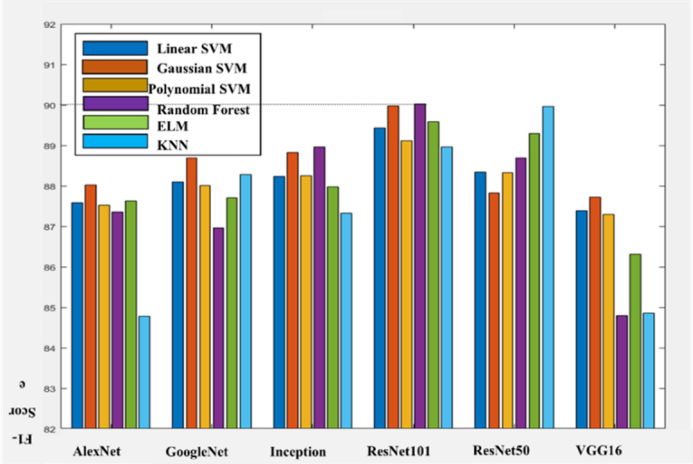

Figure 14. Comparison between the various feature extractors and classifiers in terms of the F1-score.

Table 7. Performance metrics for various pre-trained CNNs and classifiers.

Feature Extractor	Classifier	LSVM [7,10]	GSVM (Proposed)	PSVM (Proposed)	KNN (Proposed)	RF (Proposed)	ELM (Proposed)
AlexNet [7]	Accuracy (%)	83.8093	84.6136	83.8210	79.3799	84.0657	83.8326
	Precision (%)	80.2326	81.6636	80.5257	75.3211	82.3971	80.1207
	FNR (%)	3.5820	4.5463	4.1330	3.0506	7.0459	3.3064
	FPR (%)	34.5054	31.1321	33.6764	46.1407	28.8451	34.8485
	NPV (%)	7.3595	8.7500	8.3004	7.6018	12.5746	6.8655
	F1 score (%)	87.5838	88.0218	87.5292	84.7775	87.3578	87.6305
VGG16 (proposed)	Accuracy (%)	84.1940	84.5437	83.9375	80.0443	81.2566	82.6786
	Precision (%)	82.8541	82.8177	82.0940	77.0863	81.6130	81.1040
	FNR (%)	7.5576	6.7506	6.7900	5.6485	11.7693	7.7544
	FPR (%)	27.7873	28.1018	29.5312	40.7376	28.8736	31.2178
	NPV (%)	13.1959	12.0014	12.2776	12.1610	19.3778	14.0714
	F1 score (%)	87.3860	87.7245	87.2996	84.8495	84.7929	86.3168
GoogleNet [7]	Accuracy (%)	84.9866	86.2805	84.9400	85.0332	84.2522	84.3222
	Precision (%)	83.0573	86.6779	83.2543	82.2929	85.2819	81.8771
	FNR (%)	6.2192	9.2108	6.6522	4.7825	11.2773	5.5698
	FPR (%)	27.7873	20.2687	27.2727	29.7599	22.2413	30.3602
	NPV (%)	11.1189	14.3691	11.7280	9.0000	17.4005	10.4082
	F1 score (%)	88.0939	88.6859	88.0126	88.2847	86.9683	87.7068
Inception3 (proposed)	Accuracy (%)	85.5461	86.5602	85.4528	83.7860	86.5952	85.3363
	Precision (%)	85.1804	87.4952	84.6001	81.2924	86.8416	85.5099
	FNR (%)	8.4826	9.8012	7.7741	5.6682	8.8172	9.4076
	FPR (%)	23.1275	18.7250	24.3854	31.5323	20.0686	22.2985
	NPV (%)	13.8141	14.9057	12.9934	10.7343	13.8101	14.9562
	F1 score (%)	88.2353	88.8264	88.2486	87.3280	88.9593	87.9778

