A Deep Learning Filtration Framework to Eliminate Not Safe For Work content in Digital media

MSc Research Project
Masters in Artificial Intelligence (MScAI JAN24)

Rohit PrasannaKumar
Student ID: 23133872

School of Computing
National College of Ireland

Supervisor: Paul Stynes

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: Rohit Prasanna Kumar

Student ID: 23133872

Programme: Masters in Artificial Intelligence (MScAIJAN24) **Year:** 2024

Module: Practicum

Lecturer: Paul Stynes

Submission Due Date: 15/12/2024

Project Title: A Deep Learning Filtration Framework to Eliminate Not Safe For Work content in Digital media.

Word Count: 6450 **Page Count:** 22

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Rohit PrasannaKumar

Date: 15/12/2024

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

AI Acknowledgement Supplement

1 PRACTICUM

A Deep Learning Filtration Framework to Eliminate Not Safe For Work content in Digital media

Your Name/Student Number	Course	Date
23133872	Msc AI	15th Dec 2024

This section is a supplement to the main assignment, to be used if AI was used in any capacity in the creation of your assignment; if you have queries about how to do this, please contact your lecturer. For an example of how to fill these sections out, please click here.

2 AI Acknowledgment

This section acknowledges the AI tools that were utilized in the process of completing this assignment.

Tool Name	Brief Description	Link to tool
NA	NA	NA

3 Description of AI Usage

This section provides a more detailed description of how the AI tools were used in the assignment. It includes information about the prompts given to the AI tool, the responses received, and how these responses were utilized or modified in the assignment. **One table should be used for each tool used**.

Tool Name	Brief Description	Link to tool
NA	NA	NA

4 Evidence of AI Usage

This section includes evidence of significant prompts and responses used or generated through the AI tool. It should provide a clear understanding of the extent to which the AI tool was used in the assignment. Evidence may be attached via screenshots or text.

5 Additional Evidence:

6 Additional Evidence:

A Deep Learning Filtration Framework to Eliminate Not Safe For Work content in Digital media

Rohit Prasanna Kumar

x23133872

M.Sc. Artificial Intelligence

National College of Ireland

Abstract

Not Suitable for Work (NSFW) is a term which is used on the internet to warn users about content with inappropriate material such as nudity which might not be suitable to access at any work or study environment.

Most websites and chat applications have functionalities of NSFW content filtration which can disable the uploading of images or texts containing explicit words. However, this technique is not error free as there are cases where the filter blocks even the safe content. Besides that, there is no process currently in action which can filter for video content. The challenge here would be implementation of NSFW filter in a video content to identify NSFW content hidden within a video, which normally cannot be deduced by filename or the video thumbnail. This research proposes a Deep Learning Filtration Framework to filter the NSFW content in a Video. The proposed framework combines a Detection Model and Filtration technique to extract the content out of video. The Deep Learning Model is implemented using YOLO (You Only Look Once) trained on NudeNet classifier dataset for training and LSPD Dataset consisting of 500,000 images and 4,000 videos, with 93,810 labelled instances in 50,212 images for testing. The model is evaluated based on accuracy of the prediction. Results demonstrate values for safe and unsafe image where in values more than 75 percent in confidence are considered NSFW.

This research is on interest to provide a safe environment in the social media and chat sites to help avoid accidental exposure to such content and help the social websites/applications sustain their main purpose of social interaction based on demographics and not allow unmoderated content to be distributed at ease.

Keywords: NSFW, YOLO, CNN

1.Introduction

The internet is an amalgamation of content generated from various sources and mediums. Providing access to many users within the internet to access the content at ease. The only problem with such availability is that there is no way for the moderation of content available. In a period, where any form of entertainment media content like movies and games have ratings board to help ensure proper distribution of content. Movie ratings board for movies and ESRB for video games where the objective of the body is to help protect exposure of mature content from children. Although freedom of browsing on the internet is a right, there are instances where certain demographics have to be protected from content which would be harmful to young audience's psyche without proper guidance. As there are occurrences of children's mental health damaged after exposure to such content. These include Accidental discovery, Peer Pressuring, Online Challenges, Suicide, and self-harm content, eating disorder content, Pornography content and Self-generated intimate images. There is also a particular barrier amongst minors against speaking as not all children have the same confidence or knowledge about handling such cases. And even if few managed to question the same, they would be in fear of whether they would be scrutinized for the same and reporting the harmful content to the platform support would be viable.

This research would be on how to tackle such scenarios early on so that the minor audiences would not be having direct exposure to such contents and will only be able to process the child friendly version of the same. This would be achieved by utilizing the information available on previously done works which targeted the identification of NSFW contents and combining them with other works in computer vision to produce a result which would be deemed satisfactory. These works include (Warren, Yuta and Noa,2024) which studied the biasing observed in NSFW detection models when identifying the image. As the model was identified having biases in detection with higher ratio of normal female faces and poses identified as false positive this accounted to as high as 17.9 %. There was also another dilemma in the model where lighter skin tone individuals were mis-predicted at higher rates than those individuals falling under darker skin tone, pointing to imbalanced number of samples.

To address these challenges, this research aims to develop a comprehensive framework for the accurate filtering of NSFW content within video media. By leveraging advancements in deep learning, the framework integrates state-of-the-art detection methods to enable real-time identification and filtration of explicit content, safeguarding users and fostering a healthier online environment.

To address the research question, the following specific objectives were derived:

1. Investigate the state-of-the-art techniques and limitations in detection of NSFW content.
2. Design a novel framework to overcome identified shortcomings and enhance detection capabilities.

3. Implement the proposed framework for accurate detection and filtration of NSFW content in digital media.
4. Evaluate the framework's performance using metrics such as Mean Average Precision, precision, and recall.

Most of the previous work (Francesco, C. et al, 2022), (Warren, Yuta and Noa,2024) were based on identifying and detecting NSFW content in images with annotations, combining LLM's and CLIP distances etc. But nothing was put forwarded in the way of how to reduce the exposure. This research would be tackling the said problem by utilizing the knowledge from previous works and target video content filtration.

The major contribution set forth by this research would be a novel take on implementation of deep learning framework which would try to implement a real time elimination technique, which would process the video and filter out the NSFW parts. To eliminate the existence of such content and avoid exposure to the same. NudeNet is the current best OSS option for nudity detection (Bedapudi, P. (2019)). The dataset utilized in NudeNet would be the dataset the author has publicly made available through archive.org. These annotations will help in proper identification of NSFW content which needs to be filtered in the output.

2. Related Work

There have been significant studies related to NSFW content identification. Some are based entirely on text detection and identification, few are based on image identification, rest are based on image to text and identifying the same. Over in this section how it was tackled by different authors will be outlined.

Ample research was previously conducted on the study of NSFW classifiers and NSFW Content identification (Lienhart,R, & Hauke,R, 2009), (Leu,W. Nakashima,Y. & Garcia, N.,2024), (Zhelonkin, D. & Karpov, N. 2020) and (Perez, et al 2017) to list a few. The authors (Zhelonkin, D. & Karpov, N. 2020) have specified how works on NSFW detection were started as early as 1996.The research closer to a decade ago includes the usage of probabilistic Latent Semantic Analysis (pLSA) where the authors (Lienhart,R, & Hauke,R., 2009) discuss on the two prevalent techniques used then which includes text identification based on the image using website information or the latter where skin color or texture based judgement would have been processed alongside previously used models which included Neural Networks ,Support Vector Machines or non-statistical classifiers like manually tweaking the heuristics. The authors (Lienhart,R, & Hauke,R., 2009) here had implemented the same using k-Nearest Neighbor (kNN) with SIFT and Self-Similarity Features. And the pLSA approach was found to be compress high document vectors enabling faster k-NN search which helped produce better results, The limitations here seen were the limited dataset numbers. Another considerable research addition would be research done on geometric analysis of skin areas with the first stage being used to collect images with pixels having a color closer to the body.

The authors (Zhelonkin, D. & Karpov, N. 2020) have personally performed a research implementing NSFW identification using Xception, MobileNet and ResNet-50 based on the results from the ILSVRC and similar held competitions helping gain an edge knowing the performance of these components as in being fast, compact and the availability of pre-trained weights on Image-net and the associated information that ImageNet can be fine-tuned for better results and other tasks helps elevate the field of work.

The authors in (Wasserman, N. et al, 2024) have addressed the issue of adding objects to images without user-provided input masks proposing the idea of removing objects. This was followed by the idea used by the method proposed in InstructPix2Pix (IP2P) (Wasserman, N. et al, 2024) which contains source and target image with editing instruction. Although there was a dilemma in mask-based inpainting that object removal is not without having repercussions as in having parts of the artifact still being present in the image even after mask or similar other distortions to name a few. These problems had been cleared by employing Large Vision-Language Model alongside LLM and trained in diffusion models.

