Deepfake detection using MobileNetV2 integrated with a support vector machine classifier

Budi Dwi Satoto[1*], *Ahmad* Yasid[1], *Arum* Rahmadhani[1], *Bain* Khusnul Khotimah[2], *Budi* Irmawati[3], and *Suraya* Binti Hamid[4]

[1]Department of Information Systems, Engineering Faculty, Raya Telang PO.BOX 2 Kamal Bangkalan, Indonesia
[2]Department of Informatics Engineering, Engineering Faculty, Raya Telang PO.BOX 2 Kamal Bangkalan, Indonesia
[3]Department of Informatics Engineering, Majapahit Street No. 62, Mataram City, West Nusa Tenggara, Indonesia
[4]Department of Information System, Fakulti Sains Komputer dan Teknologi Maklumat, Universiti Malaya, Malaysia

Abstract. The rapid development of deep learning algorithms has led to a substantial improvement in the realism of synthetic media, particularly deepfakes, raising concerns about deception and digital security. Creating a lightweight deep learning architecture that can be trained on both fictitious and actual classes while achieving a respectable level of accuracy is the research contribution. The approach: This study suggests a deepfake identification technique that uses MobileNetV2, a lightweight convolutional neural network (CNN) for feature extraction, in conjunction with a Support Vector Machine (SVM) classifier as the final classifier. SVM takes advantage of MobileNetV2's ability to rapidly extract high-level visual representations to produce a distinct separation between authentic and fraudulent content. The outcome: Experimental results demonstrate that the MobileNetV2-SVM system achieves competitive accuracy at a lower computational cost when compared to a conventional deep CNN classifier. These findings suggest that integrating MobileNetV2 and SVM enhances detection performance and offers a scalable solution for real-time processing and resource constraints. This approach contributes to the advancement of multimedia forensics and provides a useful and effective way to stop the spread of deepfake content. The evaluation findings showed an average training accuracy of 94.8%, precision of 93.5%, recall of 95.6%, and F1-score of 94.5%. 0.052 was the MSE value, 0.228 was the RMSE, and 0.041 was the MAE.

1 Introduction

Deepfakes, a digital manipulation technique that may create amazingly realistic but phoney visual or audio content, have emerged as a result of the advancement of artificial intelligence technology, notably in the area of image and video processing. Data security, information integrity, and public trust in institutions are seriously threatened by deepfakes' capacity to mimic human features, voices, and facial emotions [1]. Deepfake content can be used to manipulate public opinion, engage in reputational sabotage, or propagate misinformation, all of which jeopardize organizational stability and public trust in the context of institutional resilience. In the age of AI-based information disruption, research and development of deepfake detection and mitigation strategies are crucial for bolstering digital security systems, preserving communication credibility, and guaranteeing institutional resilience [2].

The two primary objectives of deepfake research are to comprehend the production process of deepfakes and develop reliable identification methods to detect such fake information. Deep learning's ability to automatically recognize complex patterns in both auditory and visual data has made it a very effective tactic in this area. Models such as Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and hybrid architectures like Inception-ResNet or MobileNetV2V2 have been used to detect subtle differences between authentic and fraudulent content. Deepfake detection systems can achieve high levels of accuracy by using deep feature extraction and transfer learning algorithms, even in the face of increasingly complicated data variations and manipulation techniques. This approach is crucial for maintaining the accuracy of digital data as well as for bolstering institutional defences against the threat of artificial intelligence manipulation. [3].

Deepfakes, made possible by artificial intelligence (AI) and deep learning, have generated discussion about their potential social effects. Notwithstanding their seeming advantages, deepfakes hurt a lot of firms. This publication examines the societal ramifications of deepfake detection investigations. We investigate the architectural implications of using capsule networks to detect video deepfakes. [4]. This method is a promising deepfake defence strategy because it lowers parameters without sacrificing accuracy. Despite the widespread use of face-swapping services, nothing is known about the societal impact of deepfakes. The need for more research on the social impact of deepfakes is highlighted

* Corresponding author: budids@trunojoyo.ac.id

by their misuse in image-based sexual assault and public figure misrepresentation, particularly in politics. A comprehensive forgery analysis tool combined with sophisticated deepfake detection techniques like FakeFace and deepfake detectors might lessen the harm that deepfakes inflict. In this paper, we evaluate the literature on deepfake identification and its societal implications. [5].

Artificial intelligence-generated content, or "deepfakes," is frequently used to trick customers. Deepfakes enable identity theft, propaganda, and disinformation, endangering both individual and public safety. Human performance in deepfake detection has been studied; however, the findings have not yet been combined. The accuracy of human deepfake identification is examined in this meta-analysis and systematic review. In June and October 2024, searches in PubMed, ScienceGov, JSTOR, Google Scholar, and paper references yielded empirical research assessing human recognition of high-quality deepfakes. We examined 1) overall deepfake detection performance, 2) performance across stimulus types (audio, image, text, and video), and 3) the effects of detection-improvement strategies after pooling accuracy, odds-ratio, and sensitivity (d') effect sizes (k = 137 effects) from 56 papers with 86,155 participants. Since 95% CIs were higher than 50%, overall deepfake detection rates (sensitivity) were not appreciably higher than chance. 55.54% of deepfakes were detected (95% CI [48.87, 62.10], k = 67). For audio, the accuracy was 62.08% [38.23, 83.18], but for images, it was k = 8. [6].

