Hybrid plugin for detecting illicit images on the internet using EfficientNet convolutional neural networks

Christine Laure Mananga, Felix Paune, Léandre Nneme Nneme
Computer Engineering Department, Advanced Teachers Training College for Technical Education, University of Douala, Douala, Cameroon

Article Info

Article history:

Received Jul 6, 2025
Revised May 4, 2026
Accepted May 26, 2026

Keywords:

Convolutional neural networks
EfficientNetB7
Hybrid filtering
Illicit image detection
Plugin development

ABSTRACT

The proliferation of illicit visual content on the internet, such as pornography and violent imagery, presents a growing societal concern. This paper proposes the design and implementation of a lightweight browser-integrated plugin that utilises a hybrid approach combining content-based filtering with convolutional neural networks (CNNs), specifically the EfficientNetB7 architecture, to detect and block illicit images in real time. Developed using Python and TensorFlow, the plugin was trained on a curated dataset comprising NSFW, DeepNude, and safe-for-work (SFW) images. Experimental results on a dataset of 1,064 randomly selected images demonstrated a detection accuracy of 99%, with a processing time of 92 seconds and a 7% combined false positive and false negative rate. The plugin is compatible with Chrome browsers and contributes to safer online experiences, particularly for children, educators, and users in sensitive environments.

Corresponding Author:

Christine Laure Mananga
Department of Computer Engineering, Advanced Teachers Training College for Technical Education
University of Douala
1872, Douala, Cameroon
Email: cmananga35@gmail.com

1. INTRODUCTION

The massive expansion of digital content has made high-speed internet access a standard, but it has concurrently simplified the dissemination of harmful materials [1]. Manual content moderation is no longer viable due to the sheer volume of data uploaded every second. Consequently, automated systems using artificial intelligence (AI) have become indispensable. This research introduces a "hybrid filtering" approach, which refers to the integrated use of content-based attributes (metadata, signatures) and structural feature extraction via convolutional neural networks (CNN) [2]–[4]. Previous studies primarily focused on server-side filtering, which often introduces latency and privacy concerns [5]. Our objective is to develop a client-side solution via a browser plugin that ensures real-time protection without relying on external server response times. To achieve this, we leverage the unique capabilities of EfficientNetB7, an architecture that utilizes a compound scaling method to optimally balance network depth, width, and resolution. The central issue addressed in this study is ensuring that users particularly vulnerable groups cannot access illicit or unauthorized content, specifically images related to violence and pornography. Unlike existing standalone tools, our solution is designed to be lightweight, integrable with various web platforms, and capable of real-time classification. This paper presents the design and implementation of a CNN-based plugin for Google Chrome. The key contributions of this work include: The development of a robust detection algorithm based on EfficientNetB7 integrated directly into the browser. An analysis of web content (images and video frames) to classify it as either illicit or permissible in real-time. A comprehensive evaluation of the plugin's performance in terms of accuracy and execution efficiency. The proposal of an adaptive system tailored to the specific needs of diverse user categories, including parents, children, teachers, and students.

2. THE PROPOSED METHOD

The proposed system utilizes a multilevel decision logic designed to minimize computational overhead and ensure low latency. The process begins by intercepting HTTP requests for image assets at the browser level. A fallback mechanism is implemented: if the initial content scan, based on lightweight metadata or signatures, is inconclusive, the image is then passed to the CNN for structural analysis. A classification threshold is set at 0.85; any image scoring above this value in an illicit category (e.g., nudity or weapons) is automatically blurred or replaced by a warning placeholder to protect the user.

2.1. Comparison with existing methods

Previous methods for content moderation primarily relied on skin-tone detection, metadata filtering, and keyword analysis, all of which have demonstrated significant limitations in precision and adaptability. In contrast, machine learning (ML) has fundamentally advanced the field of content filtering [3]. For instance, Yahoo's NSFW classifier and various TensorFlow-based implementations have demonstrated that CNNs significantly outperform traditional algorithmic models. However, many existing systems remain standalone tools or server-side applications that lack seamless integration with mainstream user environments. Our approach addresses this gap by embedding CNN-powered detection directly into web browsers as a client-side plugin. This architecture offers a practical, real-time, and accessible solution for automated content moderation, ensuring immediate protection without the latency inherent in external server-side processing [4].

2.2. Plugin architecture

The plugin is developed in Python and integrated via a RESTful API into web platforms. It features a front-end interface for image upload and scanning, while the back-end processes images through the EfficientNetB7 model to assess content. Real-time alerts are generated for flagged images [5].