It was also identified that in many cases with more skin exposure like in wrestling or sunbathing there were false alarms in those cases by the previously available models (Perez, et al 2017). Their research is also the first to work on NSFW content filtration which includes frames in motion as per the authors knowledge. The proposal also considers two ways for implementation static(picture) and dynamic (motion). The dynamic portion also involves the temporal flow. The final decision in this case would be made by an SVM classifier. Datasets used by authors (Lienhart,R, & Hauke,R., 2009) were limited to bikini based models “7,676 (containing 600 bikini images) which were used for training and 13,023 (containing 512 bikini images) which were used for testing where in dataset 2 consisted of 2,668 adult content images of which 600 were used for training purposes and 2,068 for testing purposes. NSFW-CNN was trained using an open-source *NSFW-Data scraper* available online. NSFW-CNN, CLIP-classifier and CLIP-distance were tested on The *Google Conceptual Captions (GCC)* dataset, MSCOCO The Microsoft Common Objects in Context with PASS (The Pictures without humAns for Self-Supervision) as a benchmark using a random subset of 11, 685 images. Inputs were resized to 299x299 pixels for NSFW-CNN and 224x224 pixels for both CLIP classifiers.

Ample techniques were previously used based on the multiple researches in the field these includes and are not limited to image classification using BoVW (Bag-of-Visual-Words), SIFT (Scale invariant feature transform), SURF (Speed up Robust Features) and the most well-known would be the usage of AlexNet in CNN Classification problem ILSVRC 2012 (ImageNet Large Scale Visual Recognition Challenge) as mentioned in the work of authors (Zhelonkin, D. & Karpov, N. 2020).

NSFW-CNN: Based on an InceptionV3 CNN Model which categorizes the content into five classes namely neutral, drawings, sexy, hentai and porn. The model provides a value from 0-1 where it was identified as porn image if the value was more than 0.7. NSFW-CNN extracts embedding from images with a CNN. The data training was based on NSFW Data Scraper (Warren, Yuta and

Noa,2024) with no information of training samples specified. The model was previously used for Abusive Content Detection and RedCaps dataset for filtering data contents.

CLIP-classifier: CLIP (Contrastive Language-Image Pretraining) followed by a three layer Fully Connected Classifier. It is a frozen ViT L/14 network where FC Classifier is trained on the subset of LAION-5B Dataset. This classifier also follows the 0-1 convention where 0.7 confidence is identified as porn image.

CLIP-distance: CLIP (Contrastive Language-Image Pretraining) followed by distance computation. It is also a frozen ViT L/14 network which converts input image to input embedding. The distance between image embedding with a set of 17 precalculated text embedding with NSFW is calculated. The information on 17 concepts has not been revealed. LAION-400M dataset used the CLIP-Distance for filtering content.

In conclusion, the above studies provide significant information on the advancements in NSFW content identification which includes feature extraction techniques (like SIFT, SURF and BoVW) and state of the art deep learning models such as NSFW-CNN, CLIP-classifier, and CLIP Distance. The above techniques for NSFW identification are also suffering from notable limitations, most importantly on the proposal proposed by this research which involves real-time detection capabilities. As state-of-the-art methods like NSFW-CNN and clip-based classifiers focus more on image classification rather than efficient object detection. With YOLO it would be possible to target the NSFW Features and filter the digital media in real time ensuring an NSFW safe output. It would be ensured to utilize the latest yolo model available integrating ultralytics framework. Additionally, the model will be trained to produce a model which can even run on edge devices with lower computational power like a router, mobile phones etc.

3. Methodology

The research methodology consists of five steps in order which includes data gathering, data pre-processing, data-modelling, evaluation, and results. The research process can be identified as in Fig.1.

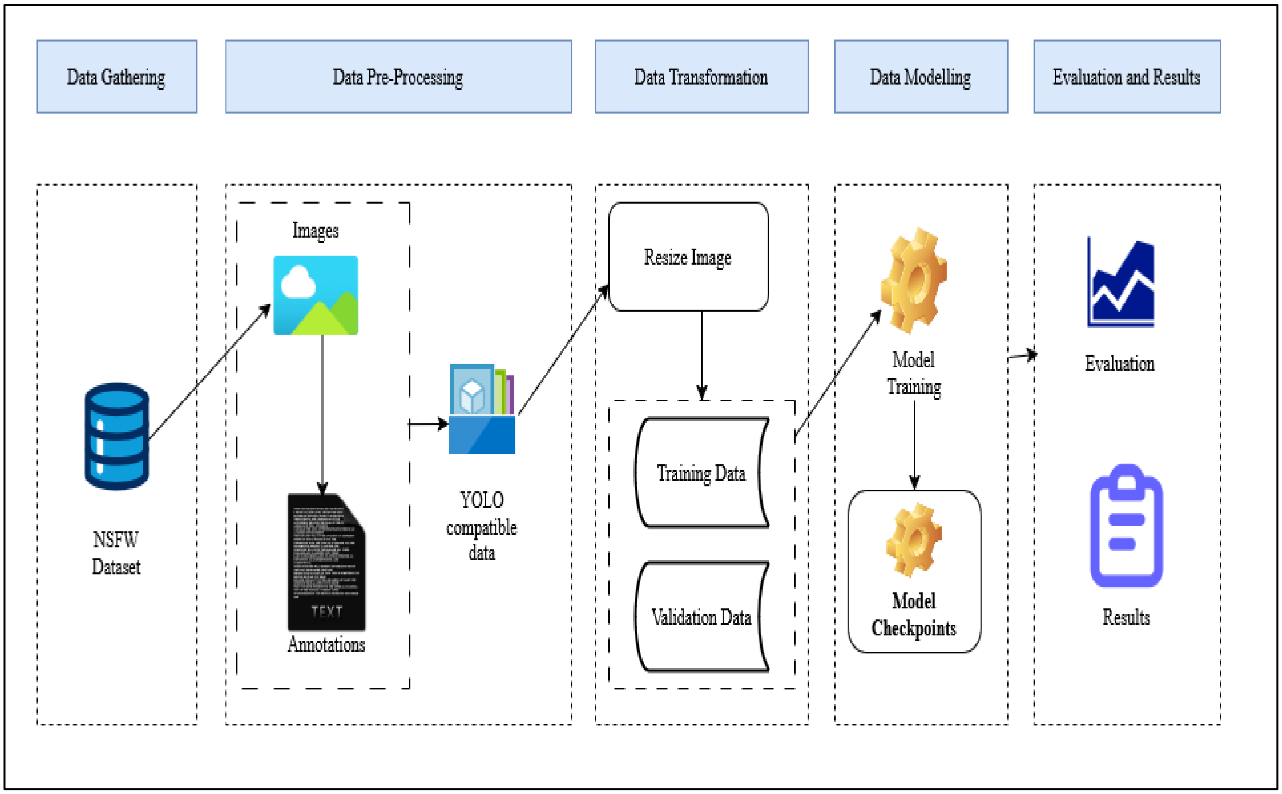

Fig. 1: Research Methodology

The first step, **Data Gathering** would be collection of datasets required to train the deep learning model (YOLO) for this process the data has been gathered from two datasets provided by (Bedapudi.P, (2019)) and (Duy, et al (2022)). The dataset provided by (Bedapudi.P, (2019)) namely Nudenet_classifier_dataset_v1 consists of data already in yolo readable format which contains images of size 320x. Another archive provided by the author has 19,999 images with better resolution, alongside annotations information and class information on csv file. This would be the dataset which would be used for training. As it provides a small step towards the training process. By identifying the pattern and losses in the training set for this approximately 20,000 files. It would be easier to process the training data in LSPD Dataset (Duy, et al (2022)). The amount of files with nsfw content required for training purposes in LSPD Dataset were close to 2,00,000 files. So, it was decided to just utilize the LSPD data for testing purposes in the current iteration and in the next iteration of this paper it would be utilized to provide custom class names and use as training data after evaluating the performance of this model in limited computational capacity and limited training data.

The second step, **Data Pre-Processing** would be to ensure image integrity and quality. Since the YOLO model requires proper annotations and a yaml file with its associated information it is essential to ensure proper integrity checks. So, it was ensured that the annotations and images were linked and there was no file corruption or file name corruption. The process followed here was three step process which includes sanitization of content, label connector and label tester. (Bedapudi.P, (2019)) dataset was provided with a csv file which contained data about bounding boxes which were placed on the image files and the name of the image file alongside images. A python script was utilized to fetch the information from csv to generate a data format structure consisting of images, labels and a yaml file consisting of class information. The python script checks for name corruption, label corruption and also generates a proper yaml folder structure. It was also during this process it was ensured to remove corrupted annotations and images associated with it as both are proportional and can cause inconsistencies in generated model if left unchecked. This was identified as sanitization phase.

Now to ensure that the annotations and images are linked properly a python script was utilized called label connector to help establish the correct linking of annotations and image. And in case of errors, try to fix the errors which might have been generated from sanitization phase.

