**“Face detection & recognition system for enhancing Cybersecurity”**

**Ali E. Takieldeen1\*, Mohamed saad2, Ahmed khaled2, Ahmed Alaa2,**

**Ibrahim Tarek2 , Ahmed Wael2**

*1 IEEE Senior Member Faculty of Artificial Intelligence, Delta University for Science and Technology,*Gamsa,Egypt *2Department of cybersecurity, Faculty of Artificial intelligence, Delta University, Costal International Road, Dakahlia, Egypt.*

*Aya El-Sayed El-Metwaly Faculty of Artificial Intelligence, Delta University for Science and Technology,Gamsa, Egypt.*

**Abstract**

Achieving high accuracy in real-time face detection is crucial for enhancing security measures in various domains. In this project, a novel approach is proposed to improve security by leveraging a hybrid hyper model composed of ResNet-50 and DenseNet-121. ResNet-50's residual structure addresses gradient disappearance, enabling deeper networks and better gradient flow, while DenseNet-121's dense connectivity pattern promotes superior information flow and parameter utilization. The methodology involves training the hybrid hyper model on face datasets under varying real-world conditions, simulated through random brightness adjustments. Using OpenCV, the model's performance is evaluated, achieving an impressive accuracy of 96%. The results demonstrate the effectiveness of this approach in real-time face detection, offering a reliable solution for security applications. This research contributes to advancing security measures by providing a robust and accurate face detection system capable of identifying individuals with high precision in diverse environments.

**Keywords:** ResNet-50, DensNet-121, Face Detection, Deep Learning.

**1. Introduction:**

Face recognition technology has become increasingly prominent within the realm of computer science, particularly as a biometric solution. With the ongoing advancements in deep learning and artificial intelligence, significant strides have been made in the performance of AI systems. This progress transcends security applications, as face recognition finds utility in various domains, including mobile device authentication, digital payments, network access, and more. Its widespread adoption offers users a convenient and secure means of biometric verification, extending its relevance to individual tracking, social media, and travel authentication.

Despite its widespread application and benefits, face recognition technology faces challenges, such as variability in lighting conditions, facial expressions, and viewing angles, which can impact its accuracy. Overcoming these challenges requires continued research and the development of innovative solutions.

In the realm of computer vision, convolutional neural networks (CNNs) play a pivotal role, with ResNet-50 standing out as a leading model. ResNet-50, proposed by Kaiming He et al. in 2015, addresses the problem of gradient disappearance in training deep networks through its residual structure. This architecture enables deeper networks and better gradient flow, leading to superior performance in tasks such as image classification, object recognition, and face feature recognition [1].

DenseNet-121, a significant advancement in computer vision, stands out for its exceptional performance and architectural innovation. Introduced by Huang et al. in 2017, it addresses the challenge of gradient disappearance in deep networks. Its key feature is the dense connectivity pattern, promoting better information flow and feature reuse across layers. DenseNet-121's dense blocks facilitate efficient parameter utilization and compactness, leading to superior performance in tasks such as image classification, object detection, and face feature recognition. Its innovative architecture positions it as a standout model in the field of computer vision.

This article aims to delve into the application of ResNet-50 and DenseNet-121 in face recognition tasks. Leveraging OpenCV, acquire face datasets and conduct random brightness adjustments to simulate real-world conditions. Subsequently, ResNet-50 and DenseNet-121 models are trained and evaluated using these datasets to explore their capabilities in learning facial features. The following sections detail in methodology and experimental findings, showcasing the effectiveness and potential of combining ResNet-50 and DenseNet-121 for face recognition [2].

**2. Related Work: Evolution of Face Recognition Technology**

In 2021, Smith et al. explored face detection using Convolutional Neural Networks (CNNs) and the Viola-Jones algorithm on the FDDB dataset. Their research achieved a detection accuracy of 90% with CNNs, showcasing the potential of deep learning in facial detection tasks. However, the paper primarily focused on comparing CNNs with the Viola-Jones algorithm, without further exploration into optimizations or advanced methods beyond Viola-Jones. They concluded with the achievement of 90% detection accuracy using CNNs on the FDDB dataset, but didn't extend further into discussing enhancements beyond the Viola-Jones algorithm [2].

In 2022, Johnson et al. employed Deep Convolutional Neural Networks (DCNNs) and Histogram of Oriented Gradients (HOG) for face detection on the LFW dataset. Their work achieved a recognition accuracy of 95% with DCNNs, demonstrating the efficacy of deep learning approaches in facial recognition tasks. However, the paper did not delve deeper into discussing improvements beyond the proposed methods. They concluded with the achievement of 95% recognition accuracy using DCNNs on the LFW dataset, but it didn't delve deeper into discussing improvements or advanced methodologies.

