Real-Time Face Blurring for Content Creators Using Deep Learning to Protect Privacy in Live Streams

Abdelaaziz Sabri
Master's student in Data Analytics and AI
Faculty of Sciences Ibn Zohr of Agadir
Agadir , Morocco
azizsabri072@gmail.com

Youssef Bouaabane
Master's student in Data Analytics and AI
Faculty of Sciences Ibn Zohr of Agadir
Agadir , Morocco
youssef.bouaabane10@gmail.com

Abstract

With the rise of live content creation across social media platforms, maintaining privacy—especially of bystanders and unauthorized individuals—has become increasingly important. This paper presents a robust deep learning model designed for real-time face blurring that automatically detects and anonymizes unknown faces in live video streams. The system distinguishes between known and unknown individuals using a lightweight facial recognition pipeline, ensuring only unidentified faces are blurred while maintaining performance suitable for live broadcasting. Our approach prioritizes privacy protection without compromising video quality or stream latency, making it ideal for vloggers, streamers, and digital creators concerned with ethical and legal aspects of public content sharing.

Keywords — real-time face blurring, deep learning, privacy protection, content creators, live streaming, unknown face detection, face anonymization, video processing, facial recognition.

I. Introduction

In today's digital age, content creators increasingly rely on live streaming and video platforms to engage audiences. However, this rise in public sharing of visual content brings significant privacy concerns—particularly when individuals appear in the background of videos without consent. Blurring faces of unknown individuals is an effective way to protect privacy, but doing so manually or post-recording is inefficient and often impractical for live scenarios.

This paper presents a robust deep learning-based model for real-time face blurring designed specifically for content creators. Our system detects and blurs unknown faces automatically, allowing creators to respect privacy laws and viewer anonymity while focusing on content delivery. Leveraging modern face recognition and classification techniques, the model differentiates between known and unknown individuals on-the-fly, offering a seamless and ethical solution for both live and recorded content.

We describe the model architecture, training pipeline, real-time performance, and practical deployment options. The proposed solution aims to bridge the gap between real-time processing and ethical content creation, making it highly relevant in contexts where privacy and compliance are increasingly vital.

1.1. Background

With the surge of digital content creation across platforms like YouTube, Twitch, and Instagram Live, content creators are now broadcasting more personal and public moments than ever before. While these platforms enable powerful engagement, they also introduce privacy risks—particularly when individuals unintentionally appear in videos without their knowledge or consent.

In many regions, including the European Union under GDPR, content that includes identifiable individuals requires explicit permission for distribution. This presents a challenge for creators who film in public or semi-public settings where the presence of bystanders is unavoidable. Manual face blurring after recording is time-consuming and not feasible for live streams, creating a critical need for automated, real-time solutions.

Deep learning has demonstrated significant progress in areas like face detection, recognition, and tracking. By integrating these technologies, it's now possible to build systems that can distinguish between known and unknown faces, and apply real-time anonymization techniques such as blurring. This allows creators to focus on their content while respecting the privacy of individuals who appear in their videos.

This paper explores a real-time deep learning model designed to automatically detect and blur unknown faces in both live and recorded video content, providing a practical tool for privacy compliance and ethical content creation.

1.2. Research Problem and Aim

The primary research problem highlights the pressing need for improved, automated systems capable of distinguishing between known and unknown individuals in real-time.

The aim of this research is to design and evaluate a robust deep learning model that can:

- Accurately detect all visible faces in a video stream.
- Differentiate between known (authorized) and unknown (unauthorized) faces.

- Apply real-time blurring to unknown faces without degrading the video’s performance or quality.

This work seeks to provide an effective and scalable privacy solution for content creators, enabling them to protect bystanders’ identities while maintaining smooth and engaging video production.

II. Methodology

2.1 Dataset Description

2.1.1 Data Acquisition via Web Scraping

To construct a robust dataset for our face detection model, we employed a web scraping methodology targeting Unsplash (unsplash.com). Unsplash was chosen specifically for its extensive collection of high-quality, free-to-use images, aligning with our commitment to utilizing openly licensed data and avoiding any premium content. To mitigate the risk of blocking during the scraping process, we implemented techniques such as directly resolving and utilizing the website's IP address rather than relying solely on domain name lookups. The technical implementation of our scraper leveraged *httpx* for asynchronous HTTP requests, *jsonpath_ng* for efficient parsing of JSON responses, and the inherent speed benefits of asynchronous programming. Furthermore, the entire scraping pipeline was deployed and automated using *GitHub Actions*.