Finally, label tester phase ensures a visualization of the all-annotated images so it can be further verified that if there are any missing annotations field or lost data on annotations/images, as there are instances of corruption in images which need not be rectified if only it is a miniscule amount. This process ensures a quick visualization of all the annotated images which will be used for training purposes and helps check the placement of annotations and rectify if there are instances of high volume of corrupted, mislabeled, or unlabeled data.

The third step would be **Data Transformation** which would include resizing images and splitting datasets into training, testing and validation. It was realized that during the sanitization phase a portion of image data and its annotation was lost due to corruption in data. Resulting in loss of over training data:12,235, testing data: 1,530 and validation data: 1,529. The splitting of data into testing and validation helps to avoid overfitting. **Data Modeling** involves transformation as well where during modeling a parameter is provided to help resize the image with annotations. Currently the model is being run at 640 image resolution. The training data was split into training, testing and validation. With approximately 10 % of data falling under validation and 10% in testing. And keeping 80% for training. It was decided to utilize the class information provided by (Bedapudi.P, (2019)) as a steppingstone for training the model. It was decided to YOLO model as it was deemed to be the best model for real time object detection. And out of the options available it was further decided to use the yolov11n model. It would be a nano model which would be suitable to implement for edge devices and devices having comparatively low computational power.

4. Design Specification

The process diagram displayed in the Fig. 2. explains the base architecture of the YOLO model.

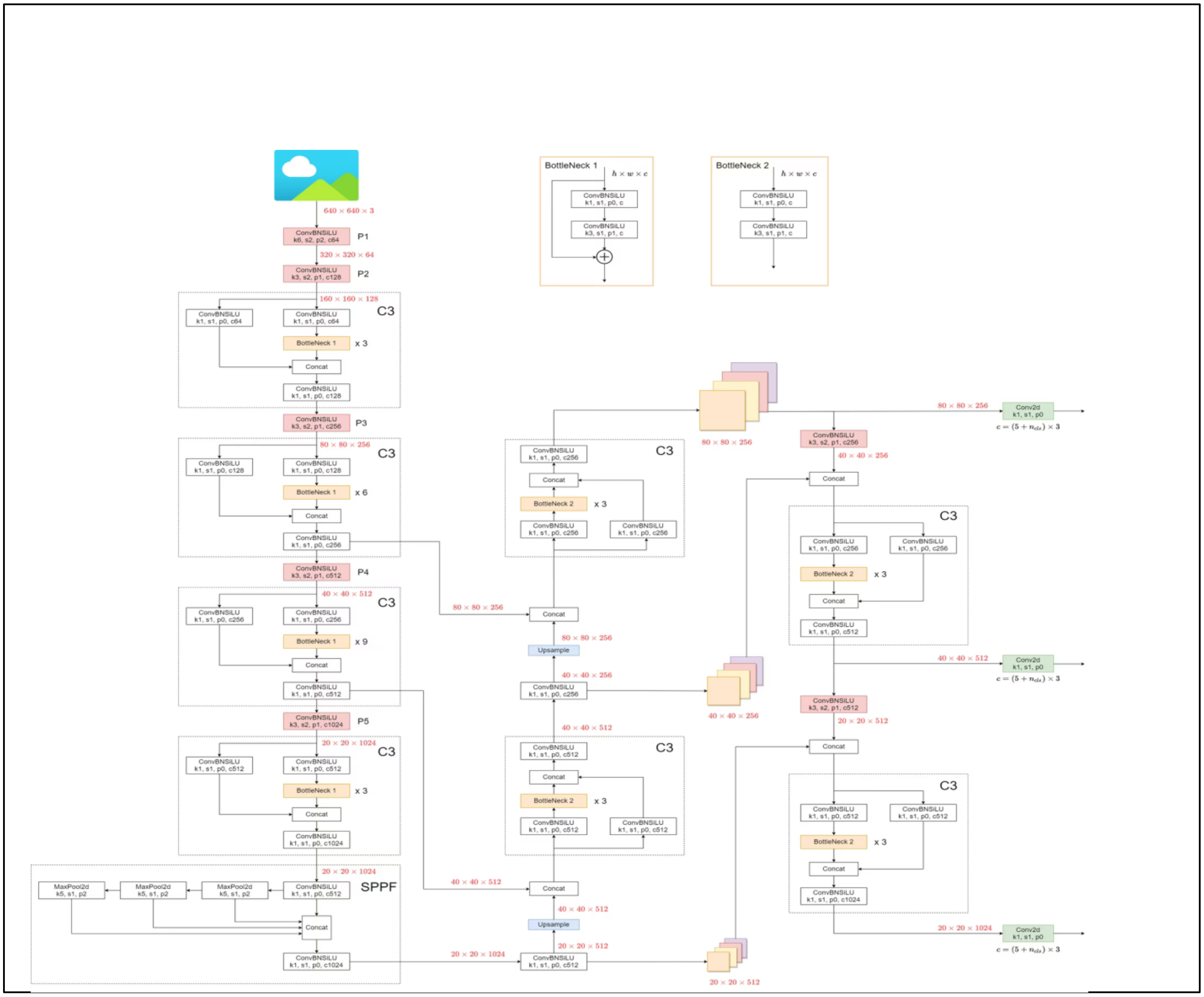

Fig.2: YOLO v5 Architecture (Jocher, G. and Qiu, J. (2024))

The YOLO architecture would be based on real-time object detection architecture which would be designed for speed and accuracy. The major components of the YOLO Architecture would include Input image, Backbone, Neck and Head. Starting with Input Image there are variants to the Image data these would be discussed in data variants. The Backbone part of the architecture is the component responsible for feature extraction it includes CSPNet(Cross Stage Partial Network) modules which consists of Convolutional Layers (Conv+BN+SILU) where each convolutional block consists of 2D convolutional layer, BN as in Batch Normalization to stabilize training process. SiLU would be the activation function. C3 modules help reduce computation while preserving performance. The bottlenecks on top help either to skip connection between input and output or not to skip the connection. As with computing this feature is necessary not to increase computational costs as in work with efficiency. SPPF (Spatial Pyramid Pooling – Fast) performs multi-scale feature extraction using MaxPool operations, this helps in object detection at varied sizes necessary for the scope of this experiment. Like the human body the neck part in this context also combines features from backbones and merges them to help detect objects at multiple scales. Of which the key components would include upsampling where features are upscaled to larger sizes during training for detection of smaller objects. The outputs from P3, P4 and P5 are concatenated to combine information from multiple scales. C3 modules are present in this section for refining fused features before sending them to the head. Head is the part where the prediction of final output takes place. These include bounding boxes containing the co-ordinates of detected objects. Class probabilities contain the scores for object classes. Objectness score would be the confidence an object exists in the predicting bounding box. And during the training process the input image goes through the backbone, neck and head in the forward pass. And loss calculation is implemented by multi-part loss function. And SGD optimizer helps minimize the loss in Backward pass. To ensure hassle free training it would be ensured to have checkpoints after every epoch so in case of any failures, the training process can be continued from the last checkpoint. Where the best perfoming model weights would be labelled as best.pt and the last checkpoint run would be saved as last.pt helping ensure continuity. The best performance measure is based on the validation metric of Mean Average Precision.

4.1 Data Variants

Since the data is digital media, in this case it is restricted to image and video.

Images: For images, the model with provide boxed annotations for the regions it identifies under the provided class name for NSFW Features. The dataset has been annotated with 6 class names. These class names were provided alongside the dataset (Bedapudi.P, (2019)).

For videos, it is further divided into two components,

4.1.1 Utilizing FFMPEG

Using FFMPEG integration, the video is extracted into frames of Portable Network Format Graphics (PNG) images to avoid losses in image quality. And finally, after all the frames have been evaluated the images in no annotations folder would be combined back into the video format. Thus, eliminating the detected NSFW content from the original video, and providing a new video with potentially few NSFW features.

4.1.2 Utilizing OpenCV

Using FFMPEG, separate folders were created for storage of frames/images to help tackle scenarios where no in- depth analysis is required, also considering the situation of no external storage space availability. OpenCV provides an alternative for real time tagging of annotations in the video. Although, in this case, it was decided to utilize the buffer capability of OpenCV to directly produce a filtered video with as few NSFW features as possible.

4.2 Deep Learning Feature Identification Model

The feature identification is an important part of this model. As with feature identification, it helps identify whether the media content has NSFW features or not. The feature identification is based on the classes provided as a part of the annotation. Where each feature would be annotated, and a class would be provided based on the type of feature. The model was built upon the YOLO v11 model provided by ultralytics framework. By utilization of transfer learning process a pretrained 'yolov11n.pt' model was used for its lowest computational cost sacrificing a bit of mAP accuracy. This model is model is identified as YOLO11n (NANO) the smallest in the series with the limitations of fewest parameters but is optimized for speed and minimal computer requirements. In this experiment, the auto parameters were used for model training as initial test s provided satisfactory results with the dataset utilized for training.