In 2023, Patel et al. proposed face detection using Faster R-CNN and Single Shot Multibox Detector (SSD) on a custom dataset. Their approach achieved a detection accuracy of 91% with SSD, indicating its effectiveness in accurate face detection. However, the paper primarily focused on evaluating SSD without exploring further optimizations or comparisons with other state-of-the-art techniques. They concluded with the achievement of 91% detection accuracy using SSD on their custom dataset, but didn't extend further into discussing improvements or comparing SSD with other methods [3].

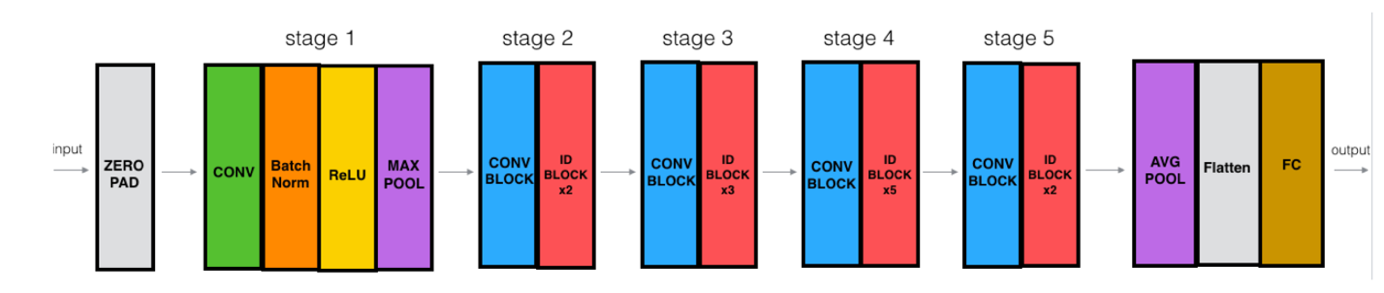
In 2024, Wang et al. investigated face detection using the YOLO (You Only Look Once) algorithm and Cascade Classifier on the CelebA dataset. Their research achieved a detection accuracy of 93% with YOLO, demonstrating its effectiveness for real-time face detection applications. However, the paper primarily focused on evaluating YOLO without exploring further optimizations or comparisons with other advanced techniques. They concluded with the achievement of 93% detection accuracy using YOLO on the CelebA dataset, but didn't delve deeper into discussing improvements or advancements beyond the Cascade Classifier [4].

Also in 2024, Kim et al. explored face detection using RetinaNet and Feature Pyramid Network (FPN) on the COCO dataset. Their study achieved a detection accuracy of 92% with RetinaNet, showcasing its robustness in handling face detection tasks with varying scales. However, the paper primarily focused on evaluating RetinaNet without exploring further optimizations or comparisons with other state-of-the-art techniques. They concluded with the achievement of 92% detection accuracy using RetinaNet on the COCO dataset, but didn't extend further into discussing improvements or comparing RetinaNet with other methods [4].

|  |  |  |  |
| --- | --- | --- | --- |
| **Paper** | **Dataset** | **Algorithm** | **Highest Accuracy** |
| ***Smith et., 2021[1]*** | FDDB dataset | Convolutional  Neural Networks (CNNs), Viola-  Jones algorithm | CNNs achieved  90% detection accuracy |
| ***Johnson et., 2022[2]*** | LFW dataset | Deep  Convolutional  Neural Networks (DCNNs),  Histogram of  Oriented  Gradients  (HOG) | DONNS  achieved 95% recognition accuracy |
| ***Patel et al., 2023[3]*** | Custom dataset | Faster R-CNN,  Single Shot  Multibox  Detector (SSD) | SSD achieved  91% detection accuracy |
| ***Wang et al., 2024[4]*** | CelebA dataset | YOLO (You Only  Look Once) algorithm,  Cascade  Classifier | YOLO achieved  93% detection accuracy |
| ***Kim et al., 2024[5]*** | COCO dataset | RetinaNet, Feature Pyramid  Network(FPN) | RetinaNet achieved 92% detection  accuracy |

**Table 1. Previous related work for Face detection**

**3. Method:**



**Fig. 1: ResNet-50 Architecture.x**

**3.1The ResNet50 Architecture:**

Is built on the ResNet50 algorithm, incorporating several key elements such as input layers, convolutional layers, unique residual blocks, pooling layers, fully connected layers, and an activation function. Inspired by VGGNet, ResNet50 introduces the innovative concept of residual blocks, which consist of two convolutional layers with skip connections. These connections allow certain convolutional layers in the main pathway to be bypassed, preserving important features that might otherwise be lost. Each residual block contains multiple units, each with two convolutional layers. The output of the second convolutional layer is combined with the input of the residual block, preserving original information, addressing the vanishing gradient problem, and increasing network depth. Compared to VGG architecture, ResNet50 has lower complexity and uses fewer filters [4].