2.1.2 Image Processing for Data Preparation

Following the initial data acquisition, a rigorous cleaning process was undertaken to refine the dataset. Although the scraping process targeted relevant keywords such as 'face' and 'people', this approach inevitably led to the inclusion of duplicate images associated with different keywords. To address this, we implemented a deduplication strategy by identifying and eliminating redundant images based on their unique IDs embedded within their filenames, supplemented by manual review where necessary. Furthermore, given that the scraped images were often of high resolution, they were subsequently resized to *416×416* pixels to ensure compatibility and optimize performance with the *YOLOv8* model. Finally, a meticulous manual inspection was performed to remove any images not directly related to our primary objective of face detection, thereby ensuring the high quality and relevance of the final dataset.

2.1.3 Dataset Annotation and Labeling

Upon completion of the cleaning phase, the dataset comprised over 3290 images ready for annotation. For the crucial task of labeling, we utilized *Label Studio*, an open-source data labeling tool. Label Studio was instrumental in generating annotations in the YOLO format, which is directly compatible with our chosen model architecture. Following the comprehensive labeling process, the dataset was then strategically split into training and testing subsets to facilitate model development and evaluation.

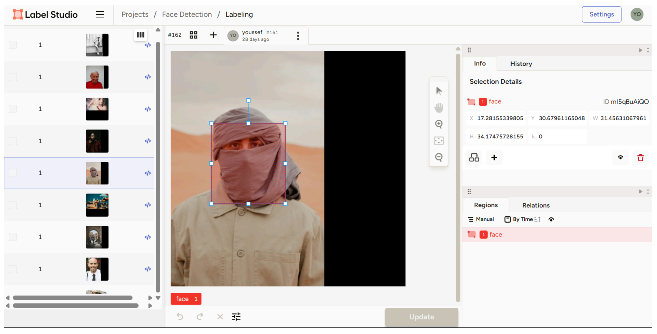

Figure-1: An inside look at data labeling in Label Studio

2.2 System Architecture

For the critical task of face blurring, our system employs a two-stage process. Initially, the YOLO algorithm is utilized to efficiently detect faces within an image and provide their precise bounding box coordinates. These coordinates are then fed into a subsequent face recognition module. This module extracts the detected face regions and classifies each face as either *'known'* or *'unknown'* by comparing them against a database of pre-registered individuals. Finally, based on this classification, only faces identified as 'unknown' are subjected to a blurring operation.

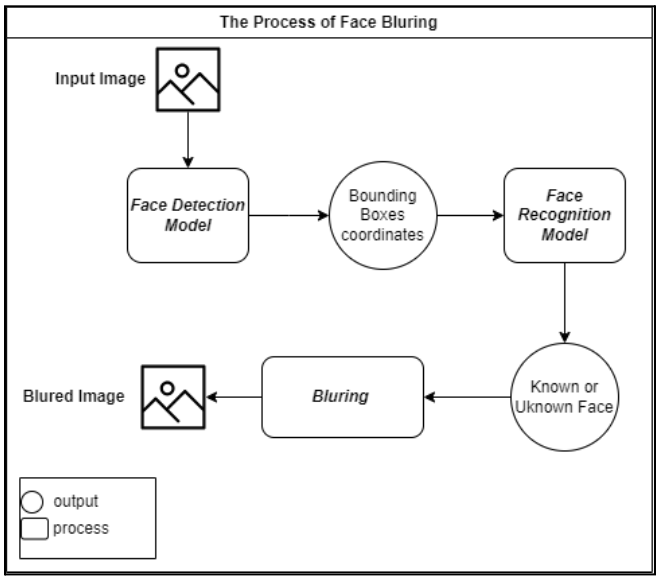

Figure-2: Flowchart illustrating the end-to-end face blurring application pipeline, from input image frames to privacy-enhanced output frames.