5. Implementation

After training the model and selecting the best checkpoint. In this case the 200 epoch model checkpoint(best.pt), is loaded into the jupyter notebook environment where the options to inputs are provided where in if it is an image file with extensions ranging from ('png','webp','jpg','jpeg') it will process the file in the trained model where in if we look at the implementation process we are considering the scenario where the input contains nsfw features. So, when such an input is provided the image gets annotated with nsfw features with a confidence. If it is a video file, based on the requirement if it's a system where there is availability of external storage and the data needs to be analysed or secured without damaging the source and ensure better safety in produced output. ffmpeg technique is utilized where ffmpeg commands are utilized to extract frames from a video, the frames are then stored as png (Portable Network Graphics) format (lossless quality) to ensure that there is no quality loss in the video. Then the model evaluates the extracted png frames to further divided into two separate folders namely Annotated and non-Annotated folder. Where in the annotated images as in the images with nsfw features identified would be moved to the annotated folder, there are also instances where a library update causes the annotated images to be saved in the root folder. The important part is the non-annotated folder which contains frames devoid of nsfw content. These frames are combined again to create a video file with same configuration which does not contain the NSFW features. The other instance is when it is run via open cv route where the frames are processed in the volatile memory (RAM) it can be used in a system where there is no storage availability and must process it real time. The frames are not stored as temp files but once the frame is in memory it gets evaluated frame by frame and the output file is generated and the frame leaves memory as soon as its processed helping save storage space. After evaluating all files the output file is generated in which the filtered video is obtained. The Fig.3 visualizes the implementation process in action.

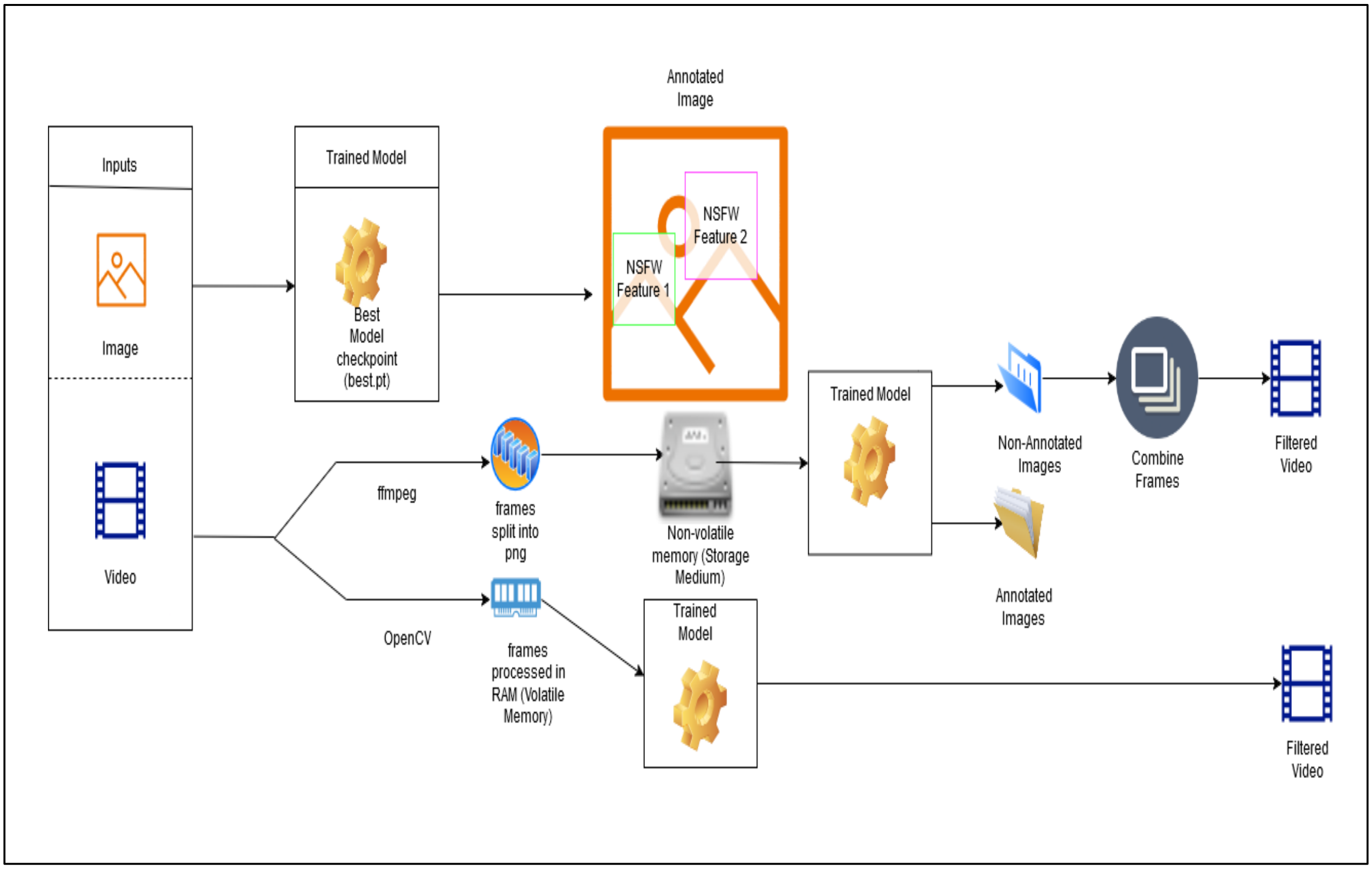

Fig 3. Implementation process.

6. Results and Discussion

The aim of this experiment is to evaluate the performance of YOLOv11 on NudeNet classifier dataset (Bedapudi.P, (2019)) and choose the best model from the trained models to help extract NSFW features within a digital media. Using transfer learning technique, yolov11n weights were used to adapt to the custom dataset and classes provided. These classes were directly utilized from the NudeNet classifier dataset (Bedapudi.P, (2019)). These include 6 classes namely exposed belly, exposed buttocks, exposed breast female, exposed genitalia, exposed breast male, exposed genitalia male. For results and evaluation comet.ml API was integrated to provide real time statistics of training to provide the minute and real-time insights on the experiment during the training process. Fig. 3.a and 3.b show the statistics based on time required to process 75 epochs and 200 epochs respectively alongside a custom name for each session. Providing the information that more epochs mean more hours of training but to check whether it has overall it is necessary to evaluate other metrics as well. The loss versus step graph was evaluated as it helps measure the model's prediction.

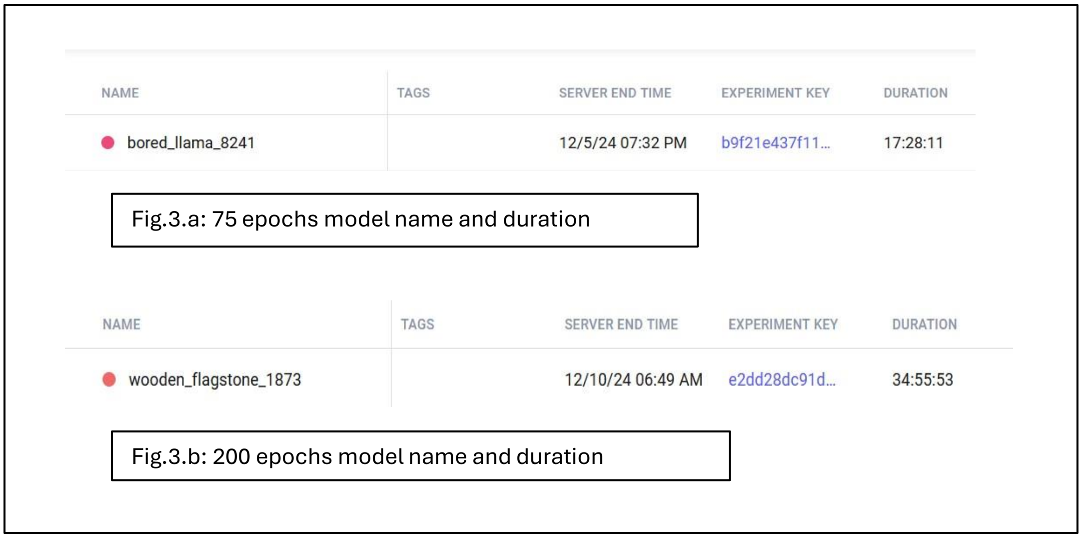

Fig.3.a: 75 epochs model name and duration

Fig.3.b: 200 epochs model name and duration

Fig 3: Time required to process different sets of epochs alongside their session names.