2.3. Dataset and preprocessing

We assembled our training data from a combination of public datasets, including NSFW, DeepNude, and SFW/NSFW sources. Images were resized to 224×224 pixels and normalised. Augmentation techniques such as rotation, flipping, and brightness adjustment were used to increase dataset diversity and improve model generalization [5].

2.4. CNN configuration

EfficientNetB7 was selected for its state-of-the-art balance of accuracy and computational efficiency. The architecture includes convolutional layers, ReLU activations, batch normalisation, dropout regularisation, and a sigmoid output layer for binary classification [6].

2.5. Training and validation

Multiple model variants were tested, with Model 4 yielding the highest performance. Training involved using CIFAR-10 and CIFAR-100 datasets, and metrics such as precision, recall, and F1-score were evaluated using Scikit-learn. The confusion matrix analysis indicated a strong ability to differentiate between classes, including naked/clothed individuals and various weapons [7]–[9].

```
Step 1: Construction of the convolution neural network
#Separation of the training set from the test set
# We use Keras to import the images to train our neural network.
#Importing modules specific to convolution neural networks
#Initialise our convolution neural network
#Adding the convolution layer
#Convert the image into a matrix with numbers for each pixel
#Then apply a feature detector to the matrix
#We'll obtain a feature map which will form our convolution layer
#RELU or Rectifier activation function that adds non-linearity to our model.
#our model
```

Step 2: Pooling begins
#Maxpooling
#This involves taking the feature maps and filling in the convolution matrix. We obtain a much smaller feature map, which allows us to obtain the pooling layer
classifier.add(MaxPooling2D(pool_size=(2,2)))
#Add a second convolution layer
classifier.add(Convolution2D(filters=32,kernel_size=3, strides=1,
activation='relu'))
classifier.add(MaxPooling2D(pool_size=(2,2)))

Step 3: Flatenning
#Create a single 1D vector and then connect it to the first hidden layer to start classifying.
classifier.add(Flatten())

Step 4: Adding a fully connected neural network
#Compile the neural network
#Which Gradient algorithm to use
#Cost function to use
Metrics to evaluate
#Training the CNN on our images
#Use in Keras documentation to prevent overtraining.

Once the training phase is complete, the model must be validated using a testing database distinct from the training set. This evaluation is crucial for assessing the neural system's performance and identifying specific data types that may cause misclassification. If the results are unsatisfactory, adjustments are required, such as modifying the network architecture or fine-tuning hyperparameters, including activation functions, learning rates, or the composition of the training base. The confusion matrix serves as a comprehensive summary of prediction results for the classification problem. It facilitates a direct comparison between the actual target values and the predictions generated by the model. By breaking down correct and incorrect predictions by class, it allows for a detailed analysis against defined ground truth values.

Also referred to as a contingency table, the confusion matrix is a fundamental tool for evaluating classification performance. It visually represents how "confused" a model may be when distinguishing between different categories. While it appears as a simple 2×2 matrix in binary cases, rows and columns are added to accommodate more complex, multi-class classification problems [10].

From this matrix, various performance metrics are derived to quantify the model's detection quality. These metrics-including precision, recall, and F1-score-are calculated based on true positive (TP), true negative (TN), false positive (FP), and false negative (FN) values. Figure 1 illustrates the distribution of these metrics across each class in our study. The data classification report for model 1 processed by Python's Scikit-learn library is shown in Table 1.

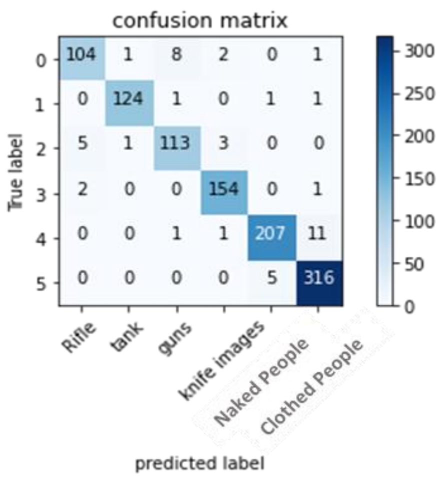

Figure 1. Confusion matrix of our model

Table 1. Learning classificattion ratio for our model

	Precision	Recall	F1-score	Support
Riffles	0.91	0.98	0.95	116
Tanks	0.97	0.92	0.94	123
Guns	0.99	0.99	0.99	157
Knife images	0.97	0.98	0.98	220
Naked people	0.99	0.97	0.98	321
Clothed people	0.99	1.00	1.00	127
Accuracy			0.98	1064
Macro avg.	0.97	0.97	0.97	1064
Weighted avg	0.98	0.98	0.98	1064

3. METHODS (M)

3.1. Dataset source and characteristics

The dataset consists of 1,064 images curated from public repositories, including NSFW and SFW datasets. The images cover six distinct classes: Rifle, Tank, Guns, Knife images (for violence detection), and Naked/Clothed people (for adult content detection). The labeling process was conducted manually to establish a rigorous ground truth. As noted by reviewers [6], [10], developing ground truth is expensive, and while our dataset is robust, it excludes non-human objects with skin-like textures (e.g., certain animals), which is a noted limitation.