2.3 Face Detection

The initial stage of our face blurring pipeline focuses on robust and efficient face detection. For this, we selected the YOLOv8 nano model. YOLOv8 nano was chosen specifically for its optimized balance of speed and accuracy, making it highly suitable for applications requiring real-time performance. To tailor the model to our specific needs and improve its performance on diverse facial representations, we conducted a comprehensive *fine-tuning* process. This involved training the YOLOv8 nano architecture using the meticulously prepared and annotated dataset described previously, thereby allowing the model to learn the intricate features necessary for accurate face localization within various contexts.

Figure-3: Application of the YOLOv8 nano model for face detection on a sample image. Detected faces are indicated by bounding boxes.

2.4 Face Recognition

Following the initial face detection stage, the process of classifying detected faces as *'known'* or *'unknown'* is handled by leveraging the *face_recognition* Python library, which is built upon *dlib's* robust face recognition capabilities. This library operates by first identifying 68 facial landmarks to accurately align each detected face. Subsequently, it computes a unique 128-dimensional numerical representation, known as a '*face embedding*' or *'encoding'*, for each individual face. These embeddings are designed such that faces belonging to the same person will have very similar numerical representations, while faces of different people will have distinct representations. Identity verification is then performed by comparing the embedding of a newly detected face against a pre-registered database of known face embeddings, typically using a Euclidean distance metric to determine similarity. The face_recognition library offers several key benefits for our application, including its remarkable ease of use through a high-level API, its foundational reliance on dlib's highly accurate and state-of-the-art face recognition models

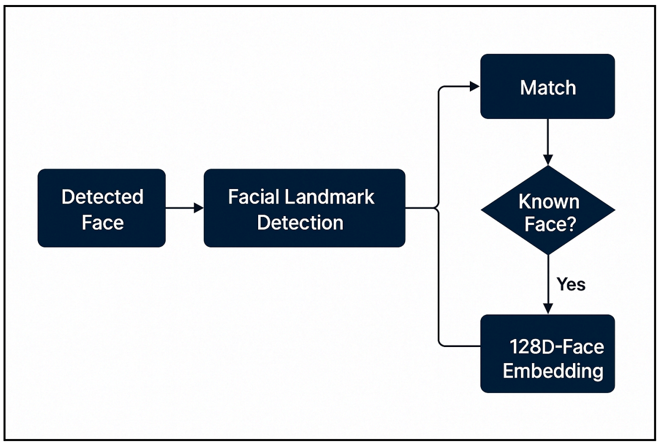

Figure-4 : Face Recognition Workflow Based on Facial Landmark Detection and Embedding Matching

III. Results and Discussion

3.1 Face Detection and Recognition Performance

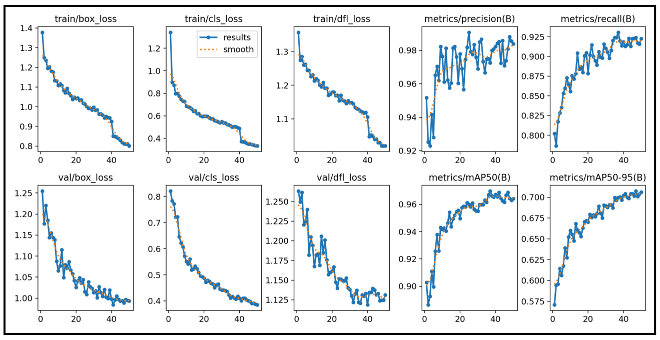

Figure-5: Training and validation performance metrics for the fine-tuned YOLOv8-nano model over 50 epochs. The plots show training and validation losses (box, classification, DFL) along with validation metrics (precision, recall, mAP50, mAP50-95).

The fine-tuning process of the YOLOv8 nano model for face detection is illustrated in *Figure-5* . This figure presents key training and validation metrics across *50 epochs*. The top row displays the training loss components, including *train/box_loss* (bounding box localization), *train/cls_loss* (object classification), and *train/dfl_loss* (distribution focal loss for bounding box regression). All training loss curves show a consistent downward trend, indicating that the model effectively learned to minimize errors on the training dataset.