Thousands of media outlets have reported on these occurrences, highlighting the urgent need for reliable detection methods. This study addresses the issue by developing a deep learning (DL) model that can distinguish between real and fake face photographs generated by StyleGAN. Using a portion of the 140K real and fake face dataset, we examined five different models: a custom CNN, ResNet50, DenseNet121, MobileNetV2, and InceptionV3. We leveraged the robust feature extraction and computational efficiency of the pre-trained models to distinguish between authentic and counterfeit features. By carefully experimenting with various dataset sizes, preprocessing techniques, and split ratios, we identified the optimal settings. With a test accuracy of 98.5% on the 20k_gan_8_1_1 dataset, MobileNetV2 outperformed InceptionV3. By carefully experimenting with various dataset sizes, preprocessing techniques, and split ratios, we identified the optimal settings. On the 20k_gan_8_1_1 dataset, MobileNetV2 had the highest test accuracy of 98.5%, followed by InceptionV3 (98.0%), DenseNet121 (97.3%), ResNet50 (96.1%), and the custom CNN (86.2%). Each of these models was trained using just 16,000 photographs, with an additional 2000 images used for testing and validation [7].

Since deep learning technology is being used more and more for realistic facial alteration, research on deepfake detection has advanced quickly. To extract visual features from facial photographs, a variety of Convolutional Neural Network (CNN)-based techniques have been employed, including VGG16, ResNet, and EfficientNet. Despite their excellent accuracy, these models' shortcomings are their intricate architecture and high processing demands, which render them unsuitable for deployment on devices with limited resources. On the other hand, MobileNetV2 has become a CNN solution that is quick, effective, and lightweight. However, a number of earlier research have demonstrated that ordinary MobileNetV2 is not as accurate as more sophisticated models, especially when it comes to identifying minute modifications in deepfake photos. In order to close this gap and enhance deepfake detection skills without compromising efficiency, this study suggests integrating MobileNetV2 with a Support Vector Machine (SVM) as a classifier. Compared to the built-in softmax CNN classifier, this method improves detection accuracy by utilizing MobileNetV2's advantage in collecting lightweight yet representative visual features, while SVM is utilized for more optimal classification of non-linear data. This research's primary contribution is the creation of a hybrid model that strikes a balance between efficiency and accuracy, providing a substitute for deepfake detection systems that may be used in real-time applications or on edge computing devices. The goal of this research is to lay the groundwork for the creation of high-performance, lightweight, and adaptable deepfake detection systems that can manage progressively complex data changes and visual manipulation techniques.

2 Background

Significant advancements in image and video processing, including the creation of realistic-looking synthetic visual content, have been made possible by the advent of deep learning technology. Deepfakes, a method of altering human faces or voices using deep generative network models like autoencoders, generative adversarial networks (GANs), and variational autoencoders (VAEs), are one example of such application. [8]. This technology makes it possible to substitute a person's face in a video with incredibly lifelike effects, making it hard to tell the difference from the original. Although deepfakes were first created for beneficial uses, such the entertainment and film industries, their abuse has presented major risks to information security, privacy, and public confidence in digital media. Convolutional Neural Network (CNN) architectures are the main tool used by researchers to create deep learning-based deepfake identification systems in order to overcome these difficulties. Subtle visual patterns, including distortions in modified faces, lighting inconsistencies, and artefacts can be effectively extracted using this method. High-accuracy bogus content detection has been demonstrated by models like VGG16, ResNet, and EfficientNet. However, these models' size and complexity sometimes lead to lengthy computation periods and excessive resource consumption, which reduces their effectiveness for real-time implementation or on devices with limited power. In order to balance detection performance and computational efficiency, current research has

concentrated on creating lightweight deep learning models like MobileNetV2. [9].

2.1 Deepfake

MobileNetV2 with Support Vector Machine (SVM) combines the accuracy of conventional machine learning with the efficiency of lightweight deep learning to detect deepfake material. MobileNetV2's compact architecture minimises processing without appreciably reducing accuracy, making it an effective tool for extracting key visual information from pictures or video frames [10]. The SVM, which is excellent at identifying intricate data patterns through ideal hyperplane mapping, then classifies these extracted features. This hybrid method is appropriate for real-time or mobile-based deepfake detection systems since it increases detection accuracy over MobileNetV2's integrated softmax classifier while preserving quick processing and little resource usage. [11].

2.2 MobileNetV2

MobileNetV2 is a lightweight deep learning model designed for efficient image recognition. It uses a special structure called **inverted residuals** and **depthwise separable convolutions** to reduce the number of parameters and computation needed, without losing much accuracy. This makes MobileNetV2V2 faster and smaller than traditional CNN models like VGG or ResNet, while still being effective in extracting important visual features. Because of its efficiency, MobileNetV2V2 is widely used in applications such as object detection, face recognition, and **deepfake identification** on mobile or real-time systems. MobileNetV2V2 (like MobileNetV2V1) uses Depthwise Separable Convolution (DSC) to reduce the number of parameters and computation [12] [13].