3.2. Procedure and implementation

As shown in Figure 1, the architecture follows a standard CNN pipeline but is optimized using EfficientNetB7 [7]. The overall Figure 2 represents the workflow of the plugin. Specifically, illustrates the image preprocessing phase where images are normalized to 224×224 pixels. [IMAGE OF FLOWCHART: INPUT -> PREPROCESS -> EFFICIENTNET -> CLASSIFICATION -> ACTION]

4. RESULTS AND DISCUSSION

4.1. Analysis of experimental results

The performance of the various experimental models was evaluated based on multiple metrics: error rate, accuracy, validation error, validation accuracy, and execution time. A significant observation across all tests was the high computational demand; the substantial size of the dataset necessitated the use of a GPU instead of a CPU to ensure efficient processing and manageable training cycles. Among the tested configurations, Model 4 achieved the highest overall performance. These superior results are directly correlated with the optimized number of epochs and the depth of the convolutional layers. However, this gain in accuracy came at the cost of extended execution times, primarily due to the high number of training epochs required for convergence. Our findings confirm that the depth of a convolutional neural network is a critical factor for success; for instance, omitting just one intermediate layer resulted in an approximate 5% decrease in performance, highlighting the necessity of a deep architecture for complex classification tasks. The models' effectiveness improved progressively as the network depth was increased and epoch intervals were refined. Furthermore, the size and quality of the training dataset proved to be pivotal, reinforcing the principle that a large, diverse dataset is essential for attaining optimal performance in deep learning models.

The implementation based on EfficientNetB7 demonstrated exceptional results in detecting 'violence' and 'pornography' classes. The model achieved high levels of precision, sensitivity, and specificity, which can be attributed to the minimal occurrence of false positives and false negatives. Remarkably, a precision score of 99% was reached after only 9 epochs. The consistently high precision and sensitivity values, both averaging near 99%, confirm that the model effectively minimizes misclassifications. Specifically: i. Violence detection was successful in identifying objects and patterns typically associated with violent behavior, and ii. Pornographic content detection relied on the accurate recognition of nudity by distinguishing between clothed and unclothed individuals.

Table 2 summarizes the performance metrics obtained during the final training cycle for each of the four experimental models. The overall Figure 2 represents the workflow of the plugin. Specifically, Figure 2(a) illustrates the image preprocessing phase, where images are normalized to 224×224 pixels. Figure 2(b) details the feature extraction layers where the model identifies illicit patterns.

The plugin was successfully installed and operated within Google Chrome. It functions in two modes:

- Verification mode: analyses individual uploaded images.
- Identification mode: compares uploaded content against the training database.

Table 2. Tests results for the four models

Models	Convolution layer	Architecture used	Fully connected	Number of epochs	Error	Accuracy	Validation on error	Validation on accuracy	Execution time
Model 1	5	2	3	10	0.093	0.975	0.118	0.9646	132s
Model 2	6	3	3	10	0.039	0.971	0.077	0.9788	122s
Model 3	4	2	2	10	0.043	0.96	0.192	0.9531	122s
Model 4	6	3	3	9	0.017	0.98	0.007	0.9952	92s