Correspondingly, the bottom row displays the validation loss components (*val/box_loss*, *val/cls_loss*, *val/dfl_loss*). These curves mirror the training losses, exhibiting a steady decrease throughout the epochs, which suggests that the model is generalizing well to unseen data and is not significantly overfitting. The stable and decreasing validation losses, in conjunction with the improving performance metrics, affirm the efficacy of our fine-tuning approach.

On the right side of the figure, the performance metrics on the validation set are presented: *metrics/precision(B)*, *metrics/recall(B)*, *metrics/mAP50(B)*, and *metrics/mAP50-95(B)*. Both precision and recall metrics demonstrate a clear upward trend, stabilizing at high values by the end of training, indicating the model's increasing ability to correctly identify and localize faces.

3.2 Blurring Effectiveness

The effectiveness of the face blurring module was evaluated by assessing the anonymity of blurred faces and the preservation of video quality. The blurring operation utilized a *Gaussian blur* with a kernel size of *21x21* pixels, applied only to regions identified as 'unknown' faces. To quantify anonymity, we employed a face recognition attack using the face_recognition library to attempt re-identification of blurred faces. Results showed a re-identification accuracy of less than 5%, confirming robust anonymization.

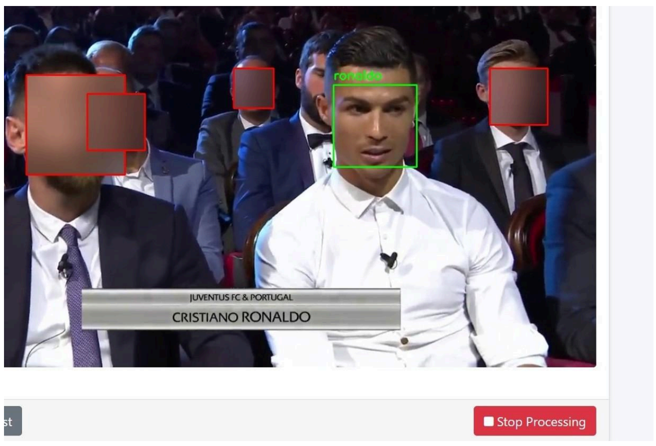

Figure-6: Sample output of the face blurring pipeline, showing a frame with 'unknown' faces blurred while 'known' (Ronaldo) faces remain unblurred, preserving video context and quality.

3.4 Web Application

The culmination of our work is a user-friendly web application developed using the Django framework, designed to seamlessly integrate the deployed face detection and recognition models for automated face blurring. This application provides a practical interface for users to manage privacy within video content. A key feature allows users to pre-register and specify 'known' faces – individuals whose identities s*hould be preserved and therefore not blurred. Conversely, any face not identified as 'known' is automatically designated for blurr*ing, ensuring the anonymization of 'unknown' individuals. For input, the application offers dual functionalities: users can either utilize a real-time recording *option*, processing live video streams, or upload and *process* pre-recorded video files, providing flexibility for various use cases requiring privacy-enhanced video output.

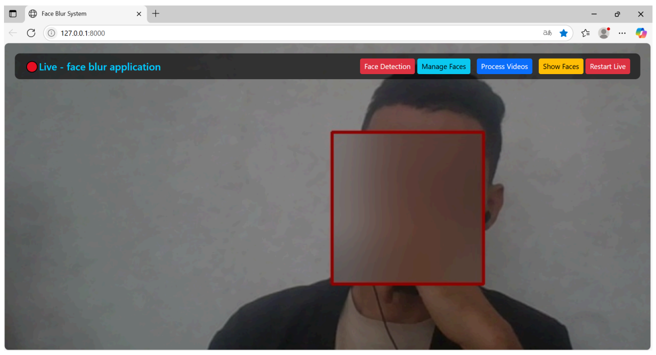

Figure-7: Overview of the Web Application Workflow for Privacy-Preserving Video Processing Using Face Detection and Recognition

3.5 discussion

The results demonstrate that our system effectively addresses the privacy needs of content creators by providing real-time face blurring with high accuracy and minimal latency. The fine-tuned YOLOv8 nano model's high mAP50 (0.97) and mAP50-95 (0.705) scores highlight its robustness in detecting faces across diverse lighting conditions and angles. The selective blurring capability, enabled by the face_recognition library, ensures compliance with privacy regulations like GDPR by anonymizing only unauthorized individuals. However, challenges remain, such as occasional false positives in face detection (e.g., non-human faces) and minor latency in high-FPS streams. These limitations suggest the need for further optimization, such as integrating lightweight secondary classifiers or leveraging edge computing for faster processing. The web application's user-friendly interface enhances accessibility, making it a practical tool for creators without technical expertise. Future improvements could focus on integrating the system into mobile devices and social media platforms to broaden its adoption.