It was identified that loss starts high for both 75 and 200 epoch models, approximately 150 and has seen to be decreasing steadily during the course of training. It is visible that at the end of 75 epochs, the loss came close to 50. Similarly for 200 epoch model, it is visible that the initial loss is similar, and there are further fluctuations along the course of training, but it averages up to 50. It provides an indication that the model has converged around 50, but when looked closely the model still shows potential as there is a good drop close to 150,000th step. Fig 4.a and 4.b Denotes the loss vs step graph for 75 epoch model and 200 epoch model.

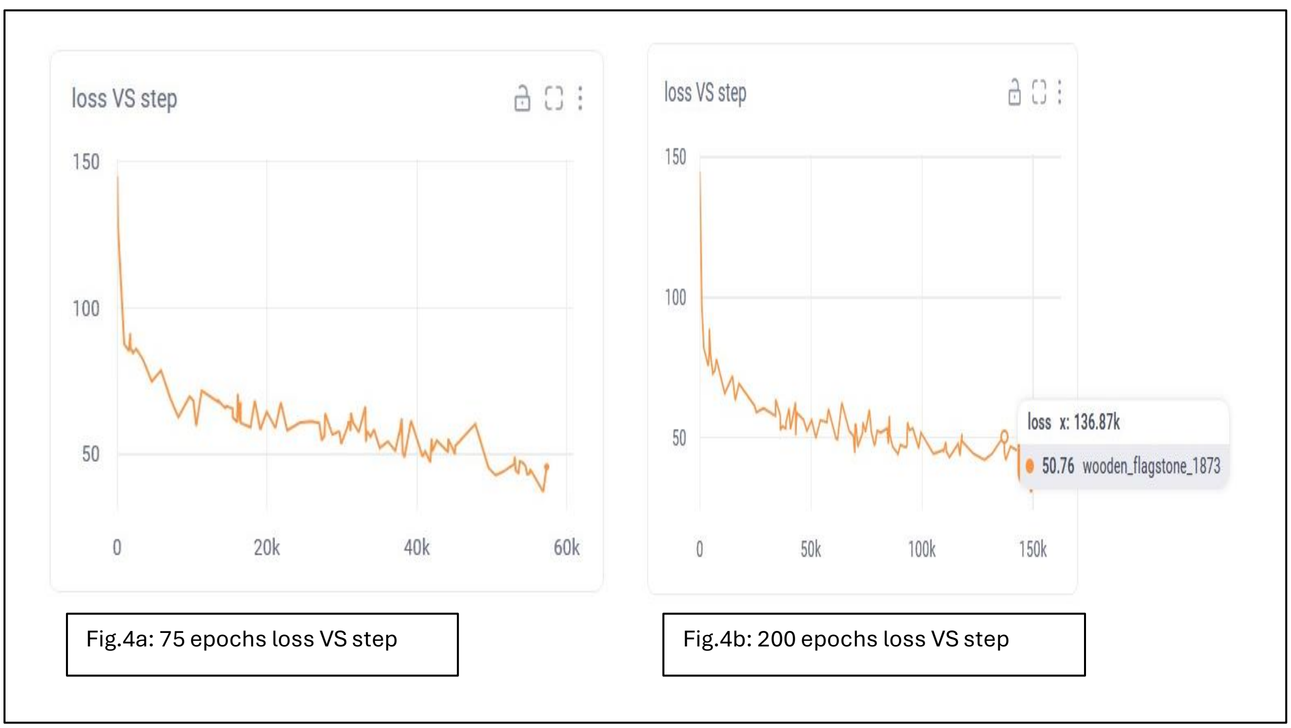

Fig.4a: 75 epochs loss VS step

Fig.4b: 200 epochs loss VS step

Fig 4.a and Fig 4.b: Loss VS step graph for 75 and 200 epochs

Now considering the learning rate for both models. As it is a critical hyperparameter as it influences the learning speed of the model. The learning rate for YOLO includes three parameter groups pg0, pg1 and pg2 to help adjust the learning rate of the model to help have control over the training process. pg0 is learning rate for biases in Convolutional layers in feature extraction section of YOLO. pg1 is for convolutional layer weights and finally pg2 is for detection layers for predicting bounding boxes, class probabilities etc. The learning set was 0.01 and it is visible that it is gradually decaying linearly close to 0, as it ensuring that parameters updates are becoming smaller as the model is approaching convergence, to help avoid overshooting. For visualization we can have a look at the 200-epoch model to understand the decay. Refer Fig.5 for visualization of the learning rate peak and decay for 200 epoch model. Now the key evaluation metrics for the YOLO training would be discussed these include mAP50 (Mean Average Precision at IoU (Intersection over Union)), mAP50-95, Precision and Recall. IoU in mAP50 measures the overlap between the predicted and actual bounding box for the image. With the threshold being kept at 0.50 if above or equal to this value it is considered true positive. The average precision is calculated across all object categories and a mean value is obtained of the AP values. With higher values considered better performance in detection of objects. MAP50-95 is much more exhaustive as it evaluates the model's performance at multiple thresholds from 0.50 to 0.95 in increments of 0.05. Then there are precision and recall metrics where precision is based on the ratio of true positive detections to the total number of detections with higher precision indicating few false positive values, meaning less occurrences of mistaking other objects as target objects. Recall metric would be the ratio of true positive detections to the total number of actual objects with high recall ensuring model properly able to detect most objects in an image, but at the possibility of false positive detections as well. The observations from the graphs generated for both models in Fig 6.a and Fig 6.b it is possible to get a conclusion that mAP50 stabilized to 0.8 in 20,000 steps in 75 epochs run and seems to have reached a plateau as well in 200 epochs run, with little improvements visible in the graph. MAP50-95 also seems to increase more gradually and have seemed to

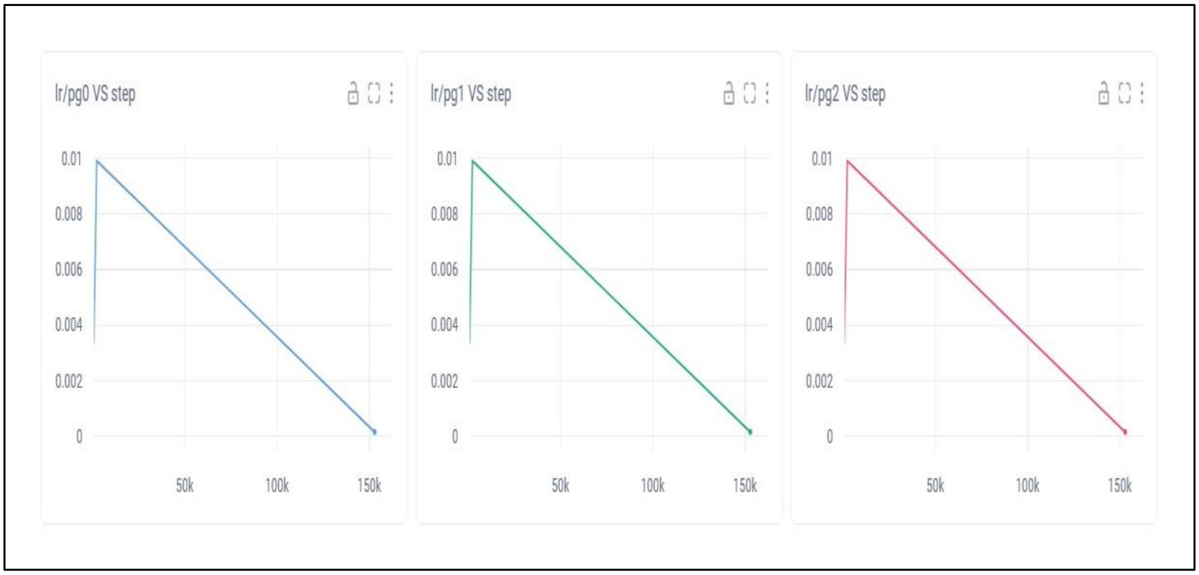

Fig .5: Learning rate decay in 200 epochs model.

stabilize near 0.5 by the end of the training, indicating satisfactory performance but in 200 epochs run it is also stagnated to the same range as no gain seems to be obtained from additional training epoch. Precision values seem to be maintaining a closer value of 0.75 and precision increasing slightly after 20,000 steps ensuring consistency in prediction with few false positives. A recall value of 0.75 provides a reasonable success ratio for identifying most objects. Not much has changed for both values over the time for 200 epochs except for a slight consistency improvement due to longer training durations. The training loss components are also to be identified which includes box loss, classification loss and Distribution Focal Loss. Box loss measures the error in prediction of bounding box compared to the original bounding box placed on training image. Class loss is dealing with the error in prediction of classes and DFL loss is about refining bounding box co-ordinates. To see how it reflects on the 75-epoch run it is visible that box loss starts around 1.8 and is decreasing at a steady pace to 1.0. Classification loss also seems to start high at 3.0 but stabilized around 1.0. DFL loss also seems to follow the same pattern where it starts at 1.8 and decreases closely to 1.2. The differences in 75-epoch and 200-epoch model does not seem to be remarkably high as in 200-epoch model it only decreased approximately to 0.8 by 150,000 steps. Same goes for classification loss which stabilized around 0.8 - 1.0 suggesting marginal improvement. DFL loss also decreased from approximately 1.2 to close to 1.0 by 150,000 steps indicating better refinement in bounding box accuracy with the additional training but not as extensive. Fig 7.a and Fig 7.b provide a graphical view of the discussed components. Identifying validation loss is as important as identifying the training loss component as these values are evaluated on validation set, helping with the model efficiency.