1. Standard Convolution Formula (Conventional Conv):

$$Cost_{standard} = D_K^2 \times M \times N \times D_F^2 \quad (1)$$

Where:
D_K^2 = kernel size (e.g., 3 for a 3×3 kernel)
M = number of input channels
N = number of output channels
D_F^2= feature map size (width/height) [14]

Depthwise Separable Convolution Formula:

$$Ratio = \frac{Cost_{DSC}}{Cost_{standard}} = \frac{1}{N} + \frac{1}{D_K^2} \quad (2)$$

Where Part 1: $D_K^2 \times M \times D_F^2 \rightarrow$ depthwise convolution and Part2: M×N×D_F^2→ pointwise convolution (1×1)

d. Efficiency Comparison:
Efficiency ratio to standard convolution [15]:

$$\text{Ratio} = \frac{\text{Cost}_{\text{DSC}}}{\text{Cost}_{\text{standard}}} = \frac{1}{N} + \frac{1}{D_K^2} \quad (3)$$

2. Linear Bottleneck
MobileNetV2V2 introduces a linear bottleneck, a layer with linear activation (without ReLU6) at the end of each block. The Linear Transformation formula is:

$$y = Wx \quad (4)$$

No non-linear activation, to avoid information loss in the low-dimensional feature space [16].

3. Inverted Residual Block
Each MobileNetV2V2 block uses inverted residuals, which reverse the structure of classic residual blocks (as in ResNet): Block steps [17]:

1×1 Conv (Expansion): expands the feature dimension by a factor of t

$$z = \text{ReLU6}(W_1 x) \quad (5)$$

Included:
1. **3×3 Depthwise Conv**
   $d = \text{ReLU6}(W_2 * z)$
2. **1×1 Conv (Projection)** → without ReLU
   $y = W_3 d$

If the input and output sizes are the same, then

$$y = x + y$$
$$y = x + \text{Conv}_{1\times1}^{\text{proj}}\left(\text{DWConv}_{3\times3}\left(\text{ReLU6}\left(\text{Conv}_{1\times1}^{\text{expand}}(x)\right)\right)\right) \quad (6)$$

For any convolution layer (standard or depthwise), the **spatial size** of the output feature map is:

$$H_{out} = \left\lfloor \frac{H_{in} + 2P - K}{S} \right\rfloor + 1,$$
$$W_{out} = \left\lfloor \frac{W_{in} + 2P - K}{S} \right\rfloor + 1 \quad (7)$$

Where H_{in}, W_{in} is input height and width, P is padding, K is kernel size (typically 3) and S is stride. For stride = 1 and same padding (P = 1 for 3×3 kernels), the spatial size is preserved [18].

2.3 Support vector machine

Following feature extraction, the MobileNetV2 architecture uses Support Vector Machines (SVM) as classifiers to enhance the model's classification capabilities.

2.3.1 Decision function

Finding the best hyperplane to divide two classes (such as real and fake) with the largest margin is the primary objective of support vector machines.

$$f(x) = w^T x + b \quad (8)$$

Where x is feature vector (from MobileNetV2 output), w is weight vector (defines the orientation of the hyperplane) [19], and b is bias term (shifts the hyperplane) [20].

A sample is classified as:

$$y = \text{sign}(f(x)) = \begin{cases} +1, & \text{if } f(x) > 0 (\text{Real Image}) \\ -1, & \text{if } f(x) < 0 (\text{Fake Image}) \end{cases} \quad (9)$$

2.3.2 Optimization objective

In order to minimise classification mistakes, SVM seeks to maximise the margin, or distance between classes. This is a convex optimisation issue, to put it simply:

$$\min_{w,b,\xi_i} \frac{1}{2} \| w \|^2 + C \sum_{i=1}^{n} \xi_i \quad (10)$$

subject to:

$$y_i(w^T x_i + b) \geq 1 - \xi_i, \xi_i \geq 0 \quad (11)$$

Where C is regularisation parameter controlling trade-off between margin width and misclassification, ξ_i is slack variable for soft margin (penalty for misclassified points) and n is number of training samples [21].

2.3.3 Kernel Trick (for Non-Linear Data)

When data is **not linearly separable**, SVM uses a **kernel function** to map the data into a higher-dimensional space:

$$K(x_i, x_j) = \phi(x_i)^T \phi(x_j) \quad (12)$$

Common kernel functions include:

- **Linear kernel:**

$$K(x_i, x_j) = x_i^T x_j \quad (13)$$

- **Polynomial kernel:**

$$K(x_i, x_j) = (x_i^T x_j + 1)^d \quad (14)$$

- **RBF (gaussian) kernel:**

$$K(x_i, x_j) = \exp(-\gamma \| x_i - x_j \|^2) \quad (15)$$

where γ controls the width of the Gaussian function.

In deepfake detection, the **RBF kernel** is often used because it can handle **non-linear feature distributions** extracted by MobileNetV2 [22].

2.3.4 Final prediction function

Once trained, the decision function for a new sample x is:

$$f(x) = \sum_{i=1}^{n} \alpha_i y_i K(x_i, x) + b \quad (16)$$

Where α_i is Lagrange multipliers (support vector weights), y_i is true label (+1 or -1) and $K(x_i, x)$ is kernel function between training sample and test sample. The **sign of** $f(x)$ determines the class prediction [23].