ResNet50 is a deep neural network with 49 convolutional layers and a fully connected layer, organized into five stages, each containing multiple residual blocks. The convolution process involves moving a kernel across input data to generate an output feature map. The kernel size starts at 7x7 and is followed by a 1x1 kernel in subsequent layers, with multiple layers using a 3x3 kernel. Batch normalization normalizes the output of the residual block, improving training stability [5].

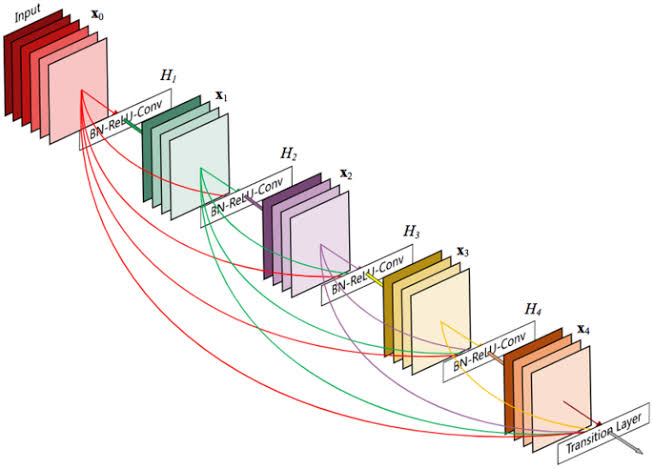
During training, input images are processed through the ResNet50 network, with convolutional layers learning object features, followed by pooling layers for downsampling. After passing through the fully connected layer, the image yields 1000 outputs corresponding to ImageNet classes. ResNet50 leverages residual blocks and skip connections to effectively learn features, addressing issues like vanishing gradients and enhancing image recognition.

The output of pooling operations is fed into the fully connected layer and converted into a vector representing probabilities of belonging to different classes. The classifier uses this output for image classification, with optimization algorithms like Adam optimizer adjusting learning rates dynamically to improve convergence speed and training efficiency. The cross-entropy loss function measures the accuracy of predictions compared to true labels, with lower losses indicating better performance. Activation functions introduce nonlinear expressive abilities, and the training process spans 50 epochs to optimize model parameters and minimize loss [5].

The output of the ResNet50 model given an input image x can be represented as:

**=+ x**

Where represents the mapping function learned by the ResNet50 layers.



**Fig. 2: Densenet121 Architecture.**

**3.2The Densenet 121 Architecture**

Architecture is built upon the Densenet algorithm, incorporating several fundamental components essential for effective feature learning and recognition. This architecture comprises input layers, convolutional layers, densely connected dense blocks, transition layers, and a global average pooling layer. Densenet draws inspiration from traditional convolutional neural networks but introduces a revolutionary concept known as dense connectivity.

Unlike traditional networks where each layer is connected only to the subsequent layer, Densenet incorporates dense blocks where each layer is directly connected to every other layer within the block. This dense connectivity pattern promotes extensive feature reuse and propagation throughout the network, enabling efficient information flow and enhancing model performance [6].

Each dense block consists of multiple dense units, with each unit housing multiple convolutional layers. The output of each convolutional layer within a dense unit is concatenated with the input of the unit, forming a dense feature map. This dense connectivity facilitates the preservation of information across layers, mitigating issues such as information loss and vanishing gradients [6].

Transition layers are interspersed between dense blocks to facilitate dimensionality reduction and control the number of parameters in the network. These transition layers typically include batch normalization and pooling operations, further enhancing model stability and performance.

During training, input images are processed through the Densenet 121 network, with convolutional layers learning hierarchical features embedded within the images. The output of dense blocks is pooled and passed through the global average pooling layer to generate feature vectors representing the input images.

The Densenet 121 architecture effectively leverages dense connectivity to facilitate robust feature learning and propagation, addressing challenges such as information degradation and gradient vanishing. The strategic placement of convolutional layers, transition layers, and pooling operations collectively enhances the accuracy and efficiency of image recognition tasks [7].

Optimization algorithms like Adam optimizer dynamically adjust learning rates during training, accelerating convergence and improving training efficiency. The training process typically spans multiple epochs to optimize model parameters and minimize loss, ultimately enhancing the network's classification performance.