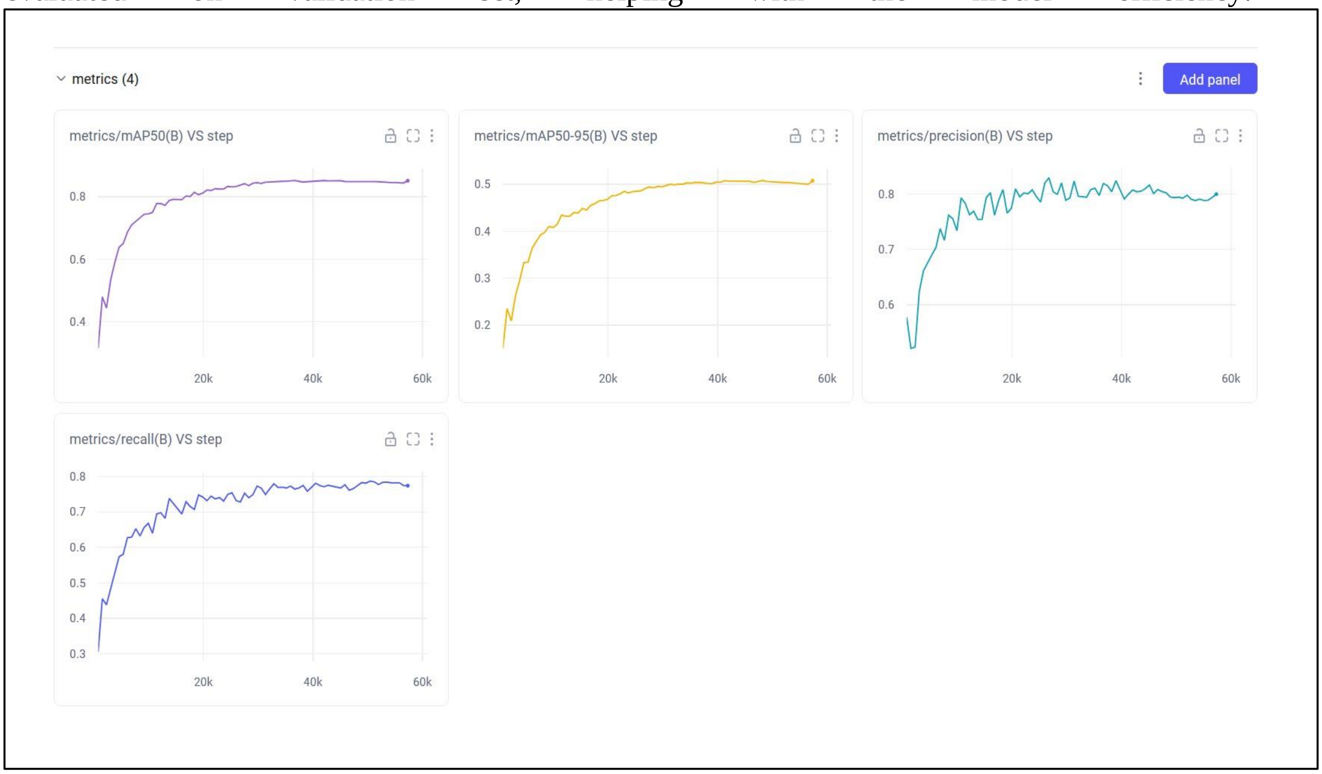

Fig 6.a: mAP50, mAP50-95, precision and recall metrics for 75 epochs run

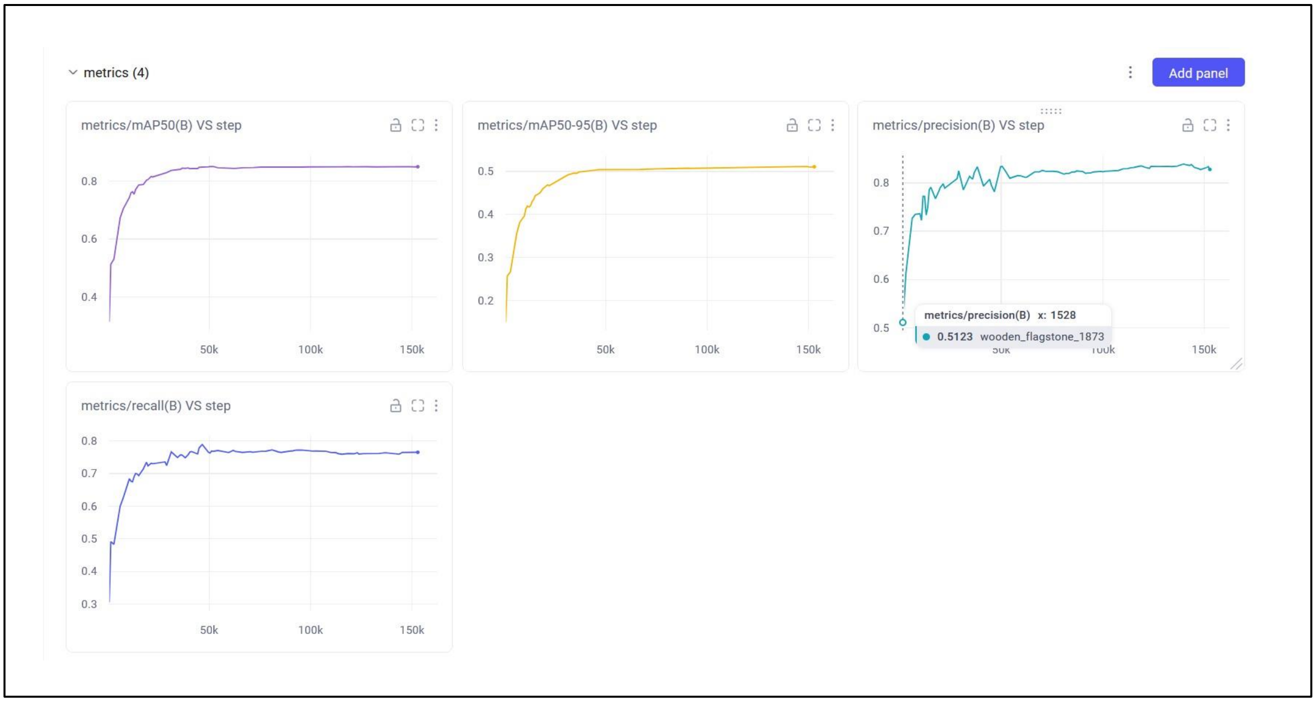

Fig 6.b: mAP50, mAP50-95, precision and recall metrics for 75 epochs run

As validation would help us identify the model's typical performance. The validation loss components would include the same components as training loss. But this time they help us evaluate the results. The box loss seems to start around 1.8 and decreases to approximately 1.4 for 75 epochs, which translates to model having acquired basic generalization on bounding box predictions, while the box loss further reduces slightly to 1.3 by 150,000 steps. It shows that further training has produced minor improvements in results. Further classification loss has been noted to stabilize early in the training procedure, as it drops effectively from approximately 2.5 to 1.0 within the early 20,000 steps and no improvement was observed between this and 200 epoch helping reach the conclusion that maximum potential for classification was obtained within 75 epochs for the said data. DFL loss also decreased from approximately 1.8 to 1.4 showing consistent improvement for bounding box in validation set and it reduced to approximately 1.3.

Fig.7.a: Box loss, Classification loss and DFL loss for 75-epoch model

Fig. 7.b: Box loss, Classification loss and DFL loss for 200 epoch model

Fig 7.a and Fig 7.b: Training Box loss, Classification loss and DFL loss for 75 epochs and 200 epochs.

Comparing validation loss and training loss helps ensure that the model is not overfitting i.e., model is performing well on training data but poorly on validation data. As the similar trends in metrics over training and validation loss suggest good generalization. And as with the pattern exhibited earlier, the stabilization of all validation loss at 75 epochs indicates convergence of the model around 75 epochs additionally computing 125 more epochs for minor increase in improvements seems an overutilization of resources for minimal gains. Fig. 8.a and 8.b will provide the visual representation of validation loss for 75 epochs and 200 epochs model.