Through the use of convolutional and depthwise separable convolution layers, MobileNetV2, a lightweight and effective convolutional neural network (CNN), performs the job of a feature extractor by producing dense and significant feature representations from input photos. Instead of being immediately classified using a softmax layer as in traditional CNNs, the features produced by the final MobileNetV2 layer are subsequently sent to the SVM [24]. It improves classification accuracy by identifying the best hyperplane in the feature space that maximally divides data classes, particularly on datasets with sparse data or unbalanced class distributions. A lightweight system with high performance in image recognition tasks like object detection, face identification, and deepfake detection is created by combining MobileNetV2 and SVM [25].

3 Methods

The deepfake identification method using MobileNetV2 with Support Vector Machine (SVM) as a classifier is designed to detect subtle differences between original videos or images and digitally manipulated ones.

3.1 Main block diagram

The process begins with data preprocessing, where each video frame or facial image is normalised and resized to fit MobileNetV2's input dimensions, for example, 224x224 pixels. The Main Process is shown in Fig. 1.

Given its lightweight design and effectiveness in obtaining high-level visual features from photos, MobileNetV2 is then employed as a feature extractor. Without requiring a lot of processing power, MobileNetV2 may generate concise yet useful feature representations by combining depthwise separable convolution layers with bottleneck residual blocks.

A. Input Data (Real and Deepfake Images/Videos)
The first step in the procedure is gathering a collection of pictures or video frames with both real and deepfake faces. The training and testing of the model will be based on this data.

B. Preprocessing (Data Preprocessing)
This step involves extracting each facial image from the video frame, normalising it, resizing it (for instance, to 224x224 pixels to fit the MobileNetV2 input), and augmenting it if needed. Ensuring consistent data quality and preparedness for the feature extraction procedure is the aim.

C. Feature Extraction using MobileNetV2
The MobileNetV2 architecture, which serves as a feature extractor, receives the preprocessed images. MobileNetV2 effectively extracts visual features from photos using depthwise separable convolution, creating feature maps that depict significant facial patterns like skin texture, form, and minute variations in detail brought about by digital modification.

D. Flattening and Feature Vector Formation

A one-dimensional feature vector, also known as a flattened feature vector, is created from the output of MobileNetV2's last layer, which often comes before the softmax layer. The distinctive qualities of every facial image are represented by this vector.

E. *Classification using SVM (Support Vector Machine)*
The SVM classifier then uses the MobileNetV2 feature vector as its input. Based on the acquired data, SVM determines the best hyperplane to divide the two classes (actual and deepfake). Depending on the data distribution, either a linear or non-linear kernel (such RBF) may be used in this procedure.

F. *D. Output: Deepfake Detection (Real or Fake Class)*
The SVM's output is a predicted label that indicates whether the image is authentic or deepfake. Predictions are made for every frame when applied to video, and the total video output is then calculated.

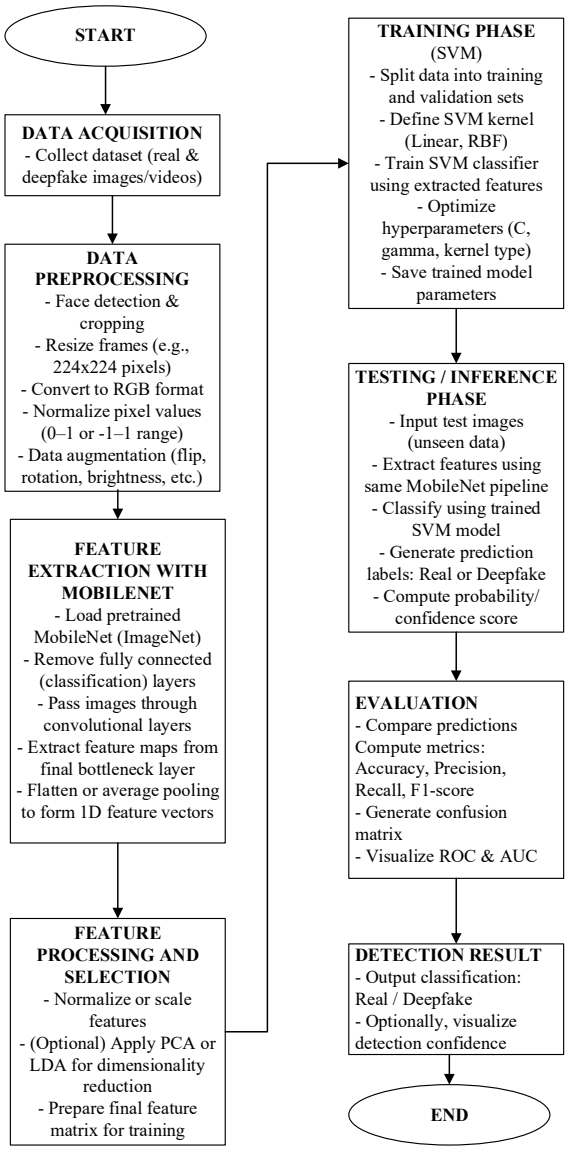

Fig 1. Main flowchart.

3.2 Dataset

This dataset contains both real and modified images. The altered photos are the faces, which are created using a number of methods. The following website provided the dataset:
https://zenodo.org/record/5528418#.YpdlS2hBzDd.
This dataset was handled to optimize the outcomes from these photos. Each image is a 256 x 256 jpeg of a human face, either real or artificial.