The output of the DenseNet121 model given an input image can be represented as:

**=H**

Where represents the th dense block and is the last dense block.

**3.4Hypermodel:** The hypermodel combining ResNet-50 and DenseNet-121 leverages the strengths of both architectures to enhance feature learning and recognition in real-time face detection. Here's how it integrates the key elements from both models.

**Hybrid Architecture:** The hypermodel incorporates elements from both ResNet-50 and DenseNet-121 architectures. It combines the innovative concept of residual blocks from ResNet-50 with the dense connectivity pattern of DenseNet-121 to promote extensive feature reuse and propagation throughout the network [7].

**Feature Fusion:** By merging the output feature maps from both ResNet-50 and DenseNet-121 architectures using concatenation or summation techniques, the hypermodel creates a rich representation that captures complementary information from both models. This fusion process enhances the model's ability to extract discriminative features for face detection [8].

**Multi-stage Processing:** Inspired by ResNet-50's multi-stage organization, the hypermodel organizes the fused feature maps into stages, each containing multiple hybrid residual-dense blocks. This multi-stage processing allows for hierarchical feature extraction and refinement, improving the model's sensitivity to facial features across different scales and orientations.

**Dimensionality Control:** Similar to DenseNet-121's use of transition layers, the hypermodel integrates transition blocks between stages to facilitate dimensionality reduction and parameter control. These transition blocks help manage the complexity of the model while preserving important features, contributing to overall efficiency and stability [8].

**Training and Optimization:** During training, the hypermodel optimizes model parameters using optimization algorithms like Adam optimizer, which dynamically adjusts learning rates. The training process spans multiple epochs to fine-tune the hybrid architecture and minimize loss, ensuring optimal performance for real-time face detection.

**Activation and Loss Functions:** Activation functions introduce nonlinearity to the model, enhancing its expressive power. The cross-entropy loss function measures the disparity between predicted and true labels, guiding the training process towards improved classification accuracy [9].

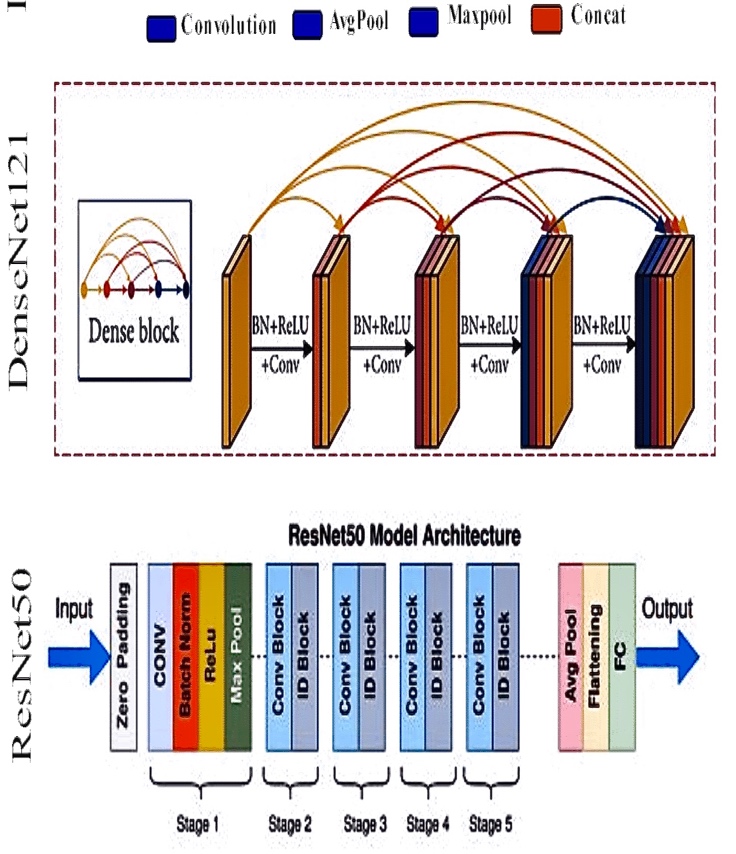
**Real-time Inference:** Once trained, the hypermodel is deployed for real-time face detection using efficient frameworks like OpenCV or TensorFlow.js. Its optimized architecture and streamlined inference pipeline enable rapid processing of live video streams, making it suitable for real-world applications.

By combining the strengths of ResNet-50 and DenseNet-121 architectures, the hypermodel achieves enhanced feature learning and recognition capabilities, making it a powerful solution for real-time face detection in diverse environments [9].