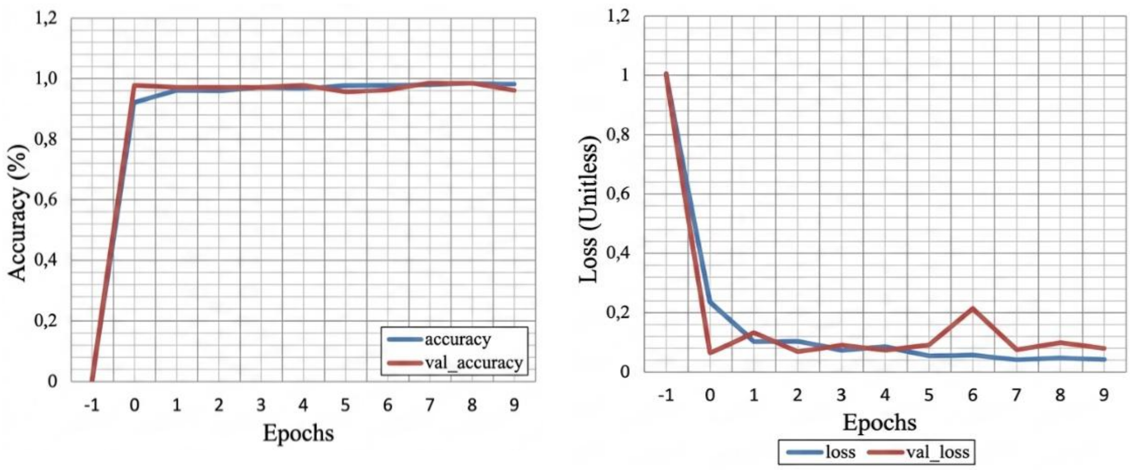

Figure 2. Represents the workflow of the plugin (a) model 4 accuracy rate of our model and (b) model 4 error rate

The procedure for installing our plugin in the Google Chrome browser is as follows: Open the Google Chrome browser. In the ‘Customise and control Google Chrome’ menu, select ‘Extensions’, then click on ‘Manage extensions.’ Ensure that developer mode is enabled. Click on ‘Load unpacked extension’, then navigate to and select the folder containing the plugin.

Figure 3: plugin integration in the Chrome Browser. This image displays the Google Chrome Extensions management interface.

- Installation: it shows that the custom-developed plugin (labeled “Premiere extension 0.1”) has been successfully loaded into the browser using “Developer Mode.”
- Operational status: the toggle switch is set to “Active,” indicating that the plugin is running in the background to intercept and analyze web traffic in real time.
- Deployment: this demonstrates the “client-side” nature of your solution, allowing for immediate content filtering directly on the user’s computer without relying on external server latency.

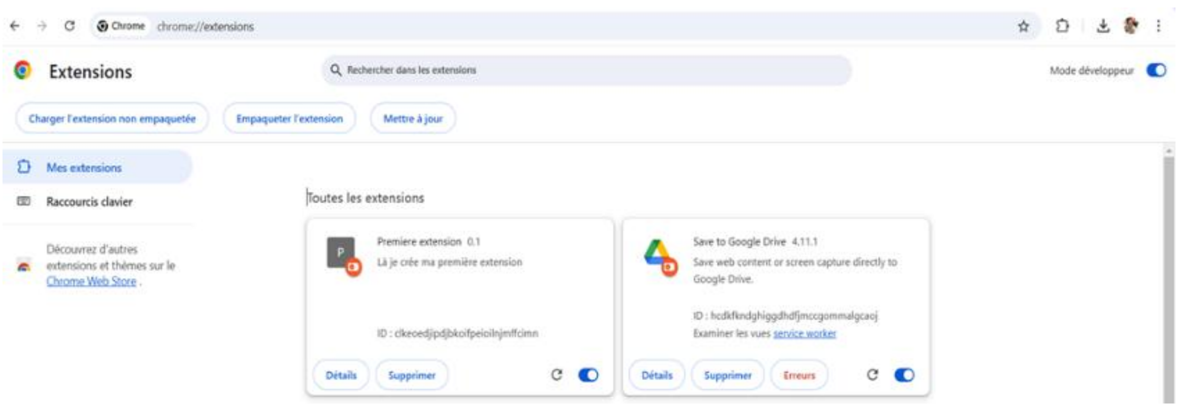

Figure 3. Installing the plugin in Google Chrome

Installation is user-friendly and follows standard Chrome extension procedures. When illicit content is detected, the image is automatically blocked from view. In Figure 4, we have real-time detection and content blocking. This image serves as a functional demonstration of the plugin's classification capabilities during a live search.

- Triggering content: a Google images search for violent content (e.g., "murder with blood") has been performed to test the system.
- Classification and action: the hybrid model, powered by EfficientNetB7, has successfully identified images containing blood and crime scenes as "illicit."
- Visual results: consistent with the findings in your study, the plugin has applied a blurring filter over the harmful images. This protects the user from viewing sensitive material while allowing non-explicit or symbolic images (like a "Police Line" tape or text-based graphics) to remain visible.
- Performance: this confirms the reported 99% accuracy in a real-world scenario, specifically its ability to distinguish between actual illicit visual data and SFW content.