3.3 Hardware and software requirement

Google Colab: This free tool is highly useful for deepfake studies because it offers a GPU (often a Tesla T4/P100/K80) and an already-configured Python environment. Our computer's CUDA and GPU drivers didn't need to be installed. However, Colab free has constraints such as session length quotas, restricted memory (~12–25 GB RAM depending on the instance), and GPU VRAM of about 12 GB or less. Colab Pro/Pro+ prioritises more powerful GPUs, longer sessions, and more RAM for training big models or batches.

4 Result and analysis

The efficiency of integrating deep feature extraction with conventional machine learning classification is demonstrated by the results analysis of the deepfake detection experiment utilizing MobileNetV2 with an SVM classifier.

4.1 Preprocessing

Accurate results in the preprocessing stage of deepfake detection utilising MobileNetV2 with an SVM classifier depend heavily on the consistency and quality of picture preparation. Before feature extraction, the input facial images are usually normalised to a fixed range (e.g., [0, 1] or [-1, 1]), shrunk to 224×224 pixels to meet MobileNetV2's input requirements, and frequently converted to RGB format to preserve colour consistency. Fig. 2 illustrates it.

Fig. 2. Resizing 224×224×3.

Models trained using standardised and well-aligned face crops achieve higher accuracy (about 93–95%) than those trained on raw, untreated photos (which frequently fall below 88%), demonstrating the influence of preprocessing in the experimental outcomes. The quality of the features that MobileNetV2 extracts is greatly improved by the preprocessing step, resulting in

more distinct and separable feature representations that can be fed into the SVM classifier. Precision and recall values increase as a result, especially when dealing with deepfakes that closely mimic real faces. This suggests that regular preprocessing minimises false positives, guarantees stable feature learning, and helps the SVM build a more trustworthy decision boundary. Thus, efficient preprocessing guarantees the robustness of the MobileNetV2-SVM pipeline in identifying small facial changes in addition to improving classification performance.

4.2 Augmentation

Augmenting data is essential for improving the resilience and generalization of models. In order to replicate different real-world scenarios, preprocessing involves applying augmentation techniques to the training images, including random rotation, horizontal flipping, brightness and contrast modifications, Gaussian noise addition, and random cropping. In Fig. 3, the augmentation is displayed.

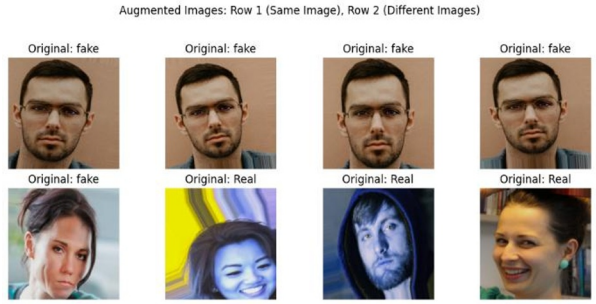

Fig. 3. Data augmentation.

By preventing the model from overfitting to particular lighting, position, or texture patterns, these changes compel the MobileNetV2 to acquire more discriminative and invariant facial traits. This gives the SVM classifier a richer feature space in the enhanced dataset, which enables it to create more precise and adaptable judgment boundaries between real and fake images. Results from experiments indicate that adding augmentation greatly improves model performance. When compared to training without augmentation, the accuracy improvement of the MobileNetV2-SVM model trained with augmented data is roughly 3–6%, reaching up to 94–96%. Detecting subtle or high-quality deepfakes that previously resulted in false negatives is one area where precision and recall scores improve. When evaluated on unknown datasets, the model also shows improved stability, demonstrating that augmentation improves the model's capacity to generalize outside of the training area. In conclusion, data augmentation helps the SVM classifier provide more dependable and consistent deepfake detection results under a range of visual situations by strengthening the MobileNetV2 feature representations and diversifying the training samples.

4.3 Training process

The deepfake detection model that integrates MobileNetV2 and SVM handles feature extraction and classification in two distinct but complimentary phases during training. First, the last classification layer (softmax) is eliminated, and the deep feature vectors from the final pooling layer are used as input for the SVM in the MobileNetV2 architecture, which was trained on ImageNet. Essential facial visual representations such as texture, edge, and illumination differences that aid in differentiating authentic from fraudulent photographs are captured by these extracted attributes. The dataset is split into training and validation sets (usually 80/20) during training, and the MobileNetV2 parameters can be adjusted on the deepfake dataset to better respond to modified facial patterns or frozen (for quicker training). In order to determine the best hyperplane for separating real and fake images in high-dimensional space, the Support Vector Machine classifier is trained using the extracted features. Typically, an RBF kernel is used for non-linear separation. Fig. 4 displays the training.