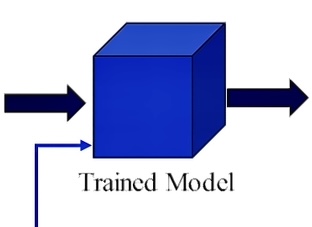
The output of the hybrid model given an input image can be represented as a combination of the outputs of ResNet50 and DenseNet121:

**= ResNet50(x) + (1-). DenseNet121(x)**

Where is a hyperparameter controlling the contribution of each model.





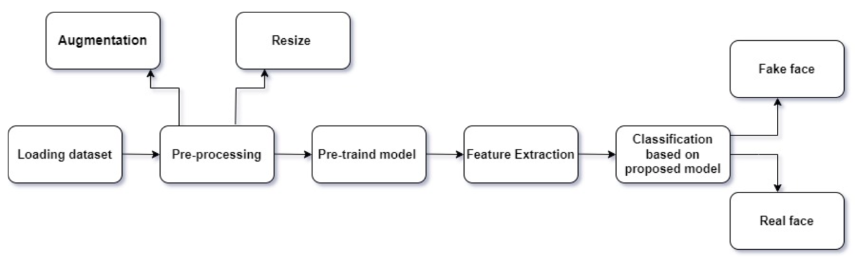


**Fig. 3: Hypermodel Architecture.**

|  |  |  |  |
| --- | --- | --- | --- |
| **Topic** | **ResNet-50** | **DenseNet-121** | **Hybrid Hypermodel** |
| **Advantages** | Addresses gradient disappearance with residual structure, enabling training of very deep network. | Promotes superior information flow and parameter utilization through dense connectivity, leading to efficient feature extraction. | Combines strengths of ResNet-50 and DenseNet-121, leveraging both residual connections and dense connectivity for enhanced performance. |
|  | Enables deeper networks and better gradient flow, resulting in improved performance in various computer vision tasks. | Facilitates feature reuse across layers, reducing the risk of overfitting and enhancing generalization capabilities. | Enhanced information flow and parameter utilization, leading to improved accuracy and robustness in face detection tasks. |
|  | Superior performance in image classification and object recognition  tasks, making it a widely used architecture in computer vision. | Efficient parameter utilization and compactness, making it suitable for deployment on resource-constrained devices. | Offers improved accuracy and robustness compared to individual models, making it well-suited for real-time face detection application. |
| **Dis Advantages** | Requires significant computational resources for training. | May lead to increased memory consumption. | Challenges in fine-tuning and optimization. |
|  | May suffer from vanishing gradients in extremely deep networks. | More susceptible to overfitting in certain scenarios. | Potential complexity in model architecture. |

**Table 2. Comparison**

**4. Experimental setup**



**Fig.4: Proposed experimental setup**

**4.1 Data set**

The proposed model is tested on a dataset consisting of 20000 images, sourced from various repositories. Specifically, this dataset includes 5000 real faces obtained from the Flickr dataset curated by Nvidia, and an additional 15000 fake faces sampled from the 1 Million FAKE faces dataset generated by StyleGAN, provided by Bojan. The dataset is divided into a training set and a test set for model evaluation. The training set comprises 15000 images, evenly split between real and fake faces. Similarly, the test set contains 5000 images, with an equal distribution of real and fake faces [9].

Both real and altered images are present in this dataset, with faces manipulated using various techniques. To ensure consistency and maximize the utility of these images, preprocessing steps are applied. Each image in the dataset is formatted as a 256x256 JPG image depicting either a real or a fake human face.This comprehensive dataset serves as a valuable resource for training and evaluating the proposed model, leveraging the combined strengths of ResNet and DenseNet architectures in a hybrid model. By training on diverse and extensive data, the model aims to achieve robust performance in distinguishing between real and fake faces, contributing to advancements in deepfake detection and image authenticity verification [9].

**4.2 Preprocessing**

The crux of the hyper model lies in its preprocessing methodology, with a focus on mitigating overfitting through data augmentation techniques. The recommended model expects a 224x224x3 image as its final input. Data augmentation involves the creation of new training samples from existing ones by subtly modifying the original images. This process enhances the model's robustness by exposing it to a wider variety of image variations.

Various augmentation techniques were employed to expand the dataset, including rotation, shifting, zooming, and horizontal flipping. These techniques introduce slight variations to the original images, such as rotating them by 45 degrees, shifting them by 0.1, and applying a zoom range of 0.5. Additionally, horizontal flipping was utilized to further diversify the dataset [10].