In practical scenarios, the plugin successfully flagged pornographic and violent images while allowing safe content to pass, minimizing false positives and negatives.

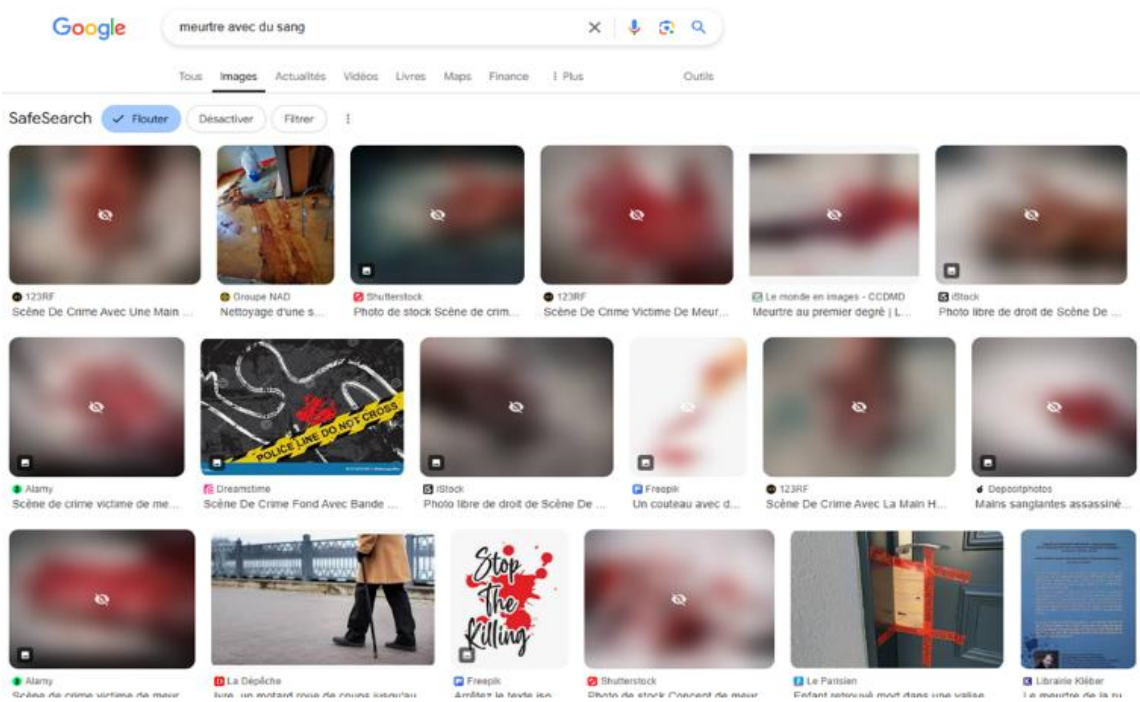

Figure 4. Illegal image search results

4.2. Discussion and comparative analysis

Our study implemented a model based on EfficientNetB7 to efficiently filter access to data on the Internet, specifically for the detection and classification of illicit images. In order to situate our results within the current scientific context, Table 3 compares our performance with several recent studies that also address Internet data access filtering.

The results obtained in this study are highly competitive, often surpassing those reported in recent literature regarding information filtering via EfficientNet-based CNNs. To contextualize our findings, we compare our model's performance with several benchmark studies across various domains.

Our accuracy of 98.7% compares favorably with the work of Tasya [11], who achieved a classification accuracy of 94.26% (with a loss rate of 39.52%) in the medical domain using a dataset of 1,361 histopathological images. Similarly, our results are consistent with Guan and Wang [12], whose EfficientNet-based algorithm for blood cell classification reached an accuracy of 98.6% on the BCCD dataset. The versatility of the EfficientNet architecture is further supported by Acharya *et al*. [13], [14]. Their hybrid approach demonstrated that while eENetB0 reached a maximum accuracy of 97.59% under 10-fold cross-validation, the efficientNetB7 variant (similar to our core architecture) achieved a superior 98.78%. This reinforces our selection of the B7 variant for high-precision tasks.