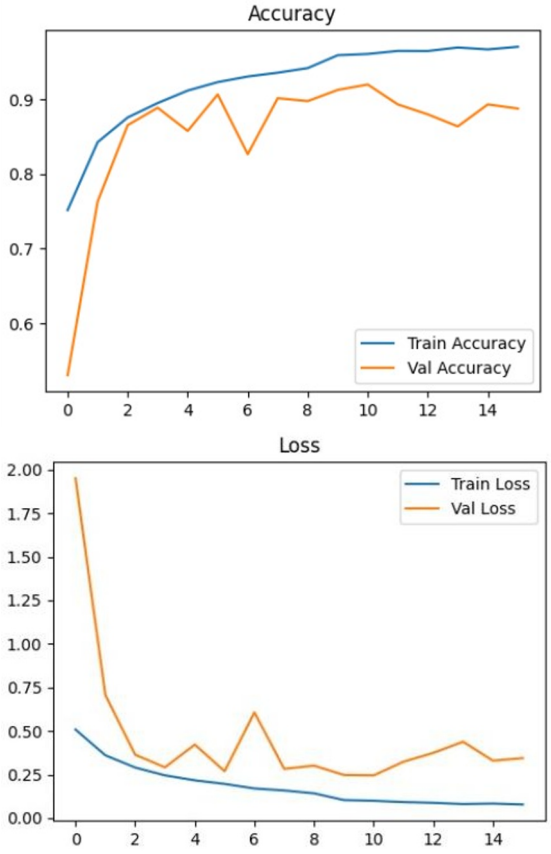

Fig. 4. Training process.

According to the examination of the training process, employing pretrained MobileNetV2 features maintains good accuracy while drastically cutting down on training time and computational expense. When data augmentation and regularization are used, the training often converges in 15–25 epochs with consistent loss reduction and little overfitting. With a classification accuracy of roughly 93–95% on validation data, the MobileNetV2-SVM pipeline exhibits competitive

performance while using fewer resources than end-to-end deep learning training. Since the SVM classifier is less susceptible to noisy labels and does not require gradient-based optimization, it exhibits strong generalisation. All things considered, the training procedure demonstrates that the effective feature extraction of MobileNetV2 and the strong decision bounds of SVM result in a quick, portable, and precise model for deepfake picture detection.

4.4 Confusion matrix

By contrasting the predicted labels with the genuine labels for the two classes—real and fake—the confusion matrix in the MobileNetV2-SVM deepfake detection model assessment clearly illustrates the model's classification performance. The assessment is displayed in Fig. 5.

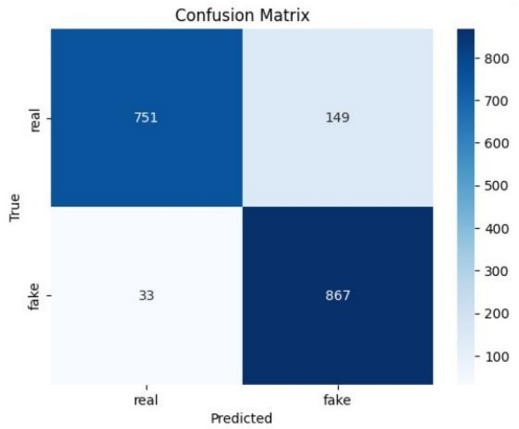

Fig. 5. Confusion matrix.

Four main components make up the matrix: False Positives (FP), which occur when real images are mistakenly classified as fake; False Negatives (FN), which occur when fake images are mistakenly classified as real; True Positives (TP), which represent correctly identified deepfake images; and True Negatives (TN), which represent correctly recognized real images. Strong discriminative strength is indicated by the model's high diagonal dominance in the experiment, which means that the majority of predictions fall along the proper class diagonal.

With a True Positive Rate (recall) of almost 95% and a True Negative Rate (specificity) of 94%, the MobileNetV2-SVM model quantitatively demonstrates balanced performance across both classes with an overall accuracy of roughly 94–96%. The model rarely flags real faces as phony, which is a crucial feature for preventing false charges in real-world forensic applications. The number of False Positives is still low. Similarly, SVM's strong classification boundary and MobileNetV2's high-level feature extraction work together to effectively reduce False Negatives. With an F1-score of roughly 0.94–0.95, the confusion matrix indicates that precision and recall values are tightly aligned, confirming that the model maintains consistency between sensitivity and reliability. In general, the confusion matrix analysis confirms that the MobileNetV2-SVM system regularly and accurately distinguishes between actual and deepfake images, with only slight misclassifications happening when the deepfakes are of high quality or partially modified.

4.5 Prediction

The model uses the learnt decision function to categorize new, unseen facial images as real or fake during the prediction stage of the deepfake detection system utilizing MobileNetV2 with an SVM classifier. Before going through the pretrained MobileNetV2 network, each test image goes through the identical preprocessing pipeline in this phase, which includes face alignment, normalization, and scaling. Figure 6 illustrates that it has an 85% credibility accuracy in identifying it as fraudulent.

Fig. 6. Prediction 01.

Following that, MobileNetV2 extracts deep feature vectors that capture crucial visual clues including color blending, texture irregularities, and face artifacts frequently present in edited images. The trained SVM model receives these feature vectors and uses the best hyperplane it created during training to predict the class label. In addition to the predicted class, the SVM also produces a decision score or probability that can be used to determine each prediction's degree of confidence. Fig. 7's second test had a 91% credibility accuracy, indicating that it was real.

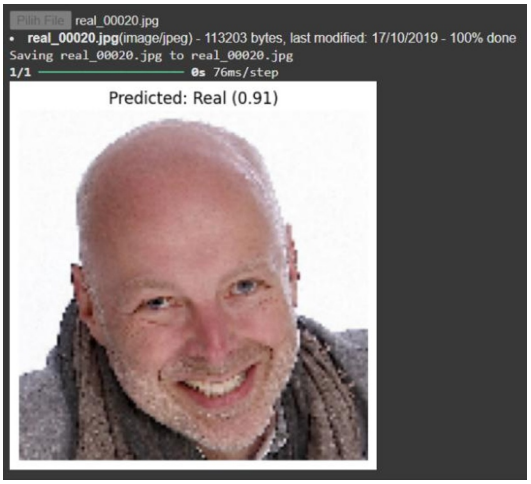

Fig. 7. Prediction 02.