Table 7. *Cont.*

Feature Extractor	Classifier	LSVM [7,10]	GSVM (Proposed)	PSVM (Proposed)	KNN (Proposed)	RF (Proposed)	ELM (Proposed)
ResNet50 [10]	Accuracy (%)	84.9400	84.0774	84.8584	**87.6559**	85.7443	86.3154
	Precision (%)	81.5803	80.2771	81.3079	86.7373	83.6180	83.1495
	FNR (%)	3.6804	3.0703	3.3458	6.5538	5.5698	3.5623
	FPR (%)	31.5895	34.5912	32.2756	20.7547	26.8725	28.3877
	NPV (%)	7.2481	6.3830	6.6955	10.7246	9.9613	6.7386
	F1 score (%)	88.3394	87.8210	88.3194	**89.9669**	88.6958	89.3020
ResNet101 [10]	Accuracy (%)	86.5252	**87.4228**	86.1056	86.7001	**87.7492**	86.8400
	Precision (%)	83.5184	85.2289	83.1656	87.4952	86.9048	84.3652
	FNR (%)	3.7591	4.7235	4.0346	9.5257	6.6129	4.5267
	FPR (%)	27.5872	23.9851	28.2161	18.7822	20.4403	25.7004
	NPV (%)	7.0117	8.2787	7.5479	14.5564	10.7727	8.1301
	F1 score (%)	89.4294	**89.9731**	89.1082	88.9598	**90.0294**	89.5762

3.2.5. Class Activation Mapping

The global average pooling layer gives the classification-trained CNN ability to localize well despite being trained on image-level labels (without bounding box) [49,50]. Class activation maps (CAMs) help to visualize the scores of predicted classes. In other words, the discriminative object parts in an image are highlighted after being detected by the CNN. CAM has been used in many applications such as medical imaging task [50] to understand the predictions. In this paper, CAM was applied to few images in the proposed dataset to highlight discriminative regions in the images as shown in Figure 15. The patterns were extracted from global average pooling layers to identify the complete extent of the objects.

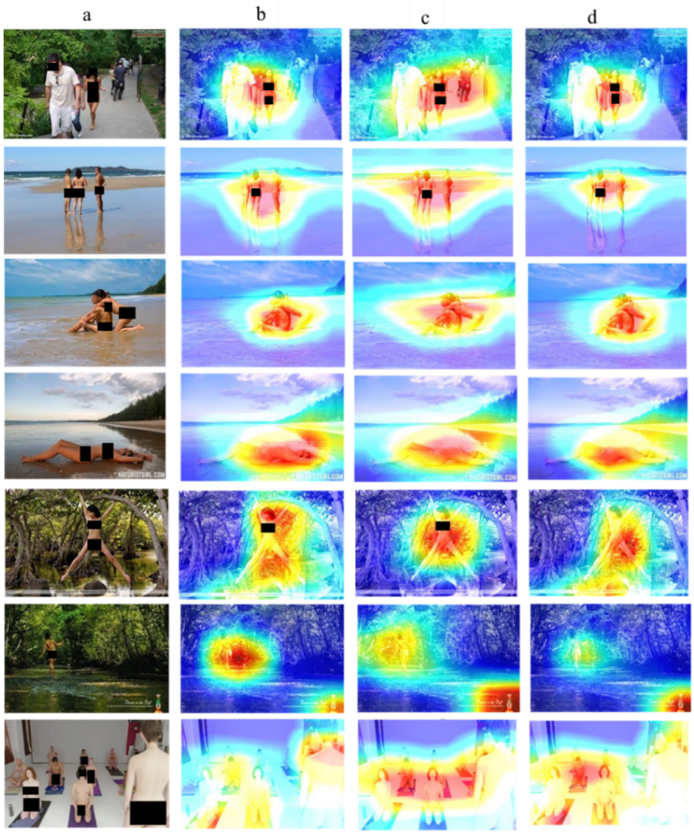

Figure 15. CAM for few samples (**a**) input image; (**b**) ResNet50 CAM; (**c**) InceptionV3 CAM; and (**d**) ResNet101 CAM.