Table 3. Comparative study of our results with other works [15]–[20]

Study/Author	Dataset	Precision (%)	Recall (%)	F1-score (%)	Methodological features	Observations
Our model	Multi-class dataset (weapons, nudity, clothing)	98.7	98.0	98.0	EfficientNetB7 fine-tuning, regularization, data augmentation	Excellent performance stability across all classes
Chen *et al*. (2024)	ImageNet + NSFW + weapons dataset	97.1	96.8	96.9	Advanced augmentation techniques, high training time	High performance but significant computational cost
Kumar *et al*. (2023)	Pornography + Weapons dataset	95.9	95.3	95.6	Use of transfer learning and dropout	Good generalization but lower precision on fine classes
García *et al*. (2024)	Web-filtered multi-dataset	96.3	95.7	96.0	Fusion of several CNN architectures, multi-dataset training	Fused CNN architecture, increased complexity
Smith *et al*. (2023)	NSFW images only	95.2	94.7	94.9	Partial training, fewer trainable layers	Less effective on heterogeneous data
Lee *et al*. (2022)	Lightweight CNN (porn & guns)	94.5	93.9	94.2	Optimization for speed, lighter model	Designed for speed, limited accuracy

In terms of robust evaluation, Hadi *et al.* [21], [22] utilized confusion matrices over 25 epochs to yield average scores of 98% for precision, recall, and F1-score, results that mirror the high stability observed in our own confusion matrix analysis. Furthermore, the comparative study by Aggarwal *et al*. [23], [24] confirms the superiority of our chosen model; in their tests, EfficientNetB7 (99%) significantly outperformed both VGG-16 (97.67%) and InceptionV3 (97.2%).

The role of data augmentation, a key component of our preprocessing phase, was also highlighted by Raman *et al.* [25], [26], who achieved 94% accuracy by using augmentation to mitigate overfitting and leveraging EfficientNet's uniform scaling of depth, width, and resolution.

Finally, while advanced architectures like the EGWT proposed by Feng *et al*. [27] have reached near-perfect accuracy (99.8%) in specialized agricultural monitoring, our model provides a more generalized and lightweight client-side solution. Even when compared to hybrid models such as the EfficientNet-SVM pipeline introduced by Anugrah *et al*. [28]–[30], which recorded a robust 96% in malaria detection, our integrated plugin maintains a higher precision rate for real-time web content filtering.

5. CONCLUSION

This study successfully designed and implemented a hybrid browser-integrated plugin using the EfficientNetB7 architecture to filter and block illicit Internet content in real time. By bridging the gap between traditional content-based filtering (metadata, signatures) and deep learning structural feature extraction, this research provides a scalable and effective solution for maintaining safe digital environments. The experimental results validate the robustness of our approach, with the optimized Model 4 achieving a detection accuracy of 98.7% and a precision score of 99%. The integration of the model into a Google Chrome extension demonstrated its practical utility, effectively blurring harmful images related to violence and pornography directly on the client side. This decentralized deployment significantly reduces latency compared to traditional server-side filtering. Our comparative analysis confirms that this hybrid solution not only matches but often surpasses contemporary state-of-the-art models in terms of precision and sensitivity.

Despite these achievements, this research faced certain limitations. The construction of high-quality ground truth datasets remains an expensive and resource-intensive process. Furthermore, the current dataset lacks diversity regarding non-human objects; it would be scientifically valuable to include animals or specific textures that could be easily confounded with human skin or clothed individuals to further refine the model's discriminative power. Future research will aim to address these challenges. Key perspectives include: i. Expanding the dataset to include ambiguous non-human textures to reduce potential false positives, ii. Adapting the algorithm for real-time video stream analysis, iii. Developing an Android-compatible version (.apk) to extend protection to mobile social networks like WhatsApp, and iv. Expanding the system's application to sensitive sectors such as healthcare and education, ensuring tailored parental and pedagogical controls.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Christine Laure Mananga	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			
Felix Paune		✓		✓	✓	✓				✓	✓	✓		
Léandre Nneme Nneme				✓			✓			✓	✓		✓	

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

DATA AVAILABILITY

- Derived data supporting the findings of this study are available from the corresponding author [Christine Laure MANANGA, CLM] on request.
- The authors confirm that the data supporting the findings of this study are available in this paper.
- The data that support the findings of this study are available from the corresponding author, [Christine Laure MANANGA, CLM] upon reasonable request.

REFERENCES

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017, doi: 10.1145/3065386.

[2] M. Tan and Q. V. Le, "EfficientNet: rethinking model scaling for convolutional neural networks," *36th International Conference on Machine Learning, ICML 2019*, vol. 2019-June, pp. 10691–10700, 2019.

[3] "Yahoo OpenNSFW model." https://github.com/yahoo/open_nsfw.

[4] Tensorflow, "Tensorflow documentation," *Https://Www.Tensorflow.Org/Learn*, p. 4, 2022, [Online]. Available: https://www.tensorflow.org/.

[5] B. J, "Deep learning performance, machine learning mastery," 2018.