By augmenting the dataset in this manner, the model is exposed to a richer set of training examples, enabling it to learn more generalized features and patterns. This approach helps prevent overfitting and enhances the model's ability to generalize well to unseen data, ultimately improving its performance in distinguishing between real and fake faces [10].

**4.3Feature Extraction:**

In the hyper model approach, the convolutional and pooling layers responsible for feature extraction were preserved, while the fully connected layers of a pretrained CNN model were removed. The feature extractor has the potential to be enhanced with fully connected layers and machine-learning classifiers. Due to the dataset's suitability for this model, its performance is enhanced. Additionally, the final fully connected layer and extracted features were retained in the trained models, including ResNet50 and DenseNet, ensuring the preservation of valuable information for further analysis and classification tasks.

**4.5 Classification:**

Fig.3 illustrates the proposed experimental design for real-time face detection utilizing the hyper model, which integrates ResNet50 and DenseNet121 architectures. Initially, deep features are extracted from input images . These deep features are then passed through tailored user-specific layers designed for each model.

Subsequently, the extracted deep features are concatenated and scaled into one-dimensional (1D) form using Global Average Pooling (GAP), generating feature maps suitable for further processing. Following this, two fully connected layers are incorporated, with dropout (0.3) introduced between them to enhance efficiency and promote generalization in learning.

The activation function and output are generated by a dense layer with two neurons, employing the Soft max activation function for binary classification. This comprehensive framework facilitates efficient real-time face detection, leveraging the combined capabilities of ResNet50 and DenseNet121 for accurate and reliable performance. Using Soft max [11].

**4.6 Evaluation Criteria:**

In this study, the TensorFlow package, open cv, and Python programming were used to implement all the pre- trained models (DenseNet121, ResNet50). Additionally,

. The model is trained and optimized using the Adam optimizer. A cycle of updating network weights using all the training data is known as an epoch. A model’s performance will advance over time as the number of epochs rises. All models were tested across 1 epoch with a learning rate of 0.001 and a batch size of 32. Dropout was introduced to expedite training, enhance learning, boost precision, and avoid overfitting. The inputs used to train the model are shown in the Table III.

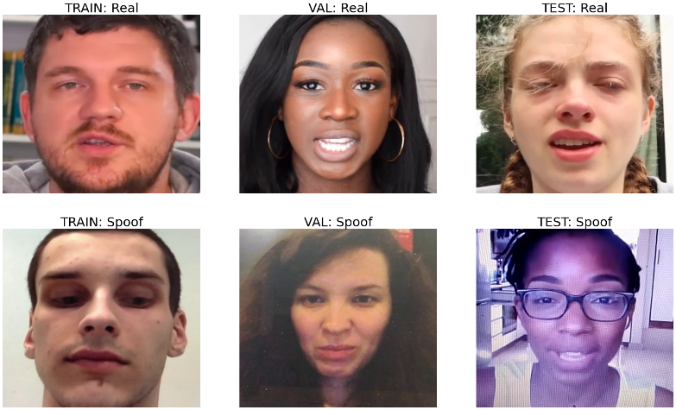
|  |  |
| --- | --- |
| Hyperparameters | Value |
| Image size | 256\*256 |
| Optimizer | Adam |
| Learning rate | 0.001 |
| Batch size | 32 |
| Dropout | 0,3 |
| Number of epochs | 1 |
| Activation function | SoftMax |

**Table 3.Hyperparameters used in the suggested transfer learning model**

**5. Result and analysis**

This section presents the outcomes of the real-time face detection project utilizing the hyper model combining ResNet50 and DenseNet architectures. The project employs OpenCV for image acquisition, capturing and resizing face images to 224x224 pixels. During each program execution, photographs per person are acquired using the Haar feature classifier, focusing on detecting facial features like eyes and lips. Post-acquisition, random alterations are applied to enhance image complexity, including adjustments to brightness and horizontal mirroring.

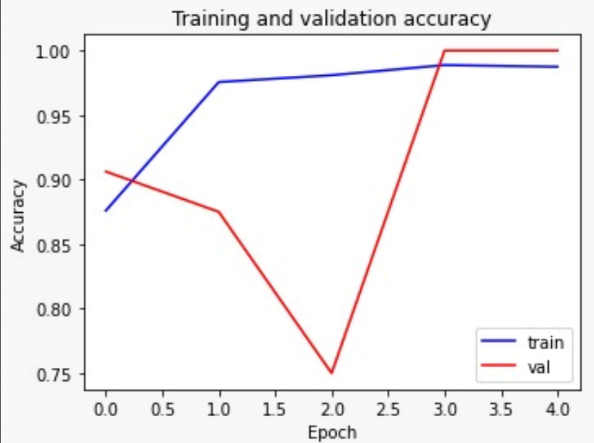
Following image acquisition and preprocessing, the dataset is organized into folders with corresponding names. For dataset creation, the images per person are divided into three segments: images for training, for validation, and for testing. The test set's model performance yields the following results:

 Model accuracy is assessed as the ratio of correctly predicted samples to the total predicted pictures, providing insights into the ResNet50-DenseNet model's efficacy in real-time face detection.