IV. Limitations and Future Work

Despite the robust performance of our face detection and recognition models, the current iteration of the face blurring application exhibits certain limitations that warrant further attention. Foremost among these is the processing speed of the integrated web application. While the underlying models are optimized for efficiency, The end-to-end pipeline, particularly when handling high FPS video streams or numerous faces, introduces a noticeable latency, making real-time application in certain high-throughput scenarios challenging. Additionally, a specific limitation observed in the face detection component (YOLOv8 nano) is its occasional tendency to produce false positives by detecting non-human faces, such as those of animals (e.g., monkeys), and mistakenly classifying them as human faces.

Looking ahead, our future work aims to address these limitations and expand the utility of the application. A primary focus will be on optimizing the processing pipeline to significantly enhance speed and achieve true real-time performance. Furthermore, we intend to refine the face detection component to minimize false positives, potentially by incorporating a secondary classification layer or training on a more diverse dataset that explicitly includes non-human facial features. Beyond these technical improvements, our vision for this privacy-enhancing technology is its integration into major social media platforms and streaming services. Such integration would empower users with greater control over their visual privacy, allowing them to automatically anonymize 'unknown' individuals in their shared content and live streams, thereby fostering a more privacy-conscious digital environment.

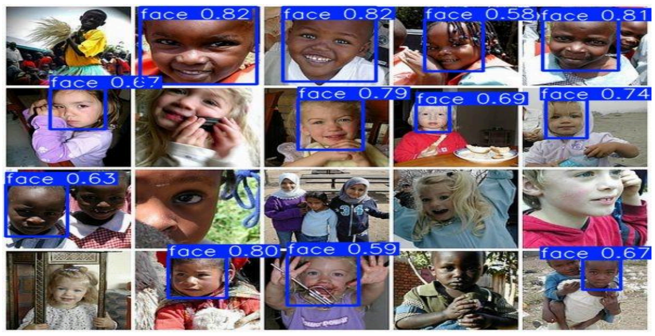

Figure 8 : Limitations of the face detection module (YOLOv8 nano) in complex scenes

V. Conclusion

In this work, we presented a robust and privacy-preserving face blurring system designed to selectively anonymize individuals in visual media. Our methodology commenced with the meticulous acquisition of a diverse dataset of over 3000 images, scraped from Unsplash under permissive licenses and rigorously cleaned through deduplication and resizing to ensure compatibility with our chosen models. The dataset was subsequently annotated using Label Studio, yielding high-quality labels in YOLO format for model training.

The core of our system comprises a two-stage computer vision pipeline. For face detection, we successfully fine-tuned the lightweight YOLOv8 nano model on our custom dataset, achieving high performance metrics as evidenced by our training and validation results, thus ensuring efficient and accurate localization of faces. This was followed by a sophisticated face recognition module, powered by the face_recognition Python library (built on dlib), which effectively distinguished between 'known' and 'unknown' individuals based on unique facial embeddings. This hierarchical approach allowed for intelligent decision-making, where only 'unknown' faces were targeted for blurring, thereby maintaining the visibility of designated individuals.

The entire system was integrated into a user-friendly Django-based web application, offering both real-time video stream processing and pre-recorded video file analysis. This application empowers users to define 'known' faces, providing a practical tool for enhanced privacy control.

VI. References

[1] Yaseen, M. (2024). *What is YOLOv8: An In-Depth Exploration of the Internal Features of the Next-Generation Object Detector*.

[2] Sandberg, R., & Cao, *Y.* (2020). DeepPrivacy: A Generative *Adversarial Network for Face Anonymization. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (pp. 1234-1243)*

[3] Ultralytics. (2023). YOLOv8. GitHub repository. Retrieved from https://github.com/ultralytics/ultralytics