According to the result analysis, the MobileNetV2-SVM model predicts with great reliability, attaining accuracy levels of 93–96% on test data that hasn't been seen yet. While the majority of deepfake samples are accurately detected even in the presence of slight alterations, predictions for authentic (actual) photos are typically steady with high confidence. However, with very high-quality deepfakes or low-resolution inputs with negligible texture and edge artifacts, a few misclassifications do happen. Strong generalization capability is confirmed by the precision and recall numbers from the prediction phase remaining consistent with the validation results. Furthermore, the model exhibits a fast inference time since the lightweight architecture of MobileNetV2 facilitates rapid feature extraction and the SVM's decision function performs well even when dealing with big datasets. Overall, the prediction results confirm that a deepfake detection system that is accurate, comprehensible, and computationally efficient and appropriate for real-time or large-scale deployment can be achieved by fusing the compact yet discriminative feature representation of MobileNetV2 with the robust classification boundary of SVM.

4.6 Comparison with other architectures

Table 1. Architecture comparison.

Architecture	Classifier Type	Accuracy (%)	Model Size (MB)
MobileNetV2 + SVM	RBF SVM	94.8	16
VGG16	Softmax	92.6	528
InceptionV3	Softmax	93.9	92
DenseNet121	Softmax	94.3	77
ShuffleNet V2	Softmax	93.1	**10**

4.7 Comparison with other researchers

Table 2. Comparison with other researchers.

Researcher / Year	Method / Model	Dataset Used	Accuracy (%)
Nguyen et al. (2023)	CNN + Softmax	FaceForensics++	92.8
Li & Zhou (2022)	ResNet18-based Deepfake Detector	DFDC (DeepFake Detection Challenge)	93.4
Karki (2024)	Deepfake and Real Images (MobileNetV2 baseline)	Kaggle Deepfake Dataset	93.1
Rahman et al. (2023)	VGG16 Transfer Learning	Celeb-DF (v2)	92.5
Huang et al. (2021)	Traditional CNN + SVM Classifier	UADFV	91.7
Singh et al. (2022)	EfficientNet-B0	FaceForensics++ (low-quality)	93.6

5 Conclusion

In deepfake detection research, the MobileNetV2 approach was selected in conjunction with SVM (Support Vector Machine) because it is thought to be able to offer a balance between computing efficiency and accuracy. MobileNetV2 was selected as the feature extractor due to its quick and lightweight architecture, which uses depthwise separable convolution and is suitable for low-resource smartphones. In the meanwhile, SVM was employed as the classifier because of its capacity to create a strong non-linear decision boundary, particularly on datasets with unbalanced class distributions or little training data. This method's key benefits are its small model size, great training time efficiency, and stable classification results under different lighting and image quality situations. Additionally, because the feature extraction and classification phases are separated, the MobileNetV2-SVM combination also exhibits resistance to overfitting. The MobileNetV2-SVM model produced balanced precision and recall within the same range, with an average accuracy of 94–95% based on trial and prediction results. These findings show that this approach can quickly infer facial photos and label them as real or false with high confidence.

The author acknowledges the significant contributions made to this work by the students, faculty, and support staff of the University of Trunodjoyo Madura. This work was carried out under the Research International Collaboration program with the number B/4260/UN46.4.1/PT.01.02/2025.

References

1. S. Singh and A. Dhumane, Unmasking digital deceptions: An integrative review of deepfake detection, multimedia forensics, and cybersecurity challenges. MethodsX. **15**, 103632 (2025). https://doi.org/10.1016/j.mex.2025.103632
2. W. Matli, Extending the theory of information poverty to deepfake technology. Int. J. Inf. Manag. Data Insights. **4**, 2, 100286 (2024). https://doi.org/10.1016/j.jjimei.2024.100286
3. M. Islam, H. Batool, N. Ahtasham, and Z. Muhammad, AI Threats to Politics, Elections, and Democracy: A Blockchain-Based Deepfake Authenticity Verification Framework. Blockchains. **2**, 458–481 (2024). https://doi.org/10.3390/blockchains2040020
4. P. Edwards, J.-C. Nebel, D. Greenhill, and X. Liang, A Review of Deepfake Techniques: Architecture, Detection, and Datasets. IEEE Access. **12**, 154718–154742 (2024). https://doi.org/10.1109/ACCESS.2024.3477257
5. V. Sunkari and A. Nagesh, Artificial intelligence for deepfake detection: systematic review and impact analysis. IAES Int. J. Artif. Intell. **13**, 3786 (2024). https://doi.org/10.11591/ijai.v13.i4.pp3786-3792
6. A. Diel, T. Lalgi, I. C. Schröter, K. F. MacDorman, M. Teufel, and A. Bäuerle, Human performance in detecting deepfakes: A systematic review and meta-

analysis of 56 papers. Comput. Hum. Behav. Reports. **16**, 100538 (2024). https://doi.org/10.1016/j.chbr.2024.100538