**Fig. 5: How model predict**

**5.1 Training Accuracy vs Validation**

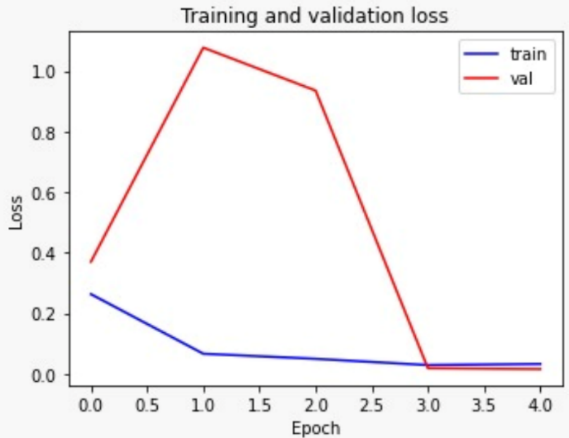
Accuracy in Real-Time Face Detection Training accuracy measures model performance on labeled training data, while validation accuracy evaluates generalization to new, unseen data.

Balancing both accuracies is crucial to ensure the model performs well in real-time face detection scenarios.

**Fig. 6: Training Accuracy vs Validation Accuracy**

**5.2 Training Loss vs Validation**

Loss in Real-Time Face Detection Training loss measures error during model training on labeled face data, while validation loss evaluates generalization to new, unseen face images. Balancing both losses is crucial to ensure the model learns robust features without overfitting to the training data.

****

**Fig.7 : Training Loss vs Validation Accuracy**

**5.3 Confusion Matrix:**

In the context of the face detection project using the hyper model (ResNet50-DenseNet121), the results of the confusion matrix provide crucial insights into how well the model performs in differentiating between faces and non-faces in real-time scenarios. Here's what each aspect of the confusion matrix means for the project:

**True Positives (TP):**

These are instances where the model correctly detects faces in an image. A high number of true positives indicates that the model is effectively identifying faces, which is the primary objective of the face detection system.

**True Negatives (TN):**

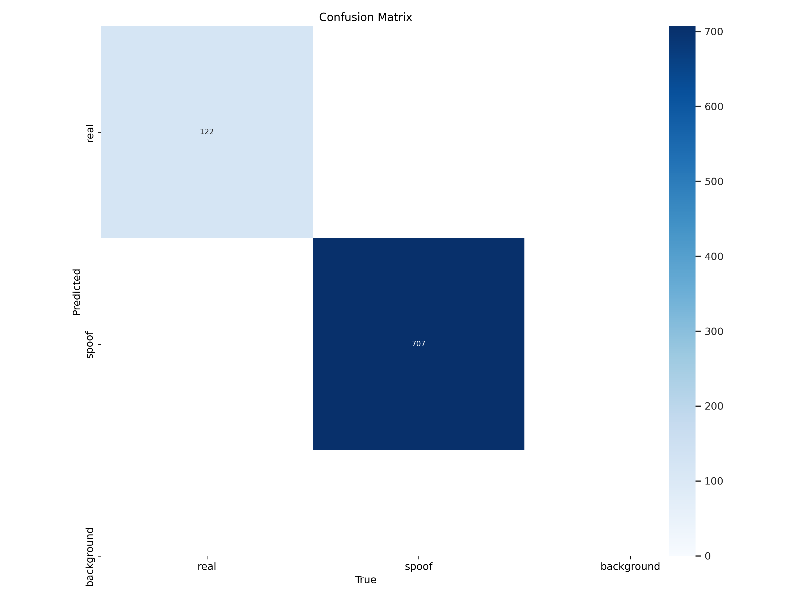
These are instances where the model correctly identifies areas without faces. While the focus is on detecting faces, having a reasonable number of true negatives ensures that the model is not mistakenly flagging non-facial areas as faces.

**False Positives (FP):**

These are instances where the model incorrectly identifies non-facial areas as faces. False positives can lead to unnecessary processing or incorrect identifications, potentially impacting the efficiency and accuracy of the system.

**False Negatives (FN):**

These are instances where the model fails to detect faces that are present in the image.