4. Conclusions and Future Work

In this paper, the problem associated with nudity detection at various scales and backgrounds was addressed. The proposed method utilised COCO-trained YOLO3 detector, which was transferred to our dataset to determine the regions of interest that included human patches. In addition, ImageNet-trained CNNs such as AlexNet, GoogleNet, VGG16, Inception3, and ResNet were transferred to extract the features automatically from human patches and classify them by a classifier into two categories: normal and nude. The proposed method of YOLO–CNN was compared to CNN-only and was found to improve the accuracy by 4% from 85.5% to 89.5%. In addition, AUC was also increased from 93% to 97%.

Various CNN-based feature extractors and classifiers were utilised and compared. ResNet101-RF was found to improve the performance of the detection system and outperform state-of-the-art methods regarding the F1-score (90.03%) and accuracy (87.75%). It also balances the trade-off between the FNR and the FPR.

The results of this work are summarised as follows:

- YOLO is a good object detector that can be transferred to detect humans in the nudity dataset as a significant stage in the proposed nudity detection pipeline. This stage plays a role to apply CNN on ROI instead of applying CNN on the whole frame.
- Pre-trained CNN can be transferred after removing the final layers to extract features from visual nudity content.
- Various classifiers such as SVM, RF, and ELM were used to replace the final layers of CNN to fit the extracted features and classify them into normal and nude.

The advantages of the proposed system include the following:

1. The proposed system can automatically detect nude humans with various scales in complex backgrounds such as forest, snowing, beach, supermarket, and streets.
2. The proposed system runs in real-time as only one frame is sufficient to detect human patches and classify them as normal and nude. This helps to censor live-captured videos.
3. The proposed nude/normal classifier is robust against various scales, positions, cloth, and skin colour.
4. It outperforms the state-of-the-art methods, such as CNN-only regarding the accuracy, F1 score, and AUC.
5. The proposed method can also edit and blur only nude regions inside the frames. In other words, there is no need to blur or cut the whole detected frames from the video.

This work focuses on the utilisation of YOLO as a human detector. The limitation of the proposed method is related to the performance of the human detector. If the detector was unable to detect a human in the frame, the nude frames would be classified incorrectly as normal ones. However, this limitation may open the door by providing opportunities for future research to enhance the detection accuracy using better detectors such as EfficientDet [51] to increase the true positive rate of human detection.

In this work, the parameters of all layers except the top ones were frozen. Additionally, the top layers were replaced by SVM. To improve the performance, fine-tuning parameters of more layers on pornography or nudity dataset can be demonstrated [52].

The proposed solution utilized CNNs pretrained on the ImageNet-1K. In future work, Facebook's ResNeXt Weakly Supervised Learning (WSL) CNNs [53,54] could be adapted for further improvements in the current solution. WSL CNNs can act as fixed feature extractors to extract image-level features from the proposed dataset [54]. Furthermore, the advantage of fine-tuning these CNNs on ImageNet-1K could also be explored.

Finally, this paper proposed a new benchmark image dataset with more challenging content for nudity detection. In this research, a medium-scale dataset was used. Hence, in the future, we intend to further enhance this work with a larger number of samples to improve the performance of classification.

Author Contributions: Conceptualization, N.A.; methodology, N.A., M.H.L.A., and H.A.K.; software, N.A., and M.H.L.A.; validation, N.A., A.S.B.W., and S.M.; formal analysis, M.F.A.F. and J.S.; investigation, N.A. and A.S.B.W.; data curation, N.A.; writing—original draft preparation, N.A.; project administration, H.A.K.; funding acquisition, H.A.K. All authors also contributed to the review and editing. All authors have read and agreed to the published version of the manuscript.

Funding: This research was fully funded by TM R&D, Malaysia under project number MMUE/180029.