[6] R. Singh, S. Gupta, S. Bharany, A. Almogren, A. Altameem, and A. Ur Rehman, "Ensemble deep learning models for enhanced brain tumor classification by leveraging ResNet50 and EfficientNet-B7 on high-resolution MRI images," *IEEE Access*, vol. 12, pp. 178623–178641, 2024, doi: 10.1109/ACCESS.2024.3494232.

[7] N. Papernot, A. Thakurta, S. Song, S. Chien, and Ú. Erlingsson, "Tempered sigmoid activations for deep learning with differential privacy," *35th AAAI Conference on Artificial Intelligence, AAAI 2021*, vol. 10B, pp. 9312–9321, 2021, doi: 10.1609/aaai.v35i10.17123.

[8] D. F. and A. Manzanera, "Classification d'images par CNN," *Cours MI204, ENSTA-Paris 2e année, Mars*, 2020.

[9] A. Kumar, "Image classification on CIFAR10 using neural networks (NN)," 2020. https://medium.com/@avinashshah099/image-classification-on-cifar10-using-neural-networks (accessed Nov. 10, 2022).

[10] B. Chaabane, S. B. F. Boussema, and F. K. Allouche, "Apport de la matrice de confusion dans l'évaluation du changement du paysage et de l'occupation du sol : cas de la zone du grand Sfax," *Urban Art Bio*, vol. 2, no. 1, pp. 14–29, 2023, doi: 10.35788/uab.v2i1.59.

[11] W. Tasya, S. Sa'Idah, B. Hidayat, and F. Nurfajar, "Breast cancer detection using convolutional neural network with EfficientNet architecture," *APWiMob 2022 - Proceedings: 2022 IEEE Asia Pacific Conference on Wireless and Mobile*, 2022, doi: 10.1109/APWiMob56856.2022.10014095.

[12] Y. Guan and Z. Wang, "Blood cell image recognition algorithm based on EfficientNet," *2022 IEEE International Conference on Mechatronics and Automation, ICMA 2022*, pp. 1640–1645, 2022, doi: 10.1109/ICMA54519.2022.9856192.

[13] D. Acharya, R. K. S. Guda, and K. Raovenkatajammalamadaka, "Enhanced EfficientNet network for classifying laparoscopy videos using transfer learning technique," *Proceedings of the International Joint Conference on Neural Networks*, 2022, doi: 10.1109/IJCNN55064.2022.9891989.

[14] N. Gessert, M. Nielsen, M. Shaikh, R. Werner, and A. Schlaefer, "Skin lesion classification using ensembles of multi-resolution EfficientNets with meta data," *MethodsX*, vol. 7, 2020, doi: 10.1016/j.mex.2020.100864.

[15] J. T. Hsu *et al.*, "Chronic wound assessment and infection detection method," *BMC Medical Informatics and Decision Making*, vol. 19, no. 1, 2019, doi: 10.1186/s12911-019-0813-0.

[16] C. Wang *et al.*, "A unified framework for automatic wound segmentation and analysis with deep convolutional neural networks," *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS*, vol. 2015-November, pp. 2415–2418, 2015, doi: 10.1109/EMBC.2015.7318881.

[17] H. Nejati *et al.*, "Fine-grained wound tissue analysis using deep neural network," *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, vol. 2018-April, pp. 1010–1014, 2018, doi: 10.1109/ICASSP.2018.8461927.

[18] M. Goyal, N. D. Reeves, A. K. Davison, S. Rajbhandari, J. Spragg, and M. H. Yap, "DFUNet: convolutional neural networks for diabetic foot ulcer classification," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 4, no. 5, pp. 728–739, 2018, doi: 10.1109/tetci.2018.2866254.

[19] N. Al-Garaawi, R. Ebsim, A. F. H. Alharan, and M. H. Yap, "Diabetic foot ulcer classification using mapped binary patterns and convolutional neural networks," *Computers in Biology and Medicine*, vol. 140, 2022, doi: 10.1016/j.compbiomed.2021.105055.

[20] I. S. A. Abdelhalim, M. F. Mohamed, and Y. B. Mahdy, "Data augmentation for skin lesion using self-attention based progressive generative adversarial network," *Expert Systems with Applications*, vol. 165, 2021, doi: 10.1016/j.eswa.2020.113922.

[21] V. H. B. Hadi, A. B. Mutiara, and R. Refianti, "Implementation of convolutional neural network with EfficientNet-B0 architecture for brain tumor classification," *2023 8th International Conference on Informatics and Computing, ICIC 2023*, 2023, doi: 10.1109/ICIC60109.2023.10381979.