Fig.8.a: Validation loss components box loss, classification loss and dfl loss for 75 epochs

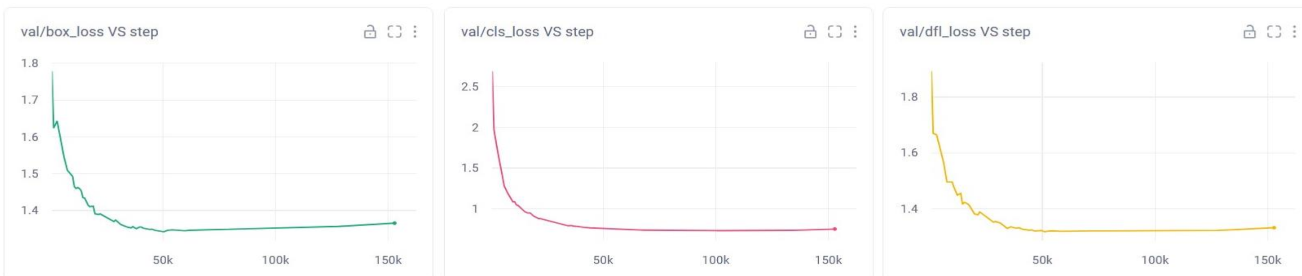

Fig.8.b: Validation loss components box loss, classification loss and dfl loss for 200 epochs

Fig.8.and 8.b: Validation loss components box loss, classification loss and DFL loss for 75 and 200 epochs.

The final four important metrics considered for the evaluation of YOLO models performance would be F1-confidence curve, Precision-Confidence curve, Precision-Recall curve, and Recall confidence curve. F1 score would be the harmonic mean of precision and recall, which would balance the two metrics. Peaks in the F1 score indicate the optimum confidence threshold for achieving balance between precision and recall. It can be evaluated that the F1 score peaks at 0.79 with a confidence of 0.352 and in 200 epochs it goes at 0.80 with a confidence of 0.397. Precision-Confidence curve indicates the proportion of correctly predicted objects among all the predictions made. The curve exhibits the pattern of precision changes as confidence threshold varies. In 75-epochs variant, the precision reaches maximum value 1 at a confidence of approximately 0.904. In 200-epochs variant, precision reaches maximum value 1 at a confidence of approximately 0.893. Precision-Recall curve displays the trade-off between precision and recall for varying confidence thresholds. mAP50 for all classes is found to be on average 0.851, while for 200 epochs it shows minimal degradation at 0.849. The class analysis over the 6 classes helps identify the classes the model is performing well in. Recall-Confidence curve measures the proportion of true positives detected among all the true objects. This curve displays the change in recall based on varying confidence threshold. For 75 epoch and 200 epoch models it peaked at 0.96 and 0.95, exhibiting consistency in training. This ensures proper spotting of features or objects but may hurt the precision. Fig. 9 and Fig 10 will showcase the complete set of graphs associated with the discussed metrices. Alongside that Fig. 11 will provide the confusion matrix for 75 epoch and 200 epoch.

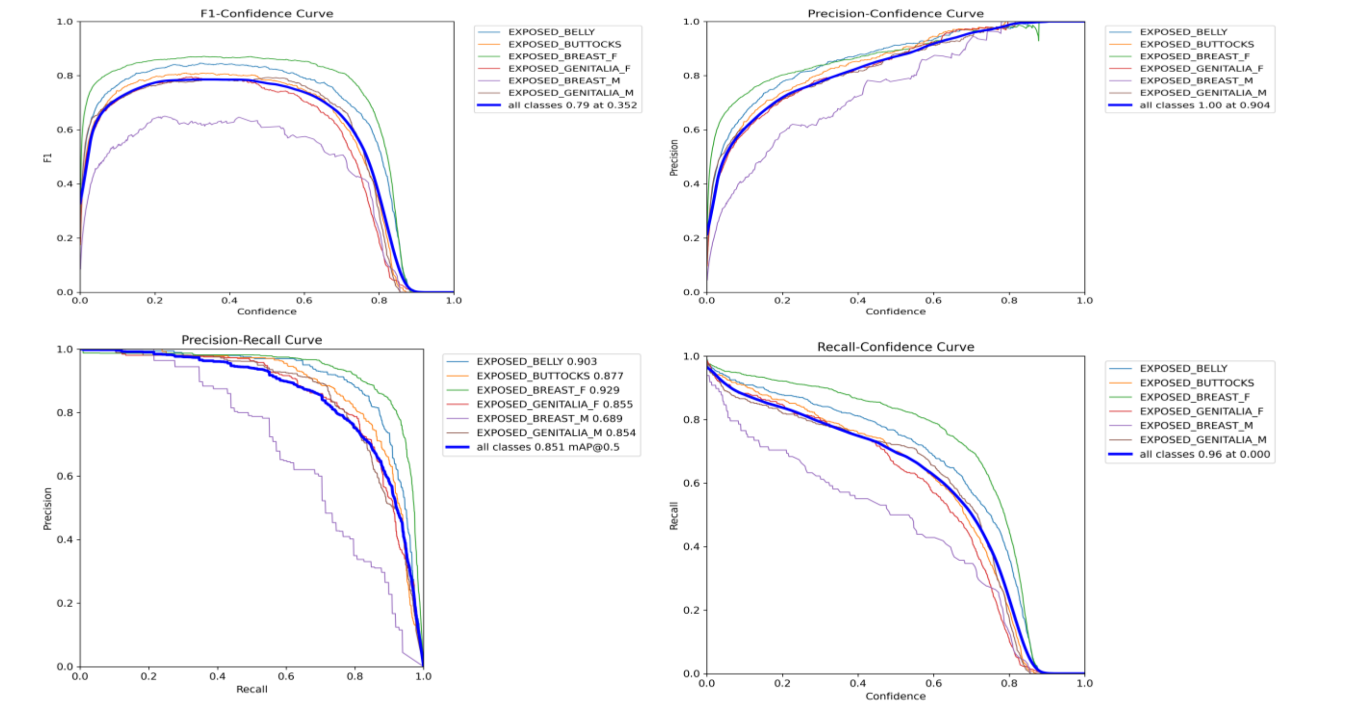

Fig 9. F1-Confidence Curve, Precision Curve, Precision-Recall Curve and Recall-Confidence Curve for 75 epoch

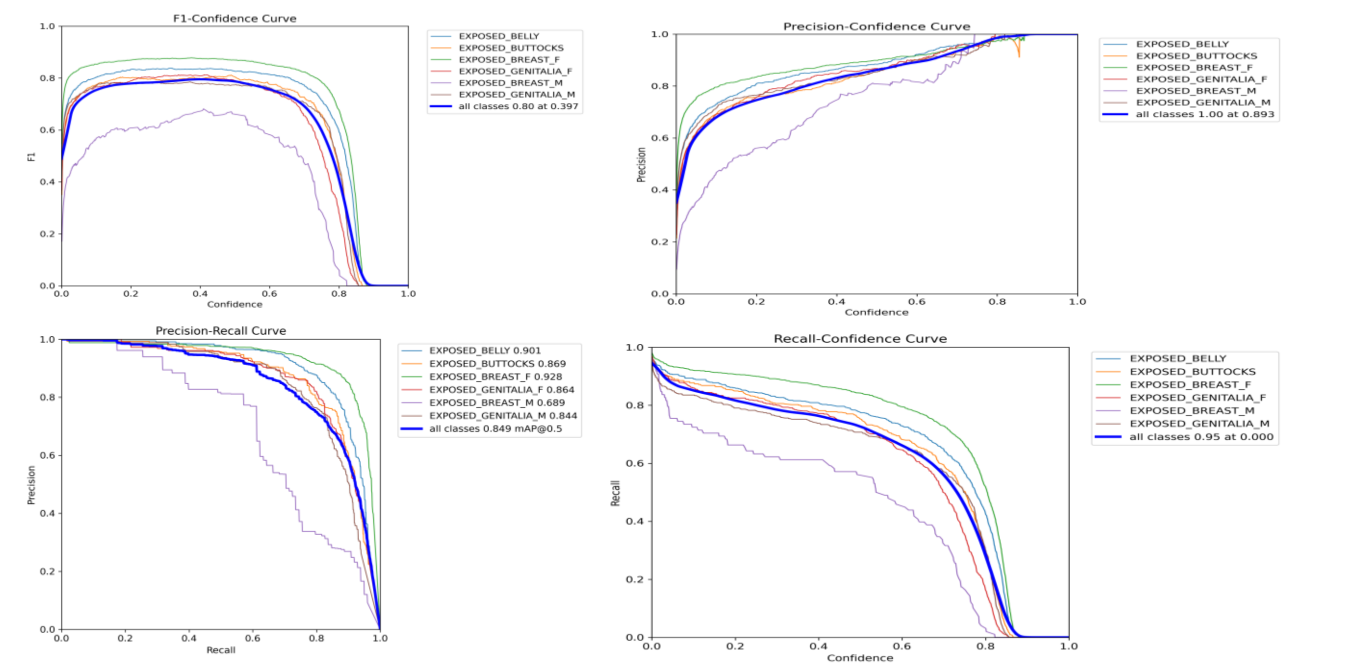

Fig: 10 F1-confidence curve, Precision-Confidence curve, Precision-Recall curve, and Recall confidence curve for 200 epoch.