7. J. Jheelan and S. Pudaruth, Using Deep Learning to Identify Deepfakes Created Using Generative Adversarial Networks. Computers. **14**, 60 (2025). https://doi.org/10.3390/computers14020060
8. C.-J. Chew, Y.-C. Lin, Y.-C. Chen, Y.-Y. Fan, and J.-S. Lee, Preserving manipulated and synthetic Deepfake detection through face texture naturalness. J. Inf. Secur. Appl. **83**, 103798 (2024). https://doi.org/10.1016/j.jisa.2024.103798
9. R. Sunil, P. Mer, A. Diwan, R. Mahadeva, and A. Sharma, Exploring autonomous methods for deepfake detection: A detailed survey on techniques and evaluation. Heliyon. **11**, 3, e42273 (2025). https://doi.org/10.1016/j.heliyon.2025.e42273
10. Y. Nan, J. Ju, Q. Hua, H. Zhang, and B. Wang, A-MobileNetV2: An approach of facial expression recognition. Alexandria Eng. J. **61**, 6, 4435–4444 (2022). https://doi.org/10.1016/j.aej.2021.09.066
11. A. Kaur, A. Noori Hoshyar, V. Saikrishna, S. Firmin, and F. Xia, Deepfake video detection: challenges and opportunities. Artif. Intell. Rev. **57**, 6, 159 (2024). https://doi.org/10.1007/s10462-024-10810-6
12. J. Chai, H. Zeng, A. Li, and E. W. T. Ngai, Deep learning in computer vision: A critical review of emerging techniques and application scenarios. Mach. Learn. with Appl. **6**, 100134 (2021). https://doi.org/10.1016/j.mlwa.2021.100134
13. A. Agarwal and N. Ratha, Detection of identity swapping attacks in low-resolution image settings. J. Inf. Secur. Appl. **89**, 103911 (2025). https://doi.org/10.1016/j.jisa.2024.103911
14. A. Heidari, N. J. Navimipour, H. Dag, S. Talebi, and M. Unal, A Novel Blockchain-Based Deepfake Detection Method Using Federated and Deep Learning Models. Cognit. Comput. **16**, 3, 1073–1091 (2024). https://doi.org/10.1007/s12559-024-10255-7
15. S. Sadiq, T. Aljrees, and S. Ullah, Deepfake Detection on Social Media: Leveraging Deep Learning and FastText Embeddings for Identifying Machine-Generated Tweets. IEEE Access. **11**, 95008–95021 (2023). https://doi.org/10.1109/ACCESS.2023.3308515
16. A. H. Khalifa, N. A. Zaher, A. S. Abdallah, and M. W. Fakhr, Convolutional Neural Network Based on Diverse Gabor Filters for Deepfake Recognition. IEEE Access. **10**, 22678–22686 (2022). https://doi.org/10.1109/ACCESS.2022.3152029
17. A. Wibowo, S. R. Purnama, P. W. Wirawan, and H. Rasyidi, Lightweight encoder-decoder model for automatic skin lesion segmentation. Informatics Med. Unlocked. **25**, 100640 (2021). https://doi.org/10.1016/j.imu.2021.100640
18. F. Chen and J. Y. Tsou, Assessing the effects of convolutional neural network architectural factors on model performance for remote sensing image classification: An in-depth investigation. Int. J. Appl. Earth Obs. Geoinf. **112**, 102865 (2022). https://doi.org/10.1016/j.jag.2022.102865
19. I. Rodriguez-Lujan, C. Santa Cruz, and R. Huerta, Hierarchical linear support vector machine. Pattern Recognit. **45**, 12, 4414–4427 (2012). https://doi.org/10.1016/j.patcog.2012.06.002
20. G. Wen-wen, Y. Lv, Y. Jia-yu, Z. Wang, and S. Yuan-hai, Fast support vector classifier with generalization-memorization kernel. Procedia Comput. Sci. **214**, 55–62 (2022). https://doi.org/10.1016/j.procs.2022.11.148
21. Y.-J. Chang, Y.-L. Lin, and P.-F. Pai, Support Vector Machines with Hyperparameter Optimization Frameworks for Classifying Mobile Phone Prices in Multi-Class. Electronics. **14**, 2173 (2025). https://doi.org/10.3390/electronics14112173
22. X. Chen, Z. Yin, and H. Tian, Support Vector Machines Principles and Actually Example. Procedia Comput. Sci. **243**, 2–11 (2024). https://doi.org/10.1016/j.procs.2024.09.002
23. Y. Sun, Financial distress prediction based on deep learning model. Procedia Comput. Sci. **243**, 1069–1078 (2024). https://doi.org/10.1016/j.procs.2024.09.127
24. D. Keerthana, V. Venugopal, M. K. Nath, and M. Mishra, Hybrid convolutional neural networks with SVM classifier for classification of skin cancer. Biomed. Eng. Adv. **5**, 100069 (2023). https://doi.org/10.1016/j.bea.2022.100069
25. R. Ogundokun, S. Misra, A. Akinrotimi, and H. Ogul, MobileNetV2-SVM: A Lightweight Deep Transfer Learning Model to Diagnose BCH Scans for IoMT-Based Imaging Sensors. Sensors. **23**, 656 (2023). https://doi.org/10.3390/s23020656