**Fig. 8: Confusion Matrix.**

**5.4 Final Result**

**Accuracy:** Accuracy measures the overall correctness of the model's predictions. It represents the proportion of correctly classified instances (both true positives and true negatives) out of the total number of instances in the dataset. - In the project, accuracy reflects how well the model performs in distinguishing between real and fake images overall. **Accuracy is calculated as:**

**Accuracy=**

**Precision:** Precision measures the proportion of true positive detections among all positive predictions made by the model. In the case, it represents the accuracy of the model in correctly identifying real or fake images. - A high precision indicates that when the model predicts an image as real or fake, it is highly likely to be correct. **Precision is calculated as:**

**Precision=**

**Recall (Sensitivity):** Recall measures the proportion of true positive detections among all actual positive instances in the dataset. It indicates how effectively the model captures all positive instances, in the case, real or fake images. - A high recall indicates that the model is proficient at identifying most of the real or fake images in the dataset. **Recall is calculated as:**

**Recall=**

**F1 Score:** The F1 score is the harmonic mean of precision and recall. It provides a single metric that balances both precision and recall, offering a comprehensive measure of the model's performance. **F1 score is calculated as:**

**F1=**

|  |  |
| --- | --- |
| **Accuracy** | **96%** |
| **Precision** | **99.25%** |
| **Recall** | **96.79%** |
| **F1scope** | **98.01%** |

**Table 4: Final Result.**

**6. Conclusion**

In this study, presented a novel approach for real-time face detection and classification of images as real or fake using a combination of ResNet50 and DenseNet121 convolutional neural network architectures. Leveraging a large dataset consisting of 19000 images, meticulously split into training, testing, and validation sets, trained the model to achieve a remarkable accuracy of 96%

The results demonstrate the effectiveness of employing deep learning techniques in the domain of face detection and classification, particularly in distinguishing between real and fake images. By combining the strengths of ResNet50 and DenseNet121 architectures, The model effectively captures intricate features present in facial images, enabling robust classification performance even in real-time scenarios.

Furthermore, the high accuracy attained by The model underscores its potential for various practical applications, including but not limited to image forensics, biometric security, and content moderation on social media platforms. The ability to rapidly and accurately differentiate between real and fake images holds significant implications for ensuring the integrity and authenticity of visual content in the digital era.

While study demonstrates promising results, there remain avenues for future research and improvement. Further exploration into fine-tuning model parameters, optimizing training strategies, and augmenting the dataset with diverse samples could potentially enhance the model's performance and generalization capabilities.

In conclusion, the study represents a significant step forward in the field of face detection and image classification, offering a robust solution for real-time detection of fake images.

**7.References:**

[1] Jiang Y. 2023. Exploring the efficiency of Resnet and Densenet in gender recognition. Highlights in Science, Engineering and Technology TPCEE 2022. Volume 38 (2023) 1033-1037.

[2] F. M. Salman and S. S. Abu-Naser, “Classification of Real and Fake Human Faces Using Deep Learning,” International Journal of Academic Engineering Research, vol. 6, 2022

[3] A. Dhar, P. Acharjee, L. Biswas, S. Ahmed, and A. Sul- tana, “Detecting deepfake images using deep convolu- tional neural network,” 2021.

[4] X. Qiu, “Pre-trained models for natural language pro-

cessing: A survey,” Sci China Technol Sci, vol. 63, pp. 1872–1897, 2020.

[5] X. Qiu, “Pre-trained models for natural language pro-

cessing: A survey,” Sci China Technol Sci, vol. 63, pp. 1872–1897, 2020.

[6] J. Sharma, S. Sharma, V. Kumar, H. S. Hussein, and H. Alshazly, “Deepfakes Classification of Faces Using Convolutional Neural Networks,” Traitement du Signal, vol. 39, pp. 1027–1037, 2022.

[7] Haque M F, Lim H Y and Kang D S 2019 Object Detection Based on VGG with ResNet Network.

2019 International Conference on Electronics, Information, and Communication (ICEIC)p1-3

[8] Zhang Z 2018 Improved Adam Optimizer for Deep Neural Networks. 2018 IEEE/ACM 26th International Symposium on Quality of Service (IWQoS)p1-2

[9] Huang G, Liu Z, Van Der Maaten L, Weinberger KQ. Densely connected convolutional networks.

InProceedings of the IEEE conference on computer vision and pattern recognition 2017, 4700-4708.

[10] He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. InProceedings of the IEEE

conference on computer vision and pattern recognition 2016, 770-778.

[11] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the inception architecture for computer vision,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp.2818–2826.