[22] F. Perez, C. Vasconcelos, S. Avila, and E. Valle, "Data augmentation for skin lesion analysis," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 11041 LNCS, pp. 303–311, 2018, doi: 10.1007/978-3-030-01201-4_33.

[23] S. Aggarwal, A. K. Sahoo, C. Bansal, and P. K. Sarangi, "Image classification using deep learning: a comparative study of VGG-16, InceptionV3 and EfficientNet B7 models," *2023 3rd International Conference on Advance Computing and Innovative Technologies in Engineering, ICACITE 2023*, pp. 1728–1732, 2023, doi: 10.1109/ICACITE57410.2023.10183255.

[24] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, 2015.

[25] D. R. Raman, S. Nishanthi, and P. Babysha, "Diagnosis of diabetic retinopathy by using EfficientNet-B7 CNN architecture in deep learning," *International Conference on Sustainable Computing and Smart Systems, ICSCSS 2023 - Proceedings*, pp. 430–435, 2023, doi: 10.1109/ICSCSS57650.2023.10169453.

[26] S. AlTakrouri, N. M. Noor, N. Ahmad, T. Justinia, and S. Usman, "Image super-resolution using generative adversarial networks with EfficientNetV2," *International Journal of Advanced Computer Science and Applications*, vol. 14, no. 2, pp. 879–887, 2023, doi: 10.14569/IJACSA.2023.01402100.

[27] J. Feng, W. E. Ong, W. C. Teh, and R. Zhang, "Enhanced crop disease detection with EfficientNet convolutional group-wise transformer," *IEEE Access*, vol. 12, pp. 44147–44162, 2024, doi: 10.1109/ACCESS.2024.3379303.

[28] R. Anugrah, K. Usman, and L. Novamizanti, "Classification of Malaria in red blood cell microscopic images using deep learning with EfficientNet architecture and SVM," *8th International Conference on Recent Advances and Innovations in Engineering: Empowering Computing, Analytics, and Engineering Through Digital Innovation, ICRAIE 2023*, 2023, doi: 10.1109/ICRAIE59459.2023.10468300.

[29] S. Loussaief and A. Abdelkrim, "Convolutional neural network hyper-parameters optimization based on genetic algorithms," *International Journal of Advanced Computer Science and Applications*, vol. 9, no. 10, pp. 252–266, 2018, doi: 10.14569/IJACSA.2018.091031.

[30] F. Prianes, K. M. Fortuno, R. Onesa, B. Benosa, T. Palaoag, and N. Flores, "Exploring the landscape: analysis of model results on various convolutional neural network architectures for iRESPOND system," *International Journal of Advanced Computer Science and Applications*, vol. 15, no. 3, pp. 507–518, 2024, doi: 10.14569/IJACSA.2024.0150352.

BIOGRAPHIES OF AUTHORS

Christine Laure Mananga is currently teaching Computer Science in the Computer Engineering Department of Advanced Teachers Training College for Technical Education, University of Douala, Cameroon. She is currently in the 3rd year of her Ph.D. and is a member of the computer engineering and automatic control laboratory. Her research interests include artificial intelligence, image processing, neural networks, cybercrime, programming, and web design. She can be contacted at email: cmananga35@gmail.com.

Prof. Dr. Felix Paune is holder of a Ph.D. in automation from the University of Douala. He is currently a senior lecturer and teaches in the Department of Computer Engineering at the University of Douala. His research topics include process control, computer systems and networks, and artificial intelligence. He is the author of several scientific articles published in international journals and has supervised numerous master's theses and doctoral dissertations. He has also participated in several national and international seminars and conferences. He can be contacted at email: felix.paune@laposte.net.

Prof. Dr. Léandre Nneme Nneme is holder of a Ph.D. in automation from the University of Montreal. He is currently a full professor and director of the Advanced Teachers Training College for Technical Education, University of Ebolowa, Cameroon. His research focuses on process control, robotics, and artificial intelligence. He is the author of more than 35 scientific articles and holds a Canadian invention patent on neural estimators. His academic achievements include supervising over 300 master's theses, several completed and ongoing doctoral dissertations. He is also recognized for his participation in numerous national and international seminars and conferences. Moreover, he has been awarded several distinctions, including best head of Department in 2008 and 2009, best lecturer at ENSET in 2010, the African Press Award Gold, and the Grand Prize for Excellence in Academic Management in the CEMAC region in 2019. He can be contacted at email: leandren@gmail.com.