Institutional Review Board Statement: The study did not require ethical approval

Informed Consent Statement: Not applicable

Data Availability Statement: Two types of Data are available in this study. The first type is available on request from the corresponding author. The data are not publicly available because they were collected from the internet and have sensitive content. The second type of data is 3rd party data that have restrictions to availability. This data can be obtained from Sandra Avila, Nicolas Thome, Matthieu Cord, Eduardo Valle, Arnaldo de A. Araújo at "https://sites.google.com/site/pornographydatabase/" with the per-mission of the mentioned authors.

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

References

1. Nuraisha, S.; Pratama, F.I.; Budianita, A.; Soeleman, M.A. Implementation of K-NN based on histogram at image recognition for pornography detection. In Proceedings of the 2017 International Seminar on Application for Technology of Information and Communication (iSemantic), Semarang, Indonesia, 7–8 October 2017; pp. 5–10.
2. Garcia, M.B.; Revano, T.F.; Habal, B.G.M.; Contreras, J.O.; Enriquez, J.B.R. A Pornographic Image and Video Filtering Application Using Optimized Nudity Recognition and Detection Algorithm. In Proceedings of the 2018 IEEE 10th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM), Baguio City, Philippines, 29 November–2 December 2018; pp. 1–5.
3. Ries, C.X.; Lienhart, R. A survey on visual adult image recognition. *Multimed. Tools Appl.* **2014**, *69*, 661–688. [CrossRef]
4. Santos, C.; Dos Santos, E.M.; Souto, E. Nudity detection based on image zoning. In Proceedings of the 2012 11th International Conference on Information Science, Signal Processing and their Applications (ISSPA), Montreal, QC, Canada, 2–5 July 2012; pp. 1098–1103.
5. Moreira, D.C.; Fechine, J.M. A Machine Learning-based Forensic Discriminator of Pornographic and Bikini Images. In Proceedings of the 2018 International Joint Conference on Neural Networks (IJCNN), Rio de Janeiro, Brazil, 8–13 July 2018; pp. 1–8.
6. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [CrossRef] [PubMed]
7. Moustafa, M. Applying deep learning to classify pornographic images and videos. Pacific Rim Symposium on Image and Video Technology. *arXiv* **2015**, arXiv:1511.08899v1.
8. Automated Nudity Recognition using Very Deep Residual Learning Network. *Int. J. Recent Technol. Eng.* **2019**, *8*, 136–141. [CrossRef]
9. Nurhadiyatna, A.; Cahyadi, S.; Damatraseta, F.; Rianto, Y. Adult content classification through deep convolution neural network. In Proceedings of the 2017 International Conference on Computer, Control, Informatics and its Applications (IC3INA), Jakarta, Indonesia, 23–26 October 2017; pp. 106–110.
10. Wehrmann, J.; Simões, G.S.; Barros, R.C.; Cavalcante, V.F. Adult content detection in videos with convolutional and recurrent neural networks. *Neurocomputing* **2018**, *272*, 432–438. [CrossRef]
11. Perez, M.; Avila, S.; Moreira, D.; Moraes, D.; Testoni, V.; Valle, E.; Goldenstein, S.; Rocha, A. Video pornography detection through deep learning techniques and motion information. *Neurocomputing* **2017**, *230*, 279–293. [CrossRef]
12. Wang, Y.; Jin, X.; Tan, X. Pornographic image recognition by strongly-supervised deep multiple instance learning. In Proceedings of the 2016 IEEE International Conference on Image Processing (ICIP), Phoenix, AZ, USA, 25–28 September 2016; pp. 4418–4422.
13. AlDahoul, N.; Karim, H.A.; Abdullah, M.H.L.; Fauzi, M.F.A.; Mansour, S.; See, J.; Alfrahoul, N. Local Receptive Field-Extreme Learning Machine based Adult Content Detection. In Proceedings of the 2019 IEEE International Conference on Signal and Image Processing Applications (ICSIPA), Kuala Lumpur, Malaysia, 17–19 September 2019; pp. 128–133.
14. Kiefer, J.; Wolfowitz, J. Stochastic Estimation of the Maximum of a Regression Function. *Ann. Math. Stat.* **1952**, *23*, 462–466. [CrossRef]
15. Bottou, L.; Curtis, F.; Nocedal, J. Optimization Methods for Large-Scale Machine Learning. *SIAM Rev.* **2018**, *60*, 223–311. [CrossRef]
16. Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; Fei-Fei, L. ImageNet: A large-scale hierarchical image database. *IEEE Conf. Comput. Vis. Pattern Recognit.* **2009**, 248–255. [CrossRef]
17. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft COCO: Common objects in context. In Proceedings of the Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). *arXiv* **2014**, arXiv:10.1007/978-3-319-10602-1_48.
18. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet classification with deep convolutional neural networks. *Commun. ACM* **2017**, *60*, 84–90. [CrossRef]
19. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
20. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015.

21. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the inception architecture for computer vision. *Conf. Proc.* **2016**, 2818–2826. [CrossRef]
22. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
23. Hearst, M.A.; Scholkopf, B.; Dumais, S.; Osuna, E.; Platt, J. Support vector machines. *IEEE Intell. Syst.* **1998**, *13*, 18–28. [CrossRef]
24. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef] [PubMed]
25. Da Silva, M.V.; Marana, A.N. Spatiotemporal CNNs for pornography detection in videos. In Proceedings of the Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), Madrid, Spain, 19–22 November 2019; pp. 547–555. [CrossRef]
26. Tran, D.; Bourdev, L.; Fergus, R.; Torresani, L.; Paluri, M. Learning Spatiotemporal Features with 3D Convolutional Networks. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015; pp. 4489–4497.
27. Tran, D.; Wang, H.; Torresani, L.; Ray, J.; LeCun, Y.; Paluri, M. A Closer Look at Spatiotemporal Convolutions for Action Recognition. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 6450–6459.
28. NPDI Pornography Database, 2013. Available online: https://sites.google.com/site/pornographydatabase/ (accessed on 1 April 2019).
29. Avila, S.; Thome, N.; Cord, M.; Valle, E.; Araújo, A.D.A. Pooling in image representation: The visual codeword point of view. *Comput. Vis. Image Underst.* **2013**, *117*, 453–465. [CrossRef]
30. Nian, F.; Li, T.; Wang, Y.; Xu, M.; Wu, J. Pornographic image detection utilizing deep convolutional neural networks. *Neurocomputing* **2016**, *210*, 283–293. [CrossRef]
31. Liu, B.-B.; Su, J.-Y.; Lu, Z.-M.; Li, Z. Pornographic Images Detection Based on CBIR and Skin Analysis. In Proceedings of the 2008 Fourth International Conference on Semantics, Knowledge and Grid, Beijing, China, 3–5 December 2008; pp. 487–488.
32. Dollár, P.; Belongie, S.; Perona, P. The fastest pedestrian detector in the west. In Proceedings of the British Machine Vision Conference, BMVC 2010-Proceedings, Aberystwyth, UK, 31 August–3 September 2010; pp. 68.1–68.11. [CrossRef]
33. AlDahoul, N.; Sabri, A.Q.M.; Mansoor, A.M. Real-Time Human Detection for Aerial Captured Video Sequences via Deep Models. *Comput. Intell. Neurosci.* **2018**, *2018*, 1639561. [CrossRef]
34. Shinde, S.; Kothari, A.; Gupta, V. YOLO based Human Action Recognition and Localization. *Procedia Comput. Sci.* **2018**, *133*, 831–838. [CrossRef]
35. Ivašić-Kos, M.; Krišto, M.; Pobar, M. Human detection in thermal imaging using YOLO. In Proceedings of the 2019 5th International Conference on Computer and Technology Applications, Istanbul, Turkey, 16–17 April 2019; pp. 254–267. [CrossRef]
36. Simoes, G.S.; Wehrmann, J.; Barros, R.C. Attention-based Adversarial Training for Seamless Nudity Censorship. In Proceedings of the 2019 International Joint Conference on Neural Networks (IJCNN), Budapest, Hungary, 14–19 July 2019; pp. 1–8.
37. Ion, C.; Minea, C. Application of Image Classification for Fine-Grained Nudity Detection. In Proceedings of the Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), Lake Tahoe, NV, USA, 7–9 October 2019; pp. 3–15. [CrossRef]