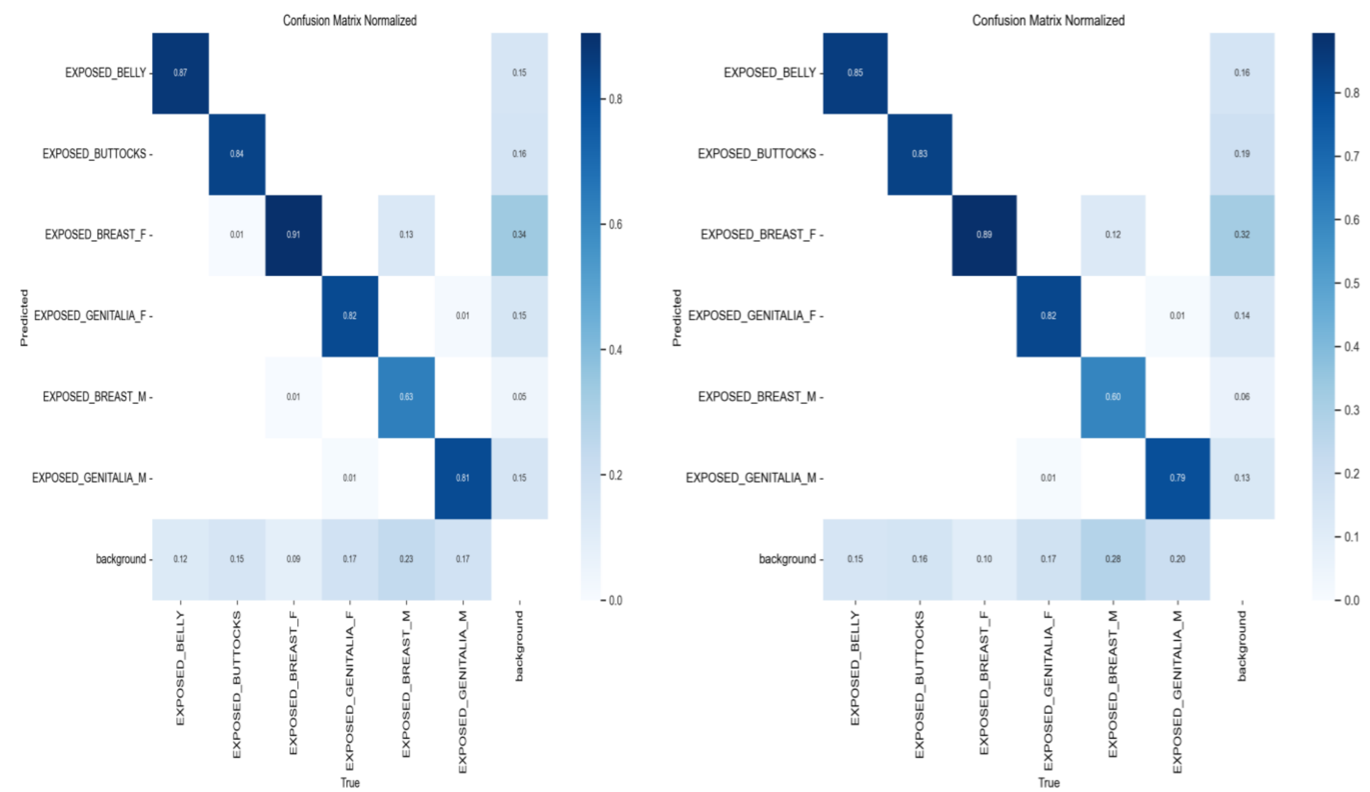

Fig. 11: Left Confusion Matrix 75-epoch and Right Confusion matrix 200 epoch

7. Conclusion and Future Scope

The aim of this research was to filter out NSFW features in media content consisting of images and videos. The research proposes a deep learning framework that utilizes YOLOv11 (You Only Look Once) Ultralytics framework to achieve precise identification of nude contents within a media content. The current results demonstrate that within the available set of training data the number of epochs of training invariably elevates the result. This statement is backed by the **mAP@0.5 (84.91%)** as the **primary detection accuracy component**. The **Precision** value of **(82.77%)** ensures reliable predictions with fewer false positive occurrences. With **recall** value coming close to **(76.53 %)** it provides confidence in the model that detects most objects. As there is no primary metric like classification cases as in this case bounding boxes are the major component and how many objects are bounded and these bounded objects are the right targets,

This research can potentially enhance safe content availability and proper moderation for media uploaded in social media sites and prevent exploitation of such platforms. This work can be improved by providing more training data and possibly expanding the annotations to include more features for NSFW identification. Additionally, improving the time taken to process and filter video content will also be a priority. Alongside that, instead of just identifying content and dropping frames a sequel to this paper would focus on nude object replacement or in-place masking to just eliminate such content from media content instead of removing the frames entirely. Techniques like this would help cater the content to each eligible individual without having the need to alter the source content, even in social media websites and applications it would also prevent unmoderated sharing of such content and keep the platform safe for all users.

Alongside it would be possible to utilize the same concept in entertainment media where the source of the medium not be destroyed and ensure adaptable streaming/viewing of the content based on the age rating, helping produce a content based on the creative vision of the artists and content be filtered on the user side based on the age rating. Thus, helping avoid losses on cutting content to make availability to all users at earlier stage i.e., normalizing it, to lose interest of all audiences, but help cater to each audience by providing the unbiased filtration at the last phase keeping the source content safe and avoiding confusion on cut contents. These would be applicable to streaming platforms, social media and video game industries where the majority of youth and adults spend their leisure time in this era.

REFERENCES

Akyon, et al. (2023) 'State-of-the-Art in Nudity Classification: A Comparative Analysis', *2023 IEEE International Conference on Acoustics, Speech, and Signal Processing Workshops (ICASSPW)*, pp. 1–5. doi: 10.1109/ICASSPW59220.2023.10193621.

Bedapudi, P. (2019) NudeNet classifier dataset. Available at: archive.org.

Comet.ml Documentation (2024) Experiment Management for Machine Learning. Available at: comet.ml.

Duy, et al. (2022) 'LSPD: A Large-Scale Pornographic Dataset for Detection and Classification', *International Journal of Intelligent Engineering and Systems*, 15, p. 198. doi: 10.22266/ijies2022.0228.19.

FFmpeg Documentation (2024) FFmpeg Multimedia Framework. Available at: ffmpeg.org.

Jocher, G. and Qiu, J. (2024) Ultralytics YOLO11. Available at: https://github.com/ultralytics/ultralytics.

OpenCV Documentation (2024) Open-Source Computer Vision Library. Available at: opencv.org.

Leu, W., Nakashima, Y. and Garcia, N. (2024) 'Auditing Image-based NSFW Classifiers for Content Filtering', *The 2024 ACM Conference on Fairness, Accountability, and Transparency (FAccT '24)*, 3–6 June, Rio de Janeiro, Brazil. ACM, New York, NY, USA. doi: https://doi.org/10.1145/3630106.3658963.

Lienhart, R. and Hauke, R. (2009) 'Filtering adult image content with topic models', *2009 IEEE International Conference on Multimedia and Expo*, New York, NY, USA, pp. 1472–1475. doi: 10.1109/ICME.2009.5202781.

Moreira, D. C., Pereira, E. T. and Alvarez, M. (2020) 'PEDA 376K: A Novel Dataset for Deep-Learning Based Porn-Detectors', *2020 International Joint Conference on Neural Networks (IJCNN)*, Glasgow, UK, pp. 1–8. doi: 10.1109/IJCNN48605.2020.9206701.

NSPCC Learning (2022) 'Children's experiences of legal but harmful content online'. Available at: https://learning.nspcc.org.uk/research-resources/2022/helplines-insight-briefing-legal-but-harmful-content [Accessed 9 August 2024].

Perez, et al. (2017) 'Video pornography detection through deep learning techniques and motion information', *Neurocomputing*, 230, pp. 279–293. doi: https://doi.org/10.1016/j.neucom.2016.12.017.

Wasserman, N. et al. (2024) 'Paint by Inpaint: Learning to Add Image Objects by Removing Them First', *arXiv preprint*. Available at: https://arxiv.org/abs/2404.18212 [Accessed 9 August 2024].

Zhelonkin, D. and Karpov, N. (2020) 'Training Effective Model for Real-Time Detection of NSFW Photos and Drawings'. In: van der Aalst, W. et al. *Analysis of Images, Social Networks and Texts. AIST 2019. Communications in Computer and Information Science*, 1086. Springer, Cham. doi: https://doi.org/10.1007/978-3-030-39575-9_31.