38. Connie, T.; Al-Shabi, M.; Goh, M. Smart content recognition from images using a mixture of convolutional neural networks. In *IT Convergence and Security 2017*; Springer: Singapore, 2018; pp. 11–18. [CrossRef]
39. Shen, R.; Zou, F.; Song, J.; Yan, K.; Zhou, K. EFUI: An ensemble framework using uncertain inference for pornographic image recognition. *Neurocomputing* **2018**, *322*, 166–176. [CrossRef]
40. Li, D.; Li, N.; Wang, J.; Zhu, T. Pornographic images recognition based on spatial pyramid partition and multi-instance ensemble learning. *Knowl. Based Syst.* **2015**, *84*, 214–223. [CrossRef]
41. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
42. Redmon, J.; Farhadi, A. YOLO9000: Better, Faster, Stronger. In Proceedings of the 30th IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 6517–6525.
43. Redmon, J.; Farhadi, A. YOLOv3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
44. Altman, N.S. An Introduction to Kernel and Nearest-Neighbor Nonparametric Regression. *Am. Stat.* **1992**, *46*, 175. [CrossRef]
45. Breiman, L. Random Forests. *Mach. Learn.* **2001**, *45*, 5–32. [CrossRef]
46. Huang, G.-B.; Zhu, Q.-Y.; Siew, C.-K. Extreme learning machine: Theory and applications. *Neurocomputing* **2006**, *70*, 489–501. [CrossRef]
47. Moreira, D.; Avila, S.; Perez, M.; Moraes, D.; Testoni, V.; Valle, E.; Goldenstein, S.; Torres, R.D.S. Pornography classification: The hidden clues in video space–time. *Forensic Sci. Int.* **2016**, *268*, 46–61. [CrossRef]
48. SciKit Learn Library for Machine Learning. Available online: https://scikit-learn.org/ (accessed on 20 September 2020).
49. Zhou, B.; Khosla, A.; Lapedriza, A.; Oliva, A.; Torralba, A. Learning Deep Features for Discriminative Localization. In Proceedings of Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 2921–2929.
50. Narayanan, B.N.; Silva, M.S.D.; Hardie, R.C.; Kueterman, N.K.; Ali, R. Understanding Deep Neural Network Predictions for Medical Imaging Applications. *arXiv* **2019**, arXiv:1912.09621v1.
51. Tan, M.; Pang, R.; Le, Q.V. EfficientDet: Scalable and Efficient Object Detection. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 10778–10787.

52. AlDahoul, N.; Karim, H.A.; Mansour, S. Convolutional Neural Network-based Transfer Learning and Classification of Visual Contents for Film Censorship. *J. Eng. Technol. Appl. Phys.* **2020**, *2*, 28–35.
53. Mahajan, D.; Girshick, R.; Ramanathan, V.; He, K.; Paluri, M.; Li, Y.; Bharambe, A.; van der Maaten, L. Exploring the Limits of Weakly Supervised Pretraining. In Proceedings of the European Conference on Computer Vision. In Proceedings of the Lecture Notes in Computer Science, Munich, Germany, 8–14 September 2018; pp. 185–201. [CrossRef]
54. Flaute, D.; Narayanan, B.N. Video captioning using weakly supervised convolutional neural networks, Proc. SPIE 11511. *Appl. Mach. Learn.* **2020**. [CrossRef]